Practical DPA Attacks on MDPL

Elke De Mulder, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede

K.U. Leuven, ESAT/SCD-COSIC and IBBT
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
{elke.demulder,benedikt.gierlichs,bart.preneel,ingrid.verbauwhede}@esat.kuleuven.be

Abstract. MDPL is a masked logic style that unites principles of dual-rail precharge as well as masked logic to achieve resistance against differential power analysis attacks. MDPL has received much attention and numerous papers discussing the security provided by MDPL as well as its weaknesses have been published. As a matter of fact, most of these papers are purely theoretical or provide evidence based on simulations. At present, it is unclear to what extent these concepts affect the security provided by MDPL in practice. We fill this gap and present results of an extensive case study of attacks against an MDPL prototype chip. We demonstrate successful DPA attacks and show that MDPL implementations, resistant to standard DPA attacks, can be broken in practice. Further, we show that the underlying concept of the folding attack, i.e. analysis of probability densities, indeed exposes MDPL's greatest weakness: the masking renders the circuit more vulnerable to attacks than a circuit with a fixed mask. In addition, our analysis leads to novel insights into the power consumption properties of MDPL in real silicon.

Keywords: Differential Side Channel Analysis, Masked Dual-rail Pre-charge Logic, MDPL

1 Introduction

Embedded cryptographic devices become increasingly pervasive. The fact that a (malicious) user has physical access to a device lead to a new class of attacks against cryptosystems, that do not exploit weaknesses in cryptographic algorithms but in their implementations. Since the discovery of side-channel leakage, a considerable amount of research has been performed on techniques to retrieve secret information of side-channel signals as well as on countermeasures. We restrict our attention to one specific type of side-channel leakage, namely the power consumption. Power analysis attacks exploit the relation between the power consumption of a device and the data it is processing. As this relation is not always straightforward, several statistical techniques to extract the information have been proposed. The most prominent attack methodologies are Differential Power Analysis [8] and Correlation Power Analysis [2].

The research on side-channel attack countermeasures got momentum thanks to the growing market of embedded devices and the need to have them secured. Amongst the first ones were noise generators [10], masking at the algorithmic level [1, 11] and random process interrupts [4]. These countermeasures do not attempt to eliminate the source of the leakage, but rather to thwart its exploitation.

Later, commercial and scientific research started to address the problem at its root: the logic gate level.

Over the past years, numerous logic styles have been proposed to deal with leakage at the gate level. They can be grouped in three main categories: i) masked logic (single-rail) which is difficult to protect from glitches, ii) dual-rail pre-charge logic which requires custom routing to balance the loads of complementary wire pairs, and iii) masked dual-rail pre-charge logic, which is a combination of the two former. In this paper we focus on one specific logic style from the last category, though our findings may affect logic styles with similar constructions as well.

Masked Dual-rail Pre-charge Logic (MDPL) was published at CHES in 2005 by Popp et al. [12]. It follows straight and simple design principles in order to eliminate the exploitable side-channel leakage of logic gates. One year later, at CHES in 2006, Suzuki and Saeki described a systematic weakness of MDPL, known as the early propagation effect (EPE) [6]. Another year later, at CHES in 2007, the authors of MDPL presented results of power analysis experiments based on an MDPL prototype chip. They confirmed the EPE in practice but pointed out that highly regular hardware designs seem unaffected. At the same conference, Tiri and Schaumont announced a new attack [15] known as the folding attack and showed that (in theory) dual-rail pre-charge logic and masking do not add up to a higher level of protection.

As a matter of fact, most of the papers discussing the security of MDPL are purely theoretical or provide evidence based on simulations. At present, it is unclear to what extent these concepts affect the security provided by MDPL in practice. We fill this gap and explore the level of protection provided by MDPL in praxis. We expose an MDPL prototype chip to a series of standard and particularly crafted power analysis attacks. Our main results are successful and doubtless attacks as well as novel insights into the power consumption properties of MDPL in real silicon. Our most remarkable observation is that masking renders the circuit more vulnerable to attacks than an unmasked circuit.

The paper is organized as follows. We recapitulate the main properties of MDPL in Sect. 2 and briefly recall the different attacks that have been performed so far. As this paper describes practical attacks on MDPL, we describe the measurements and measurement setup in Sect. 3. Next, we describe attack results on a reference core implemented in sCMOS and on an MDPL core with a fixed mask in Sect. 4. The main part of our contribution is Sect. 5, where we analyze an MDPL core with active masking and present our attack results. We conclude our work in Sect. 6.

2 MDPL and Known Weaknesses

MDPL combines the ideas of Wave Dynamic Differential Logic (WDDL) [19] and Random Switching Logic (RSL) [17]. The former is a dual-rail pre-charge logic style, designed to consume a constant amount of dynamic power with respect to data that is handled, but requiring a custom routing step to achieve this goal.

The latter uses a mask bit to randomize the data processed by internal nodes, ensuring that the power consumption is uncorrelated to predictable values. Note that the randomness is determined by the quality of the (pseudo) random number generator.

