Successfully Attacking Masked AES Hardware Implementations *

Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald

Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria
{Stefan.Mangard, Norbert.Pramstaller, Elisabeth.Oswald}@iaik.TUGraz.at
http://www.iaik.TUGraz.at/research/sca-lab

Abstract. During the last years, several masking schemes for AES have been proposed to secure hardware implementations against DPA attacks. In order to investigate the effectiveness of these countermeasures in practice, we have designed and manufactured an ASIC. The chip features an unmasked and two masked AES-128 encryption engines that can be attacked independently.
In addition to conventional DPA attacks on the output of registers, we have also mounted attacks on the output of logic gates. Based on simulations and physical measurements we show that the unmasked and masked implementations leak side-channel information due to glitches at the output of logic gates. It turns out that masking the AES S-Boxes does not prevent DPA attacks, if glitches occur in the circuit.

Keywords: AES, ASIC, DPA, Masking, Power Analysis

1 Introduction

Power analysis attacks pose a serious threat to implementations of cryptographic algorithms. This is why there has been a lot of research during the last years to develop countermeasures. In particular, there have been quite some efforts to find methods to protect implementations of the Advanced Encryption Standard (AES) [10] against differential power analysis (DPA) attacks [7].

A commonly used approach to protect implementations of AES against DPA attacks is to randomize all intermediate results that occur during the computation of the algorithm. Usually, this is done by adding a random value to the intermediate results. This approach is called masking. The first article describing a masking scheme for AES was published by Akkar *et al.* in 2001 [2].

During the last years, several alternative masking schemes have been proposed (see [3], [6], [12], [16], and [17]). These publications focus on alternative

* This work has been supported by the Austrian Science Fund (FWF Project Number P16110) and by the European Commission under the Sixth Framework Programme (Project SCARD, Contract Number IST-2002-507270).

methods to mask the AES S-Box. All other operations of AES are linear and hence they are easy to mask. Most of the articles that have been published on masking so far, are mainly theoretical. The security or insecurity of the different masking schemes has primarily been analyzed by assuming certain power consumption characteristics of the hardware that is used to implement the schemes.

The current article is different. We have designed and manufactured a chip that features an unmasked version and two masked versions of an AES-128 encryption engine. For the masked versions we have used the approach presented by Oswald *et al.* [12] and the original approach of Akkar *et al.* [2]. We have restricted our implementation to these two masking schemes for the following reasons.

The approach presented in [3] is very similar to the one of Oswald *et al.* Both schemes are provably secure in theory and therefore we have implemented only one of them. The masking scheme proposed in [17] has not been considered because it has been shown in [1] that this scheme does not prevent standard first-order DPA attacks. The approach of Golić and Tymen [6] seems not to be suitable for hardware implementations due to its big area requirements. The approach of Trichina and Korkishko [16] is based on masking at the gate level. However, we only wanted to compare implementations of masking schemes that are applied at the algorithm level.

Our chip implementing the unmasked AES processor and the two masked versions has been designed using a 0.25 μm CMOS technology. In order to perform DPA attacks on this chip, we have built a dedicated printed circuit board (PCB) that provides easy access to the power supply of the chip.

In this article, we present and compare the results of two types of DPA attacks. The first type of DPA attacks was targeted at intermediate results of AES that are stored in registers in our AES implementations. Attacks on registers of an unmasked AES implementation have also been analyzed by Örs *et al.* in [11]. Like the attacks of Örs *et al.*, also our attacks on the unmasked implementation have been successful. As expected, the cipher key of the masked versions could not be revealed by this type of attack. The second type of DPA attacks we have performed, was targeted at intermediate results that occur only at the output of logic gates. This is an important type of attack because in typical AES hardware implementations not all intermediate results that are suitable for a DPA attack are stored in registers.

In this article, we present successful DPA attacks of this type on the unmasked as well as on the masked AES implementations of our chip. Masking does not prevent this kind of DPA attacks because of glitches that occur in the masked S-Boxes of our chip. Glitches are switching operations of logic gates that are caused by timing properties of gates and by interconnection delays.

The fact that glitches lead to a side-channel leakage of masked gates has already been shown in [8] based on SPICE simulations. Also Suzuki *et al.* [15] have recently discussed the effect of glitches on the DPA-resistance of masked circuits. The current article shows that it is actually possible to exploit the side-channel leakage of masked AES implementations in practice.

