Evaluating Resistance of MCML Technology to Power Analysis Attacks Using a Simulation-Based Methodology

Francesco Regazzoni[1], Thomas Eisenbarth[2], Axel Poschmann[2], Johann Großschädl[3], Frank Gurkaynak[4], Marco Macchetti[5,⋆], Zeynep Toprak[6], Laura Pozzi[7], Christof Paar[2], Yusuf Leblebici[6], and Paolo Ienne[8]

[1] ALaRI – University of Lugano, Lugano, Switzerland
regazzoni@alari.ch
[2] Horst Görtz Institute for IT Security, Bochum, Germany
{eisenbarth,poschmann,cpaar}@crypto.rub.de
[3] University of Bristol, Department of Computer Science, Bristol, UK
johann@cs.bris.ac.uk
[4] Swiss Federal Institute of Technology – ETH, Zurich, Switzerland
kgf@ee.ethz.ch
[5] Nagracard SA, Cheseaux-sur-Lausanne, Switzerland
Marco.Macchetti@nagra.com
[6] School of Engineering – EPFL, Lausanne, Switzerland
{zeynep.toprak,yusuf.leblebici}@epfl.ch
[7] Faculty of Informatics – University of Lugano, Lugano, Switzerland
laura.pozzi@unisi.ch
[8] School of Computer and Communication Sciences – EPFL, Lausanne, Switzerland
Paolo.Ienne@epfl.ch

Abstract. This paper explores the resistance of MOS Current Mode Logic (MCML) against attacks based on the observation of the power consumption. Circuits implemented in MCML, in fact, have unique characteristics both in terms of power consumption and the dependency of the power profile from the input signal pattern. Therefore, MCML is suitable to protect cryptographic hardware from Differential Power Analysis and similar side-channel attacks.

In order to demonstrate the effectiveness of different logic styles against power analysis attacks, two full cores implementing the AES algorithm were realized and implemented with CMOS and MCML technology, and a set of different types of attack was performed using power traces derived from SPICE-level simulations. Although all keys were discovered for CMOS, MCML traces did not presents characteristic that can lead to a successful attack.

1 Introduction

During the past ten years, a number of new techniques for attacking implementations of cryptographic algorithms have been discovered. These techniques

⋆ This work was done while at C.E. Consulting (Altran Group) Milan, Italy.

M.L. Gavrilova et al. (Eds.): Trans. on Comput. Sci. IV, LNCS 5430, pp. 230–243, 2009.

exploit information leaking from a device (e.g., a smart card) while data is being processed. The term *side-channel attacks* summarizes all possible ways of collecting the leaked information: power consumption, timing, and electromagnetic emission are possible examples [MOP07]. Side-channel attacks which exploit the power consumed by a device were reported for the first time in 1999 by Kocher et al [KJJ99]. The power consumption of a device strongly depends on the data being processed, thus leaks information about the secret key. Among the different types of power-based attacks available in literature, the most common are *Simple Power Analysis (SPA)* and *differential power analysis (DPA)*. The latter and its powerful variant called *correlation power analysis (CPA)* are of particular interest since they do not require specific knowledge about the implementation of the target device to be effective.

In this paper we present a design flow that enables the evaluation of Power Analysis Attack resistance and using it we demonstrate the robustness of a special logic style, namely MOS Current Mode Logic (MCML), against such attacks, considering in particular SPA and CPA. Previous papers on this subject just argued robustness qualitatively or required hardware manufacturing to prove it. Contrary to past works, we evaluate the robustness of MCML *with real attacks* and *without the need for manufacturing prototypes*. To achieve this result, we developed a design flow and a SPICE-level simulation environment derived from the one presented by Bucci at al. [BGLT04], that allows collection of power traces in reasonable time, thus enabling a more direct experimental study of the resistance of complex blocks, such as entire cryptographic cores. As a result, our traces are much closer to the circuit real behavior than those obtained simulating only small portion of a core. A clear advantage of the proposed simulation-based evaluation is that in this way it is easy and thus possible to iterate the design flow to investigate further points of optimization before the fabrication of the real chip.

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 overviews the AES algorithm, and Section 4 describes MCML technology. The design flow, including simulation-based power analysis, is explained in Section 5, and simulation results are presented in Section 6. Finally, conclusions are drawn in Section 7.