By combining the concepts of dual-rail pre-charge and masked logic, the authors of MDPL aimed at getting rid of two problems at once. Using dual-rail pre-charge logic in combination with a mask bit, the authors wanted to avoid the tedious balancing of differential wire pairs. The imbalances would be accepted but randomized and thus not exploitable. The combination of dual-rail pre-charge logic and the use of monotonic increasing positive functions guarantees that no glitches, which render masking useless, will occur.

In an MDPL circuit, all logic gates are masked with a mask bit m and its complement $\overline{m}$. All MDPL flip-flops are fed with masks $m \oplus m_n$ and $\overline{m \oplus m_n}$ (where m_n is the mask of the next clock cycle) to entail that the masks are switched correctly from one cycle to the next. MDPL works in two phases: when the clock is high, the pre-charge wave is started by the MDPL flip-flops and travels gradually through the circuit bringing all differential pairs to a $(0, 0)$-state. At the same time, also the signal trees for all mask signals are pre-charged to $(0, 0)$. In the next phase, the evaluation phase, when the clock is low, the flip-flops output the internally stored values and all combinational logic gates evaluate to either $(0, 1)$ or $(1, 0)$ depending on the input data and the masks.

2.1 Early Propagation

Suzuki and Saeki showed that MDPL suffers from a systematic weakness known as early propagation effect (EPE) [6]. If inputs to a combinational gate have different delay times, the MDPL gate will leak side-channel information because the evaluation of the output does not wait until all inputs have arrived. This can result in a transient, data dependent, and mask independent leakage. Suzuki and Saeki explained the theory behind the EPE in MDPL gates and investigated the leakage of different delay scenarios with the aid of an FPGA implementation of 32 MDPL AND gates. Popp et al. investigated the same deficiency in [13] by analyzing the leakage of an 8051 microcontroller implemented on a MDPL core while it executes a MOV operation. The DPA traces showed severe leakage. Oddly enough, the likewise analyzed AES coprocessor implemented in MDPL did not show the same leakage. This phenomenon was attributed to the different implementation procedures. Popp et al. explained that the microcontroller implementation leaked because it is an irregular design, which provokes the EPE, while the AES coprocessor is based on a highly regular design.

2.2 The Folding Attack

A folding attack as described by Schaumont and Tiri in [18, 15] exploits the fact that a random mask bit switches the circuit between two complementary states with different power consumption profiles. They explain how a single mask bit

influences the power consumption in a binary way. In a masked dual-rail pre-charge circuit, the mask bit decides which of the complementary signal trees propagates the correct values. If the complementary signal trees have unbalanced loads, they have distinct power consumption profiles. When constructing the probability density function (PDF) of the mean power consumption of a masked circuit in one evaluation or pre-charge phase of a single clock cycle, these two profiles show up as two symmetric and distinguishable distributions. Tiri and Schaumont clarified that these distributions are directly related to the values of the mask bit. Given this fact, an adversary can construct the PDF, fold the left area on top of the right area, which cancels the effect of the mask, and perform a standard DPA attack. Note that this approach would not succeed if the dual-rails were perfectly balanced because the two distributions would perfectly match. Tiri and Schaumont confirmed their theory with cycle accurate weighted toggle count simulations. We note that the folding attack is equivalent to the zero-offset second order DPA attack by Waddle and Wagner [20].

3 Measurement Setup and Measurements

Our goal is to investigate the security provided by MDPL in real-world experiments using a prototype chip. As for all empirical studies, experimental settings are important and we thus describe our setup in detail.

Our experimental platform is a prototype chip that consists of an Intel 8051-compatible microcontroller and an AES-128 cryptographic co-processor in 0.13 µm technology. These two components are implemented in several cores using several DPA-resistant logic styles and standard CMOS logic (sCMOS). The chip further comprises a pseudo random number generator (PRNG) that can be used to provide random bits to the cores implemented in masked logic styles. We focus our analysis on the AES-coprocessors in the cores implemented in sCMOS and MDPL, the former is supposed to serve as a reference.

The AES implementation follows the highly regular architecture described in [14] (see Fig. 9 in the Appendix). The AES encryption operation is included in the data unit of the core, next to it is the roundkey generation unit. The AES state is represented by 16 data cells $C_{i,j}$ with $i, j \in [0, 1, 2, 3]$ in a 4×4 matrix outline. Each data cell can perform the bitwise-xor addition of the roundkey. Below this matrix is a row of four implementations of the AES Sbox, which are all one-stage pipelined implementations, such as the one described in [7]. On the left side of the matrix is an implementation of the MixColumn operation.

Encryption works as follows: the plaintext bytes are shifted into the data unit from right to left, four bytes (one column) at a time. After simultaneous roundkey addition in all data cells, the rows are rotated vertically through the Sboxes and bytes within the rows are shifted horizontally according to the ShiftRows transformation. After 5 clock cycles all bytes have been processed by SubBytes and Shiftrows. Next, the columns of the matrix are rotated horizontally through the MixColumns implementation.