2 Background and Related Work

Side-channel cryptanalysis has emerged as a serious threat for smart cards and other types of embedded systems performing cryptographic operations. It was demonstrated in a number of publications that side-channel attacks are an extremely powerful and practical tool for breaking unprotected (or insufficiently protected) implementations of cryptosystems. These attacks exploit the fact that the execution of a cryptographic algorithm on a physical device leaks information about sensitive data (e.g., secret keys) involved in the computations. Many sources of side-channel information have been discovered in recent years, including the power consumption and timing characteristics of a cryptographic algorithm [Koc96, KJJ99], as well as deliberately introduced computational faults

[BS97]. *Simple Power Analysis (SPA)* uses the leaked information from a single computation, while *Differential Power Analysis (DPA)* utilizes statistical methods to evaluate the information observed from multiple computations [KJJ99]. Currently, there exists no perfect protection against DPA attacks. However, by applying appropriate countermeasures, it is possible to make the attacker's task more difficult and expensive.

A multitude of so-called *DPA-resistant logic styles* have been proposed during the past five years. The idea behind these logic styles is to tackle the problem of side-channel leakage at its actual root, namely at the hardware level. The power consumption of circuits realized with DPA-resistant logic cells is uniform and, in the ideal case, independent of the processed data and the performed operations. The first concrete implementation of a DPA-resistant logic style was reported by Tiri et al. in 2002 [TAV02]. Their *Sense Amplifier Based Logic (SABL)* combines the concepts of dual-rail logic and pre-charge logic [MOP07]. SABL cells have a constant power consumption, provided that they are designed and implemented in a carefully balanced way. All SABL cells of a circuit are connected to the clock signal and become pre-charged simultaneously, which causes very high current peaks. Furthermore, SABL cells require at least twice as much silicon area as conventional CMOS cells and suffer also from high delay. Besides the logic cells, also the wires connecting these cells must be routed in a special balanced way to achieve a uniform power profile.

The present work improves on the results of our previous work [RBE+07] in several substantial ways. Below is a list of the main differences along with a brief explanation.

- The custom design flow for MCML has been completed. Thus, it is now possible to start from the same HDL netlist for both CMOS and MCML, rather than completely design by hand the netlist for the latter case.
- The simulation flow has been extended in order to support a back-end design phase. This task, particularly challenging for MCML technology, has been carried out for all analyzed circuits. Net parasitics have been extracted into *SPEF* files and back-annotated on the netlists. Although this increased the simulation time, results mimic more closely the actual behaviour of the fabricated device.
- Full cryptographic cores (implementations of the AES block cipher algorithm) have been considered as targets of the attacks. This has surely a negative impact on simulation speed, but is representative of a typical real attack. This step was made possible by the level of maturity reached in the simulation flow.

Additionally, note that post-processing and stimuli writing procedures have been fully automated, the same has been done for the simulation and attack routines that have also been extended to support Simple Power Analysis attacks.

3 Overview of the AES Algorithm

In this section we provide an overview of the Rijndael (AES [IoSTN01]) algorithm and some highlights on its possible implementation, focusing on the two that we used.

The Rijndael algorithm implements a block cipher for symmetric key cryptography, supports a key size of 128, 192 and 256 bits, and allows for a block size of 128 bits. Every block is represented using 32-bit words. The number of words that compose the input block is equal to 4, while the length of the key can be a sequence of 128, 192 and 256 bits, and can take the values 4, 6, or 8, which reflects the number of words the key is composed by.

The algorithm works on a two dimensional representation of the input block called state, that is initialized to the input data block and holds the intermediate result during the cipher and decipher process, and ultimately holds the final result when the process is completed. All the transformations of the algorithm are grouped in a single function called round. The round is iterated a specific number of times that depends on the key size; specifically, for a key length equal to 128, 192 or 256 the number of rounds is equal to 10, 12 and 14, respectively.

The encryption process starts by copying the input block into the state array, followed by the first key addition. In the encryption process, the round function is composed by four different transformations. *ShiftRows* cyclically shifts to left the bytes in the last three rows of the state with different offsets. *SubBytes (or S-box)* operates independently on each byte of the state and is composed by the multiplicative inverse in the finite field GF(2^8) followed by an affine transformation over GF(2). *MixColumns* multiplies modulo $x^4 + 1$ the columns of the state by the polynomial $\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. *AddRoundKey* adds the round key to the state. To generate all the needed round keys, the AES algorithm takes the secret key k and performs the expansion routine to generate a total of $\mathrm{Nb} \times (\mathrm{Nr}+1)$ words. The round transformations are cyclically executed at every round: all the *Nr* rounds are identical with the exception of the final round, which does not include the *MixColumns* transformation.

Decryption is similar to the encryption process and uses the same basic transformations, but inverted. The key schedule is identical to the one described for the encryption process but starts with the last round key.

Many alternatives are available when implementing an AES core, the choice among them being driven by application constraints and by performance, area, and power trade-offs. Our goal is to estimate the level of robustness given by the MCML technology with respect to the CMOS one, thus instead of attacking a single implementation of the block cipher, for each of the two attacks we considered a different core, selected to perform this evaluation in the best possible conditions for the adversary. The two considered AES implementation have a datapath of 128 and 32 respectively. In the first core considered, a single register of 128 bit is used to store the result of the first key addition and the results of the rounds computation, and the key is unrolled *on the fly*, while ciphering the data. The latter has a number of 32-bit registers that are used to store the result of each transformation inside the round, with the only exception of

the shift rows. In this implementation, all the round keys are computed before starting ciphering.

4 Design of DPA-Resistant Functional Units Using MCML Gates

The circuit-level implementation of DPA-resistant logic gates requires systematic use of circuit techniques that (i) have significantly reduced power supply current levels, (ii) do not produce prominent current spikes or fluctuations during the switching events, and (iii) do not exhibit a significant input pattern-dependence with respect to current drawn from the power supply [TV03]. It is worth noting that the classical CMOS logic gates do not fare particularly well in any of these categories, and therefore, are not considered to be a good choice for DPA-resistance, in general. Standard CMOS digital gates are notorious for generating sharp and input-pattern dependent current pulses (also referred to as delta-I noise [GR99, AAE02]) due to charging and discharging of the gate's parasitic capacitances and fan-out.

Due to the differential and current steering nature of the its logic style, Current Mode Logic (CML) reduces the generated switching noise by about two orders of magnitude [TAY+05, MKA92]. The low delta-I noise generation makes the CML style an excellent candidate for DPA-resistant logic gate design.

In detail, a MOS Current Mode Logic (MCML) gate consists of a tail current source, a current steering logic core, and a differential load, as shown for the simplest MCML gate, the MCML buffer, in Figure 1. The operation of MCML circuits is based on the principle of re-directing (or switching) the current of a constant current source through a fully differential network of input transistors, and utilizing the reduced-swing voltage drop on a pair of complementary load devices as the output. A logic inversion without additional delay is possible by simply exchanging the differential terminals. The operation principle already suggests that the power consumption is static (the circuit must dissipate the same amount of current continuously) regardless of the switching activity and fan-out conditions. True differential operation of the circuit with small output voltage swing ensures fast switching times. Note that the propagation delay is proportional to the output swing, and independent of the power supply voltage. Other advantages include better noise immunity compared to classical CMOS logic circuits, and significantly less switching noise.

The supply current fluctuation in MCML gates is typically 5% of the nominal tail current during switching events. Figure 2 shows the simulated current variation of an MCML buffer for a fan-out of 5. MCML circuits are also more robust against common-mode fluctuations (power supply noise) due to their inherent common-mode rejection as a result of full differential signaling property.

From the DPA-resistance point-of-view, it can be seen that the supply-current variation of the MCML gate will remain significantly smaller during switching events, compared to that of a conventional CMOS gate. At the same time, the magnitude of the supply-current variation is largely independent of the applied

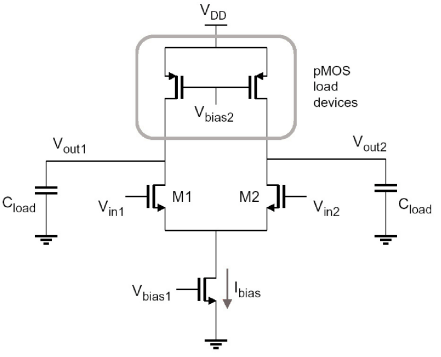

Fig. 1. Schematic of an MCML buffer (or MCML inverter, depending on the output signal definition)