All experiments we describe in this paper focus on the power consumption of data cells $C_{0,0}$ and $C_{0,1}$ that store the Sbox output values related to plaintext bytes 1 and 5 in the first round of AES. We choose these two plaintext bytes uniformly at random, while keeping the other 14 plaintext bytes constant to reduce algorithmic noise. This particular choice reflects a chosen plaintext attack scenario. Later in the paper, before drawing conclusions, we show that the same results could have been achieved in a known plaintext setting were all plaintext bytes are chosen at random.

Concerning the sCMOS core, cells in the left column of the matrix as the ones we target are easier to attack than others, because they have to drive a higher load (each data cell is connected to four 1/4 MixedColumn cells). In MDPL this particularity of the architecture vanishes because not the absolute load of a cell is important but only the load difference of complementary wire pairs.

The measurement setup consists mainly of a printed circuit board, which contains the chip and an on board measurement circuit. The measurement circuit exploits an active circuit as introduced by Bucci *et al.* in [3]. The clock signal is provided by a waveform generator and the power traces are recorded with an oscilloscope with 1GHz bandwidth and 8bit resolution at a sampling rate of 2GS/s.

We had to take special care to ensure that the measurement does not clip while at the same time using as much of the vertical (amplitude) range of the oscilloscope as possible to allow a good sampling resolution. Clipping causes annoying artifacts in the histograms and blurs the information.

4 Experiments and Results I: Warming Up

All attacks that we conducted used a distance of means test [8] and a correlation test [5] in combination with each of the following prediction functions: the Hamming weight of each single bit stored in cells $C_{0,0}$ and $C_{0,1}$ after the Sbox computation, the Hamming weights of the bytes stored in cells $C_{0,0}$ and $C_{0,1}$, the Hamming distances between the single bits stored in cells $C_{0,0}$ and $C_{0,1}$, and the Hamming distance of the bytes in the two cells. However, we report only the most meaningful results, i.e. the combinations of prediction function and statistical test that lead to the clearest results. When using Hamming distance prediction functions we assumed that the key byte associated to cell $C_{0,1}$ is known and tried to reveal the other key byte, which decreases the computational load without affecting the generality of the result. It turned out that both statistical tests perform very similar in our attacks and that in all cases both or none of them would reveal the key. We decided to report the results of the correlation test, because the coefficient is normalized and hence to some extend interpretable.

Before diving into the analysis of the MDPL core with random masks, we performed attacks on the sCMOS core and on the MDPL core with fixed mask values.

4.1 Attacks on the sCMOS Core

We first attacked the sCMOS core to have a reference in terms of difficulty that we can compare the attacks against MDPL to. For this attack we used a set of 5000 measurements. As expected, attacks with Hamming distance prediction functions worked best as these relate to bit-flips. Oddly enough, we observed that not all bits leak similarly when flipping. Figure 1 shows the result of a correlation DPA attack using the HD of the MSB in cells $C_{0,0}$ and $C_{0,1}$ as prediction function.

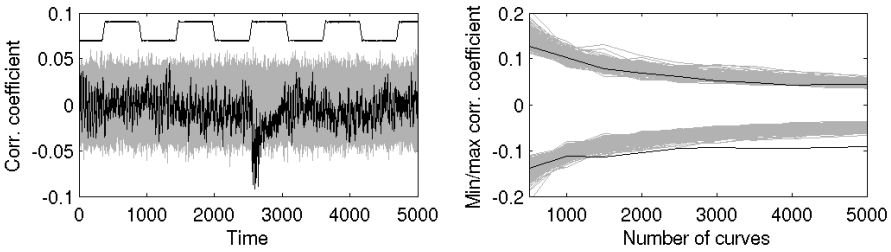

Fig. 1. DPA results for sCMOS, corr. attack with prediction of MSB of Byte1 $\oplus$ Byte5; left: corr. traces for all key hypotheses using 5000 measurements; right: evolution of min and max corr. per key hypothesis over number of measurements.

On the left side of the figure we show the correlation traces for all key hypotheses when using 5000 measurements. On the right side of the figure we show how the maximum and the minimum correlation coefficient for each key hypothesis (taken from the overall time interval) evolve over an increasing number of measurements. The traces for the correct hypothesis are plotted in black, all other are plotted in gray. Note that the DPA peak appears in the clock cycle when the data in cell $C_{0,1}$ is shifted to the left into cell $C_{0,0}$ at a time index about 2600.

4.2 MDPL with Fixed Masks

Next we attacked the MDPL core with the mask value being permanently fixed to 0. In this setting, MDPL is dual rail pre-charge logic with unbalanced routing of the complementary wire pairs and therefore vulnerable to DPA attacks. One can expect that the outcome of an attack mostly depends on measurement precision and the number of measurements, as the exploitable imbalance between complementary wires is tiny. For our attack we obtained a set of 400 000 measurements. We obtained the best results, shown in Fig. 2, when predicting the HW of the MSB of $C_{0,0}$ right after the Sbox computation. Since the attack against the sCMOS core worked best when attacking the HD on the same bit, we assume that the net carrying this bit is somehow particularly difficult to route. Note that, as before, a clear DPA peak appears at a time index of about 2600. This time the peak appears at a falling clock edge because MDPL evaluates at falling clock edges.