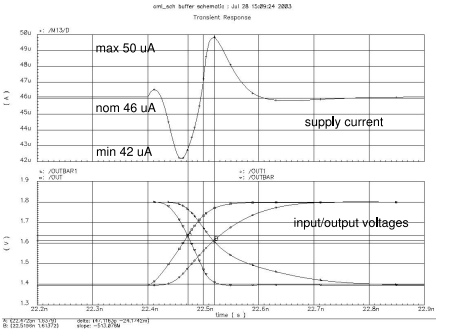

Fig. 2. Simulated gate delay and supply current fluctuation of an MCML buffer for a fan-out of 5

input vector, as well as of the fan-out load capacitance. The amount of static current dissipation can be reduced dramatically while preserving all of the advantages concerning the DPA-resistance, at a lower speed, when the transistor sizing is done to satisfy modest speed constraints (e.g., a typical switching speed of 400MHz). It was shown [TAY+05] that the peak current fluctuation of the classical CMOS realization is in the order of 28mA, while the current fluctuation of the MCML version remains confined to a narrow band of about 0.5mA, around the constant value of 11.5mA. A more detailed analysis was performed by modelling also the measurement set-up: a probing instrument having a low-pass filter characteristic and the filtered output was monitored. As expected, the design based on CMOS logic still shows large variations (400μA peak), sufficient to be distinguished quite easily. On the other hand, the maximum current fluctuation in the MCML-based design remains below 25μA, further increasing DPA-resistance of the security-critical block.

Besides being beneficial from the security point of view, usage of the MCML technology also poses additional constraints while designing secure devices. The main drawbacks regard the area requirements, MCML gates are generally 1.5 to 2 times larger than their CMOS counterparts. The power consumption is also higher, particularly for low operating frequencies, where the constant power consumption of the MCML gates does not offset the dynamic switching power of the CMOS gates. In particular, this increased consumption can adversely affect the current budget in power-constrained devices such as smart cards or cryptographic tokens. When low power is a design constraint, it may become necessary to isolate the areas of the design which are critical from the point of view of security and implement them using MCML gates, the rest of the circuit being realized using standard CMOS libraries. This approach obviously increases the complexity of the back-end design phase, as it introduces the need to deal with multiple technology libraries.

5 Design and Simulation Flow

The robustness of a hardware implementation of a block cipher against power analysis attacks can be evaluated by means of circuit simulation at different stages of the design flow. The decisive proof is obtained when the actual fabricated microchip is attacked using high frequency probes and an oscilloscope; nonetheless, *mounting an attack using the current traces obtained from transistor-level simulation can be useful to get a good approximation of the actual level of Power Analysis resistance, and an indication of possible sources of weakness.*

The simulation techniques used by designers for this robustness evaluation are typically divided in two groups: at the *analog level* or at the *logic level* [MOP07]; the first provides higher precision, while the second is faster. A common way of achieving the best of two worlds, i.e., precise results while keeping the simulation speed high, is to divide the circuit into blocks and simulate at the SPICE level only the parts under analysis. Although this approach provides results with good approximation, the problem with limiting the simulation to only parts of the circuit is that one eliminates the contribution of the algorithmic noise to the power consumption (the noise produced by all components of the circuit that are not targeted by the attack hypothesis). The only way to obtain a more realistic situation is to simulate the entire core, but this raises two main challenges: the required simulation time and the capability of the simulation flow to handle complex designs automatically. Indeed, automation is of capital importance: while it might be possible to manually adapt a small portion of a cryptographic core, it is certainly not so for a complex design.

To achieve both goals of fast simulation time and high accuracy, we developed an automated flow based on existing tools and described in the following. The actual design flow presents some differences for the two cases of CMOS and MCML; even though the RTL description of the core is the same, the flow of generating the netlist and the extraction of parasitic is different for the two considered technologies.

In the CMOS case, an HDL description of each of the cores under attack has been synthesized using Synopsys Design Compiler on the UMC 0.18μm process. Placing and routing phases are carried out using Cadence Design Systems SoC Encounter, and a parasitic information file is produced along with the verilog netlist of the circuit. These are used together with the SPICE models of the technology cells to run a transistor level simulation using Synopsys Nanosim. The employed transistor models are the BSIM3 p-MOS and n-MOS models.

In the MCML case, the standard ASIC design flow has been extended by introducing a complete library with views for transistor level simulation (schematic), schematic capture (symbol), synthesis (lib), placement and routing (abstract), and physical design (layout). It is worthwhile here to discuss the way in which this library has been built and under which aspects the extended flow is different from the standard CMOS flow; for a more comprehensive discussion the reader is referred toas the work of Badel at al. [BGI+08].

We first describe the process of the MCML logic cell generation. Generally speaking, the logic function implemented by an MCML differential cell is given

by the network of nMOS transistors (e.g. M1 and M2 in Figure 1); thus, as a first step, a basic set of nMOS networks (called footprints) has been created. Since the speed characteristics of the cell are adversely impacted by the number of levels of nMOS transistors in the footprint, a limit of 3 levels was considered. The basic set comprises 19 different footprints, which by exhaustive search have been found to be sufficient to implement all functions that can be mapped to 3-level MCML gates. Starting from the set of footprints, we can explore all different ways of assigning logic inputs to the transistor gates; 63 unique functions with 1 to 7 input variables are produced in this way, as well as 45 redundant functions, whose cell realizations present different electrical characteristics. Thus, a total of 108 standard cell templates are obtained. As a last step, we exploit a property of the differential cells for which a switching of the differential (input or output) pins results in a complementation of the associated boolean variable. We extensively apply this transformation and obtain a total of 4660 differential cells describing different logic functions that make up the core of the dedicated technology library.

In a typical design flow, different realizations of the same logic cells characterized by different driving strengths are usually needed; in the MCML case, switching speed is directly proportional to the amount of static current injected by the current source. Thus, by simply scaling the footprints of the cells we easily obtain such variants that will be added to obtain a rich and versatile cell library.

A second important process is the generation of a so-called fat library, in addition to the fully differential library. The two libraries share the same cell footprints and electrical characteristics, but the former contains single-ended (non differential) variants of the cells; this library is the one actually used during the synthesis and place&route phases. The reason is that modern place&route tools treat differential wires as different variables, which are thus routed in the layout in different ways; as a result the differential pair will typically undergo different noise contributions, and will suffer from mismatches in the capacitive loads. When the place&route phase has ended, the fully differential library is used for the layout phase; the single-ended cells are replaced by the differential ones (with zero impact on the footprint) and the single-ended wires are cut into pairs of wires which are thus (by-design) always side by side in the chip layout. Of course to render this step seamless, the cell input/output pins must be designed in an opportune way so that connections with the differential wires are easily introduced; an exact description of this achievement can be found in the work of Badel et al [BGI+08]. The result of the full process is an extended design flow, ranging from RTL design to layout, that can be used to design fully-differential MCML digital circuits with enforced differential routing. We use the output of the place and route phase along with the MCML technology SPICE models to run transistor level simulations of the full circuit.

As already mentioned, analog simulation can be done on netlists produced by different stages of the design flow. A first possibility is to use the current traces obtained from a post-synthesis power simulation. This approach allows

evaluation of circuit DPA resistance at a very early stage. However, the current traces obtained are rather inaccurate, because the contribution to the power consumption of the wire loads and parasitics is not considered. Depending on the attack point, such a consumption can have a significant effect on the side-channel resistance. Therefore, going one step further, and using the outputs of place&route tool for simulation, allows us to obtain power traces that are much more realistic.

In our work, we have decided to run accurate transistor-level simulations of the post-place&route netlist, at very high timing resolution (about 10ps) and with no additional noise coming from the measurement device or the environment. From one point of view, this is a best-case condition for an attacker; on the other side there are certainly some physical effects that cannot be correctly modeled, for instance crosstalk between adjacent nets or variations in the manufacturing process. A worthwhile-mentioning advantage of simulations is that in this way it is also possible to iterate the design flow to investigate further points of optimization.

We would like to underline that simulation results of Nanosim are comparable to those of SPICE (when Nanosim is set to run with full capabilities), but the simulation process requires significantly less time to be carried out. This is beneficial not only because the number of possible design iterations is greatly increased, but also because it enables simulation and verification of the robustness of complex designs such as an entire cryptographic core. Results obtained by simulating only a small portion of a core can miss a correct simulation of algorithmic noise as well as correct wire loads and parasitics, while simulation results of an entire cryptographic core are much closer to the real behavior of the circuit.