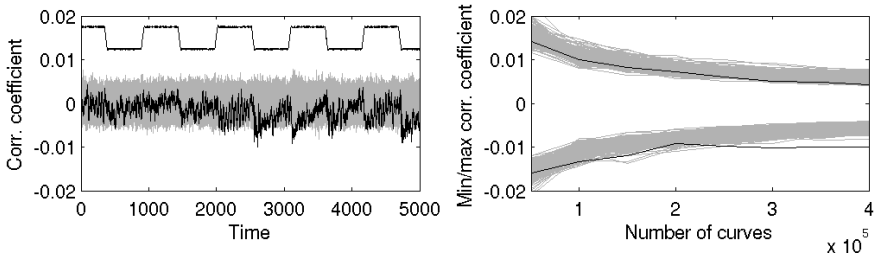

Fig. 2. DPA results for MDPL with fixed mask, corr. attack with prediction of MSB of Byte1; left: corr. traces for all key hypotheses using 400000 measurements; right: evolution of min and max corr. per key hypothesis over number of measurements.

5 Experiments and Results II: Real Stuff

For the next experiments we made sure that the PRNG that generates the mask bits for MDPL is seeded, initialized and started up correctly. We obtained a set of 1.2 million measurements.

5.1 Standard DPA against MDPL

In a first attempt we simply tried a "brute-force" DPA attack. Theoretically, MDPL should withstand standard DPA attacks independent of the statistical test, prediction function or number of measurements used. As mentioned earlier, the EPE may open a security hole but previous work indicated that the highly regular design of the AES co-processor prevents the EPE [13].

Figure 3 shows the result of an attack using the HW of byte 1 as prediction. We can see local DPA peaks near the rising clock edges at about time indexes 1000 and 2000.

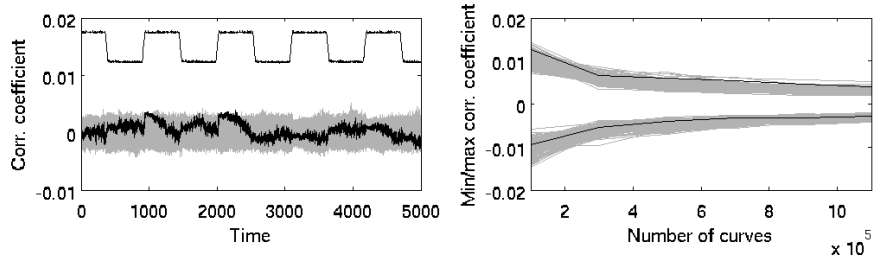

Fig. 3. DPA results for MDPL with random mask, corr. attack with prediction of HW of Byte1; left: corr. traces for all key hypotheses using 1.2M measurements; right: evolution of min and max corr. per key hypothesis over number of measurements.

However, as the plot on the right side of Fig. 3 shows, these peaks do not stand out with respect of the overall timeframe. One could speculate whether using more measurements would lead to unambiguous results, but we consider this attack not successful.

5.2 Features in Histograms of MDPL Consumption

In order to perform a folding attack, the adversary has to make histograms of the measurements. In [18, 15] the attack was performed based on simulations done in GEZEL [16]. The simulation provides toggle counts of 0 to 1 transitions that replace real measurements. The resulting histograms of the "power consumption" of the simulated MDPL circuit showed two distinct symmetric distributions. In order to mimic the toggle count with physical power measurements, we need to reduce the parts of each measurement trace that are associated with either of the two phases in every clock cycle to one value. We decided to represent the toggle count for each pre-charge and evaluation phase by the empirical mean of the power consumption during that period.

The resulting histograms of an evaluation phase and a pre-charge phase are shown in the first row of Fig. 4. We used 50 000 measurements to generate these histograms. At first glance, the histograms of the pre-charge phase follow the theory of Tiri and Schaumont, but the histogram of the evaluation phase looks remarkably different than what is expected. To reduce the noise in the histograms, we decided to take only particularly meaningful points in time into account and to represent the toggle count by the empirical mean of the power consumption at those points in time. To identify this interesting part of the power traces, we calculated the sum of the absolute differences of the measurements. The result is shown in the second row of Fig. 4. The solid black line is the sum of the absolute difference per time instant computed from 50 000 measurements. The grey lines indicate the exact time span we used to generate new histograms and the dashed black line is a power trace for reference. Essentially, we skip the transient oscillations in the beginning and the fading out time at the end of each phase. The new histograms based on the selected time span are shown in the third row of Fig. 4. Each of the phases show four distinct distributions in the histograms, although less visible in the pre-charge phase, very explicit in the evaluation phase. In the pre-charge phase the areas under the four distributions are equal. In the evaluation phase the first and last distribution contain each $\frac{1}{8}$-th of the measurements, the two in the middle each $\frac{3}{8}$-th.