6 Resistance against Power Analysis

In this section we describe the attacks we mounted on the CMOS and MCML implementations of AES core and we compare the results. In this work we focused on attacks based on the analysis of the power consumption, and in particular in the two powerful ones: Simple Power Analysis (SPA) and Differential Power Analysis (DPA).

In an SPA, an attacker measures the power consumed by a device while performing cryptographic operations and, by observing the traces, deduces information produced either by the *Hamming weight leakage* or by the *transition current leakage*. Both of them are justified by the fact that the amount of current is directly proportional to the Hamming weight of the processed data, hence it can be derived. DPA attacks, on the other hand, are more effective than SPA attacks, but also more difficult to mount. A typical DPA attack consists of four steps: At first, an intermediate key dependent result is selected as the target, then the attacker encrypts (decrypts) a certain number of known plaintexts (ciphertexts) and measures the corresponding power consumption traces. Subsequently, hypothetical intermediate values are calculated based on a key guess and they are used as input of a selection function. This function is used to partition the

power traces into sets, depending on the values of the intermediate results. The difference of means of the two sets is then calculated and a peak is clearly visible for the right key hypothesis in correspondence to the time frame where the information is leaked.

An improvement with respect to DPA attack, called *correlation power analysis* (*CPA*), was discussed in [BCO04]. It hypothesizes the Hamming weight or the distance of Hamming of the targeted register and evaluate the hypothesis statistically. Usually CPA shows better results than the original DPA because it uses hypotheses based on multiple bits rather than the single bit typical of DPA. The statistical correlation $\rho_{(P(t),H)}$ between the power traces $P_n(t)$ and the hypothesis H is a normalized value between $-1 \leq \rho \leq 1$ where $\rho = 1$ ($\rho = -1$) means that the variables $P(t)$ and H are perfectly correlated (anti-correlated) and $\rho = 0$ means there is no correlation at all. The strongest correlation corresponds to the right key hypothesis.

Mounting the Attacks

Using the simulation flow described in Section 5, we obtained the power traces for attaking the AES algorithm described in Section 3. It is important to notice the differences between the simulated and the real attack. In a real environment, an attacker has to collect a huge number of traces in order to filter out the noise. In fact, when power consumption of any device is measured, the collected traces include noise, both thermal and algorithmic, the latter is produced by other components of the device. Since it is an uncorrelated normally distributed random variable, the noise can be filtered out by increasing the number of traces. The simulation environment we used is partially noise free: only algorithmic noise is present into the traces. Furthermore, the simulation was performed with a very high resolution both for the current (1μA) and the time (10ps), which is the best possible condition for an attacker.

To evaluate the resistance against the power based attacks, we considered the two most powerful univariate ones: the Simple Power Analysis Attack, and the powerful variant of Differential Power Analysis based on correlation. This decision is coherent with the goal of the paper: we aim to provide a realistic evaluation of Power Analysis resistance by means of simulation for MCML and CMOS rather then attack a specific implementation of a cryptographic algorithm. Since none of the considered logic styles offer masking schemes, practically there would not be any benefit in considering attacks of higher order. We thus concentrate on the SPA and CPA and we selected for each of the attacks a core that would represent the best possible situation for the attacker and we realized each of the two cores using both of the technologies. For the same reason, we decided to use as attack point a register, since is well known in literature that this is the easiest point for mounting an attack (because the signals are synchronized by the clock)[MOP07].

The first evaluation was done performing a Simple Power Analysis attack: in the target core of this attack, the output of the first key addition is stored into a register. As depicted in Figure 3, our SPA attack targeted that register. Coherently with the purpose of this paper, with this attack we are interested in

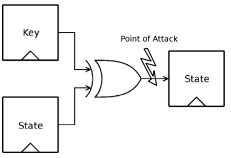

Fig. 3. Point of attack for SPA

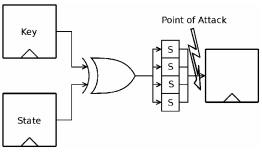

Fig. 4. Point of attack for CPA

verifying if the MCML logic, that is fully differential, is sufficient to protect the circuit rather then the robustness of a specific implementation under test.