The four distinct distributions are due to the masking. MDPL flip-flops are fed with a different mask signal than the combinational MDPL logic, namely $m \oplus m_n$ instead of m. The combination of m and $m \oplus m_n$ puts the circuit (more precisely the mask signal trees) in four different states. Tiri and Schaumont reported on only two of them because they did not take the MDPL flip-flops and thus the signal tree $m \oplus m_n$ into account. This results in a fourfold appearance of the distribution that one could expect for an unmasked single-rail circuit. We validated this idea with a GEZEL simulation, see Appendix B. Note that the power consumption profiles of the MDPL circuit with fixed mask show only one distribution.

Herding Measurements. Interestingly we observed that there is a strong correlation between the four distributions that occur during each pre-charge and

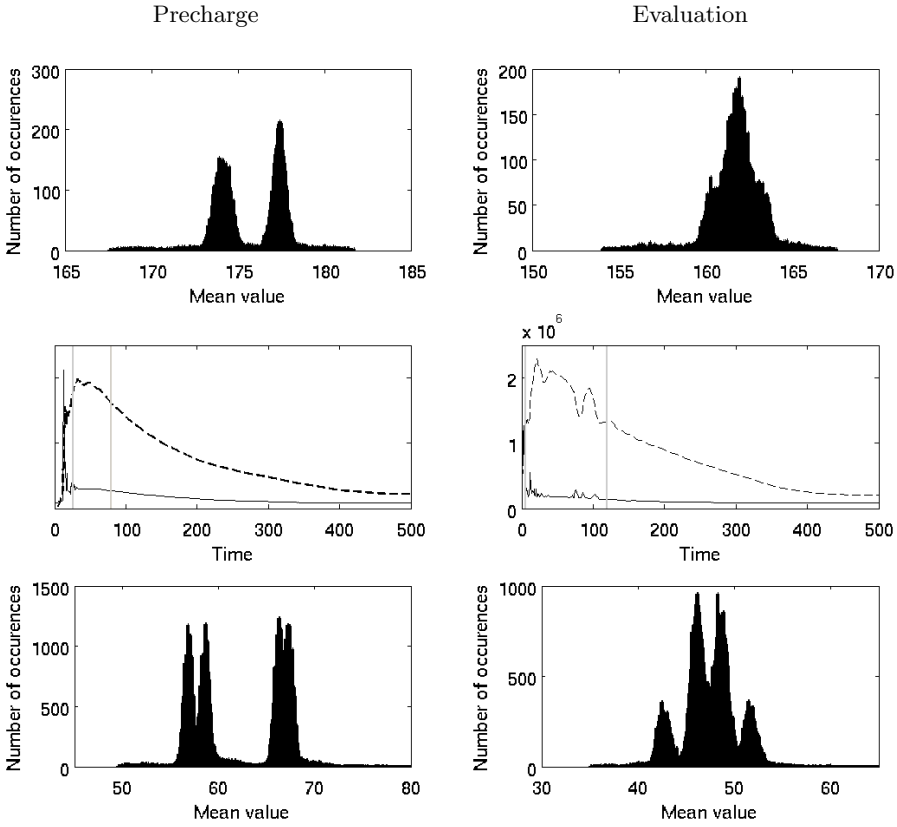

Fig. 4. The evolution of the histograms of a pre-charge phase in the left column and an evaluation phase on the right for 50 000 measurements. The first row shows the histograms for the mean of the complete clock cycle. The second row represents the time intervals chosen to extract the final histograms which are shown in the third row.

evaluation phase. We begin with a pre-charge phase and assign each out of 50 000 measurements to one out of four possible groups according to its membership to one of the four visible distributions. This yields a partitioning in four groups (Pr_A,Pr_B,Pr_C,Pr_D) of equal size. Next, we consider the following evaluation phase and repeat the partitioning which yields four groups (Ev_A,Ev_B,Ev_C,Ev_D) with relative cardinalities 1,3,3,1.

The table on the left side of Fig. 5 shows how the 50 000 measurements transfer between groups Pr_A,Pr_B,Pr_C,Pr_D and Ev_A,Ev_B,Ev_C,Ev_D when the circuit switches from pre-charge phase to evaluation phase. We repeated the same analysis for a transition from evaluation to pre-charge phase and show the transitions between groups Ev_A,Ev_B,Ev_C,Ev_D and Pr_A,Pr_B,Pr_C,Pr_D in the table on the right side. The numbers are given as percentages and we note that the numbers in one row do not necessarily add up to 100% as outliners are not counted.