The attack was performed bit by bit. After the application of a reset signal, that sets all flip-flops in the register to zero, we applied a plaintext in which all bits were set to logical 0. The result of the xor of this input with the secret key was then stored in the register and the power consumption in this point was measured. A second measure was performed as the first, with the only difference that one bit of the input plaintext was changed from 0 to 1. The two traces obtained for CMOS are plotted in Figure 5, that present the two measure overlapped. As is possible to see, the trace corresponding to the plaintext with the bit set to 1 has a higher power consumption with respect to the same situation when the plaintext is 0. This clearly indicated that the value of the secret key in corrispondence to this specific bit is 0. By iterating this procedure for all the bits of the key, it is possible to reconstruct all the correct values of the secret key by simply looking at the difference between the reference power trace with the plaintext all zero and the one with a bit set to 1 in corrispondence to the target bit of the key.

The same attack was perfomed on the same AES core implemented using MCML technology. As can be seen from Figure 6, the two traces corresponding to the plaintext 0 and the plaintext with the target bit set to 1 are completely overlapped and thus is impossible to derive the value of the key bit.

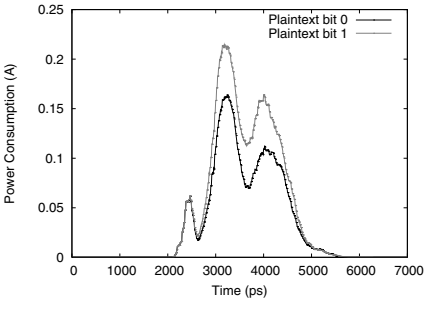

Fig. 5. SPA on CMOS: the power consumption traces clearly indicate the value of the target key bit

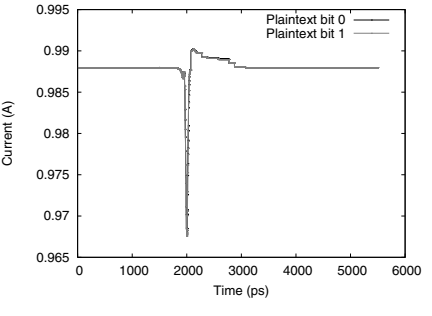

Fig. 6. SPA on MCML: the power traces are completely overlapped, thus not distinguishable

A second evaluation of the MCML resistance was done mounting a CPA. For this purpose we used a 32 bit datapath AES core, and we targeted the register that stores the output of 4 S-boxes, as depicted in Figure 4. To mount the attack, we used a selection function based on the Hamming weight of one byte of the target register. It is important to notice that 24 bits are not part of the hypothesis, thus they are contributing as algorithmic noise. This makes our attack more difficult with respect to a situation where only a single S-box is considered, but the problem can be easily solved by increasing the number of collected traces.

Repeated attacks performed by ciphering 600 random plaintexts were computed both for CMOS and MCML technology. In all these cases our attacks on the CMOS logic were always successful. The differential trace of the correct key (plotted in black) is the one that clearly shows the highest value for the correlation, thus it is clearly distinguishable from the other ones, as can be seen from Figure 7 (hypotheses based on the Hamming weight of the register), where the correlation value of $\rho_{(P(t),H)} = 1$ clearly indicates the guessed key was the correct one.