Rougly said, the groups from the pre-charge phase split into two equally sized parts when making the transition to the evaluation phase. In the other case, when

↱	Ev_A	Ev_B	Ev_C	Ev_D
Pr_A	47	49	2	0
Pr_B	3	49	46	0
Pr_C	0	42	53	4
Pr_D	0	5	50	43

↱	Pr_A	Pr_B	Pr_C	Pr_D
Ev_A	1	2	82	7
Ev_B	30	33	28	5
Ev_C	31	32	5	26
Ev_D	2	3	11	77

Fig. 5. Transition of measurements between groups from pre-charge to evalution phase in the left tabular, transition of measurements between groups from evaluation to pre-charge phase in the right tabular.

a transition from evaluation phase to pre-charge phase is made, the measurements from Ev_A and Ev_D are completely transfered to Pr_C and Pr_D, respectively, while the two larger groups Ev_B and Ev_C are spread equally over three groups of the pre-charge each. A reasonable explanation for this observation would be some analog effect that, in addition to m and $m \oplus m_n$, has a systematic impact on the power consumption. One can think of the EPE, but this is only speculation.

5.3 Subset Attacks on MDPL

During our research it became clear that we can easily assign each single measurement to one out of four distinct groups for each pre-charge and evaluation phase. Instead of folding directly, we first followed a different approach. We assumed that each of the four distributions represents a particular state of the masks m and $m \oplus m_n$. Thus, selecting a subset of measurements that all belong to the same distribution should yield a strong bias of the masks.

For each pre-charge and evaluation phase, we assigned the 1.2 million measurements to one of four distinct groups according to their distribution membership based on the histogram for that particular phase. We denote the groups A to D in the order of their appearance in the histograms from left to right. Next, we mounted DPA attacks using the original power traces as follows: i) depending on the time index use the grouping previously determined for that particular pre-charge or evaluation phase ii) evaluate the prediction functions as usual iii) attack the four groups of measurements separately.

Figure 6 shows the result of this approach when using the prediction function Hamming weight of bit 2 of byte 1. The plot on the top, left hand side shows the DPA results based on measurements that were assigned to group A for the point in time considered. Next to it, on the top, right hand side, the plot shows the result for measurements assigned to group B. The same for groups C and D in the second row. We can see that all four attacks lead to peaks that reveal the correct key, in particular the attacks against groups B and C. These two attacks lead to clear peaks at the beginning of the same pre-charge phase at time index 1000.

The attack launched on measurements belonging to group C yielded an unexpectedly clear DPA peak. We were interested in finding out how many measurements would be necessary to reproduce this attack. The answer can be deduced

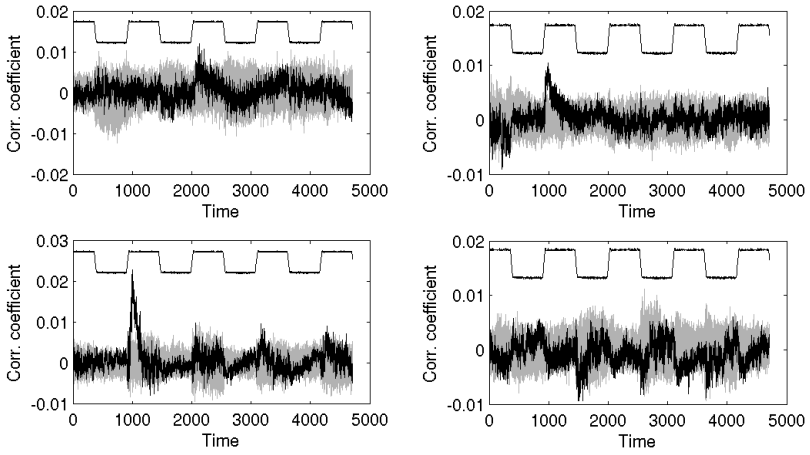

Fig. 6. DPA results for MDPL with random mask, corr. attack against HW of bit 2 of byte 1; top left: group A; top right: group B; bottom left: group C; bottom right:group D; corr. traces for all key hypotheses using 1.2M measurements.

from the plot on the right hand side of Fig. 7. According to our results, 300 000 samples should be enough for this attack to be successful.

The next step towards implementing the folding attack, is to fold the PDFs of each pre-charge and evaluation phase. It turned out that, using real measurements, the folding is not as straight-forward as described by Schaumont and Tiri. The main problem is that neither the PDFs of pre-charge phases nor the PDFs of evaluation phases are symmetric, which makes it difficult to decide where to fold. Nevertheless, we folded all PDFs once around the empirical mean, which yields PDFs with two distributions. Exposing one of them to a DPA attack resulted indeed in correlation peaks in some cases, but they were less clear than the peaks we achieved with our subset attack.

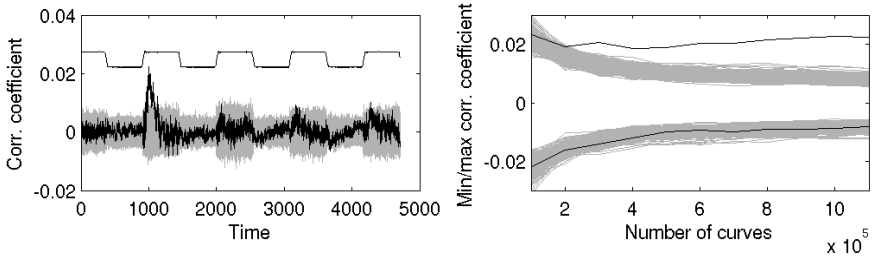

Fig. 7. DPA results for MDPL with random mask, corr. attack with prediction of HW of bit2 of byte1; left: corr. traces for all key hypotheses using 1.2M measurements; right: evolution of min and max corr. per key hypothesis over number of measurements.