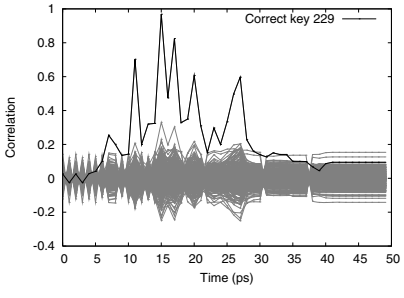

Fig. 7. CPA on CMOS technology

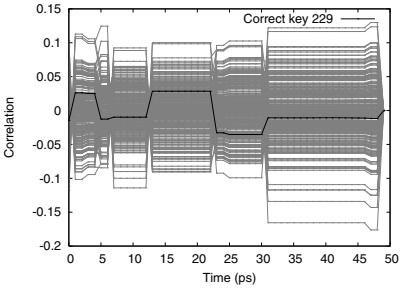

Fig. 8. CPA on MCML technology

As for the SPA, the same CPA attack was performed on the same core implemented using MCML technology. In this case the situation is completely different: in all the experiments in fact, no keys were found. An example of CPA attack on MCML technology is plotted in Figure 8. As can be seen, the black line representing the correct key is not distinguishable from the remaining differential traces that are plotted in gray. Additionally, it can be noticed that, for all the key guess, the maximum absolute value for the correlation is about 0.17. It is important to notice that in an attack mounted on a real device, this values are so small that they are very likely to be completely overshadowed by the noise of the measurement set-up, making the attack more difficult. Once again, we stress the fact that the attacks were mounted within a simulation environment, thus in an ideal condition for an attacker, both in terms of sampling rate accuracy and absence of noise.

7 Conclusions

In this paper we introduced a simulation-based methodology for evaluating the resistance of cryptographic circuits to power analysis attacks. We used our methodology to evaluate the MCML technology as a possible counter measure against Side Channel Attacks based on Power Analysis, and demonstrated the robustness of MCML against the SPA and against the powerful variant of DPA based on correlation.

Contrary to previous papers on this subject, we did not argue robustness just qualitatively, but with real attacks. Furthermore, since our approach is based on SPICE-level simulations, it does not rely on the manufacturing of prototypes, which allows a more direct experimental study of Power Analysis-resistance.

Our results show that the power traces obtained by simulating two full cores, implementing the AES algorithm and realized in MCML, are very difficult to attack, as opposed to a CMOS implementation for which the same attacks were always successful.

References

[AAE02] Anis, M., Allam, M., Elmasry, M.: Impact of technology scaling on CMOS logic styles. Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on [see also Circuits and Systems II: Express Briefs, IEEE Transactions on] 49(8), 577–588 (2000)

[BCO04] Brier, É., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)

[BGI+08] Badel, S., Guleyupoglu, E., Inac, O., Martinez, A.P., Vietti, P., Gurkaynak, F., Leblebici, Y.: A Generic Standard Cell Design Methodology for Differential Circuit Styles. In: Design Automation and Test in Europe 2008, pp. 843–848 (2008)

[BGLT04] Bucci, M., Guglielmo, M., Luzzi, R., Trifiletti, A.: A Power Consumption Randomization Countermeasure for DPA-Resistant Cryptographic Processors. In: Macii, E., Paliouras, V., Koufopavlou, O. (eds.) PATMOS 2004. LNCS, vol. 3254, pp. 481–490. Springer, Heidelberg (2004)

[BS97] Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (1997)

[GR99] Gonzalez, J.L., Rubio, A.: Low delta-I noise CMOS circuits based on differential logic and current limiters. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on] 46(7), 872–876 (1999)

[IoSTN01] National Institute of Standards and Technology (NIST). Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197 (November 2001)

[KJJ99] Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)

[Koc96] Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)

[MKA92] Maskai, S.R., Kiaei, S., Allstot, D.J.: Synthesis techniques for CMOS folded source-coupled logic circuits. IEEE Journal of Solid-State Circuits 27(8), 1157–1167 (1992)

[MOP07] Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Advances in Information Security. Springer, Heidelberg (2007)

[RBE+07] Regazzoni, F., Badel, S., Eisenbarth, T., Großschädl, J., Poschmann, A., Toprak, Z., Macchetti, M., Pozzi, L., Paar, C., Leblebici, Y., Ienne, P.: A Simulation-Based Methodology for Evaluating the DPA-Resistance of Cryptographic Functional Units with Application to CMOS and MCML Technologies. In: International Symposium on Systems, Architectures, Modeling and Simulation, SAMOS VII (2007)

[TAV02] Tiri, K., Akmal, M., Verbauwhede, I.M.: A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In: Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC 2002), September 2002, pp. 403–406. University of Bologna, Bologna (2002)

[TAY+05] Toprak, Z., Verma, A., Leblebici, Y., Ienne, P., Paar, C.: Design of Low-Power DPA-Resistant Cryptographic Functional Units. In: Workshop on Cryptographic Advances in Secure Hardware (2005)

[TV03] Tiri, K., Verbauwhede, I.: Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In: Walter, C.D., Koç, Ç.K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 125–136. Springer, Heidelberg (2003)