Extrapolation to a Known Plaintext Scenario. We were interested in determining whether our attack could be reproduced in known plaintext scenarios where the measurements are polluted with algorithmic noise, and in quantifying how difficult the attack would be.

We obtained a set of 50 000 measurements for which all 16 plaintext bytes were randomly chosen from a uniform distribution and generated histograms for all pre-charge and evaluation phases as described in Sect. 5. Figure 8 shows exemplary histograms for a pre-charge and an evaluation phase. Again, the PDF

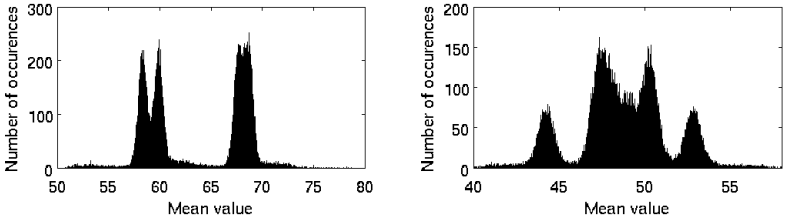

Fig. 8. The histograms in case all plaintext bytes are chosen at random.

contains four distinct distributions, though they appear slightly blurred due to the enhanced algorithmic noise. Nevertheless, it is clear that the division in four different sets can be carried out.

In order to determine how difficult our attack would be in this scenario, i.e. how many measurements would be necessary, we first compute the expected height of the correlation peak and then use this number to estimate how many measurements would be required.

The correlation peak for the correct key hypothesis in Fig. 7 converges towards a value of $\rho = 0.023$. Note that this peak is caused by partial correlation as we target a single bit while 16 bits (the 2 chosen bytes) are active. Using the formulas for partial correlation from [2] we calculate that, in the known plaintext scenario, the correlation peak decreases to $\rho' = \rho \cdot \sqrt{\frac{\frac{1}{16}}{\frac{160}{160}}} = 0.0073$ (where 160 is the number of active bits: 128 in the AES state and 32 in the four Sbox implementations).

With the formulas provided by Mangard in [9] we estimate the number of samples required for our attack to reveal the key byte with high probability ($\alpha = 0.9999$) as $\sim 520\,000$.

6 Conclusion

We presented results of an extensive case study of power analysis attacks against an MDPL prototype chip. MDPL withstands standard DPA attacks but it can be easily weakened by choosing only a subset of the available power measurements based on an analysis of the power distribution profiles. MDPL does not resist Standard DPA attacks using only subsets of the measurements. Analysis

of power probability densities indeed exposes MDPL's greatest weakness: the masking renders the circuit more vulnerable to attacks than a circuit with deactivated masking. Additionally, our analysis leads to novel insights into the power consumption properties of MDPL in real silicon.

A The AES-128 Architecture

Figure 9 shows the architecture of the AES-128 coprocessor.

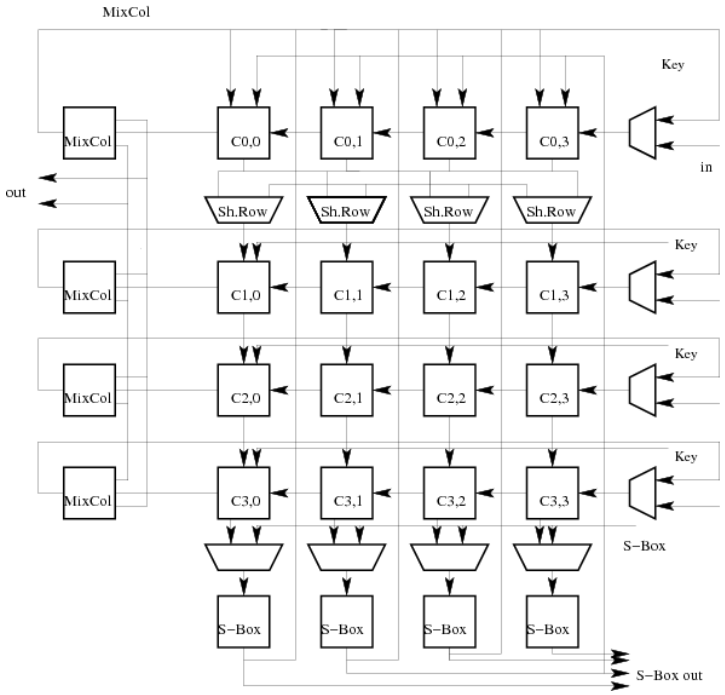

Fig. 9. Architecture of the AES-128 implementation

B Toggle Count Simulation

To test the hypothesis that m and $m \oplus m_n$ actually quadruple the original PDF instead of doubling it, a GEZEL simulation and toggle counting has been done. For the simulation we used a fully-fledged AES implementation in MDPL. The gate level model was obtained with synopsis and afterwards converted into GEZEL. The design uses a total of 22642 MDPL gates of which 393 are flip-flops. Routing imbalances were simulated with a weighted toggle count.

Note that the simulation is performed with uniform balances, although this is an unrealistic situation, the only purpose of this simulation is to check whether the combination of m and $m \oplus m_n$ really quadruples the PDF. Fig. 10 shows

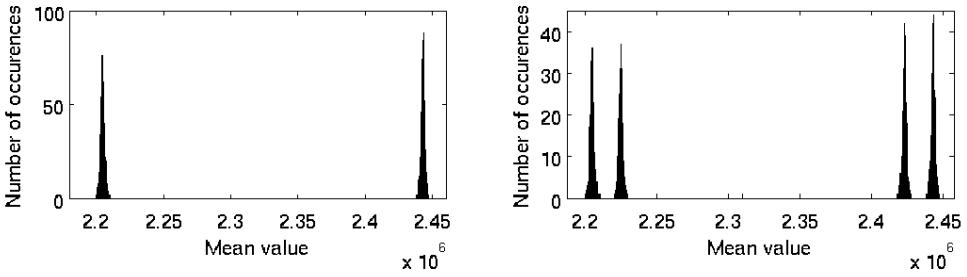

Fig. 10. The histograms of a toggle count simulation in GEZEL for the evaluation phase. The picture on the left is the situation *without* $m \oplus m_n$-tree, the picture on the right *wit*h $m \oplus m_n$-tree.

that this is indeed the case. The plot on the left shows the histogram in case the imbalance of the $m \oplus m_n$-tree and its complement is put to zero, the one on the right when the imbalance is included in the weighted toggle count.

References

1. M.L. Akkar and C. Giraud. An implementation of DES and AES, secure against some attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2001*, pages 309–318, London, UK, 2001. Springer-Verlag.
2. E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, number 3156 in Lecture Notes in Computer Science, pages 16–29. Springer-Verlag, 2004.
3. M. Bucci, L. Giancane, R. Luzzi, G. Scotti, and A. Trifiletti. Enhancing power analysis attacks against cryptographic devices. In *Circuits and Systems Symposium*, May 2006.
4. C. Clavier, J.-S. Coron, and N. Dabbous. Differential power analysis in the presence of hardware countermeasures. In *Cryptographic Hardware and Embedded Systems - CHES 2000*, pages 252–263, London, UK, 2000. Springer-Verlag.
5. J.-S. Coron, P.C. Kocher, and D. Naccache. Statistics and secret leakage. In *Financial Cryptography*, pages 157–173, 2000.
6. M. Saeki D. Suzuki. Security Evaluation of DPA Countermeasures Using Dual-Rail Precharge Logic Style. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, number 4249 in Lecture Notes in Computer Science, pages 255–269. Springer-Verlag, 2006.
7. M. Lamberger J. Wolkerstorfer, E. Oswald. An ASIC implementation of the AES sboxes. In *Topics in Cryptology - CT-RSA 2002*, volume 2271, pages 67–78. Springer, 2002.
8. P.C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.
9. S. Mangard. Hardware Countermeasures Against DPA – A Statistical Analysis of Their Effectiveness. In *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004*, pages 222–235. Springer, 2004.
10. T.S. Messerges, E.A. Dabbish, and R.H. Sloan. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.*, 51(5):541–552, 2002.
11. E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen. A side-channel analysis resistant description of the AES s-box. In H. Gilbert, editor, *Fast Software Encryption*, Lecture Notes in Computer Science, pages 413 – 423. Springer, 2005.

12. T. Popp and S. Mangard. Masked dual-rail pre-charge logic: DPA-resistance without routing constraints. In *Cryptographic Hardware and Embedded Systems - CHES 2005*, pages 172–186, 2005.
13. Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the masked logic style mdpl on a prototype chip. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *LNCS*, pages 81–94. Springer, 2007.
14. S. Dominikus S. Mangard, M. Aigner. A highly regular and scalable AES hardware architecture. In *IEEE Transactions on Computers*, volume 52 issue 4, pages 483–491. IEEE, 2003.
15. P. Schaumont and K. Tiri. Masking and dual-rail logic don't add up. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 95–106, Berlin, Heidelberg, 2007. Springer-Verlag.
16. P. Schaumont and I. Verbauwhede. Domain specific tools and methods for application in security processor design. *Design Automation for Embedded Systems*, 7:365–383(19), November 2002.
17. D. Suzuki, M. Saeki, and T. Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive. Technical report, IACR ePrint, 2004.
18. K. Tiri and P. Schaumont. Changing the odds against masked logic. In *Selected Areas in Cryptography*, pages 134–146, 2006.
19. K. Tiri and I. Verbauwhede. I.: A digital design flow for secure integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25:1197–1208, 2006.
20. J. Waddle and D. Wagner. Towards efficient second-order power analysis. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, number 3156 in Lecture Notes in Computer Science, pages 1–15. Springer-Verlag, 2004.