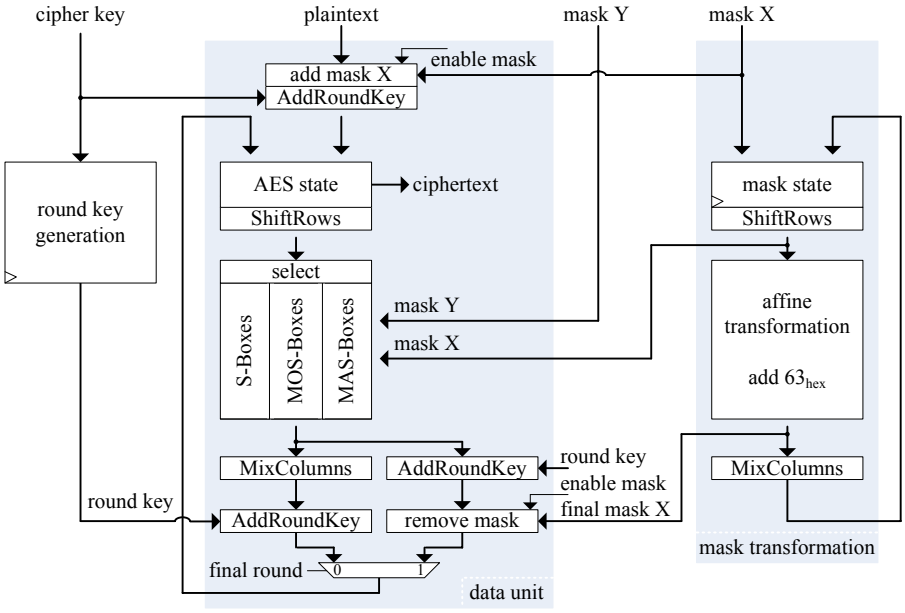

Fig. 1. Architecture of the AES chip

It is important to point out that it is not necessary to perform higher-order DPA attacks (see [9] and [18]) in order to exploit this side-channel leakage. All attacks presented in this article are first-order DPA attacks.

The remainder of this article is organized as follows: Section 2 describes the architecture of our AES chip. Results of DPA attacks that have been targeted at the output of registers are discussed in Section 3. Section 4 presents the results of the attacks on logic gates. This section provides an extensive discussion of the measurement results and it also analyzes glitches based on simulations. A summary of the results of the attacks on logic gates is presented in Section 5. Future research topics and a conclusion are stated in Section 6.

2 Architecture of the AES Chip

The architecture of our AES chip is schematically depicted in Figure 1. Based on this architecture, AES-128 encryptions can be performed in three different modes. In the first mode, an unmasked encryption is computed. The second mode performs a masked encryption based on the masking scheme proposed by Oswald *et al.*, and the third mode encrypts plaintexts based on the masking scheme proposed by Akkar *et al.*

During the design of the chip, special attention has been paid to ensure that only those parts of the chip are active that are actually needed for the selected mode—all other parts are completely disabled.

The main components of the architecture of our chip are the round-key generation, the data unit, and the mask transformation (see Figure 1). The round-key generation calculates the round keys as specified in [10]. The data unit implements all round transformations: AddRoundKey, ShiftRows, SubBytes, and MixColumns. The S-Boxes that are needed for the SubBytes transformation have been implemented separately for the unmasked mode and for the two masked modes. We refer to the masked S-Boxes as MOS-Boxes (masked as proposed in [12]) and MAS-Boxes (masked as proposed in [2]). Our architecture is based on a 32-bit datapath and therefore, four S-Boxes, four MOS-Boxes, and four MAS-Boxes are present in the design.

The mask transformation is the third main component of the architecture. It computes how the input mask (mask X) is altered by the linear transformations of the AES algorithm. Transformed mask values are required as input for the MAS-Boxes and the MOS-Boxes, as well as for the mask removal in the final round of an encryption. The multiplicative mask Y is only required for the MAS-Boxes.

3 DPA Attacks on Registers

In our architecture, the register labelled "AES state" in Figure 1 stores the AES state [10] after each round of an AES-128 encryption. If masking is enabled, this register stores the corresponding masked AES state, *i.e.* the sum of the unmasked AES state and the mask stored in the register labelled "mask state".

For the DPA attacks on this register, we assume that the attacker knows the ciphertext, *i.e.* she/he knows the content of the register after the final round of AES has been computed. In the final round, no MixColumns transformation is performed. Hence, it is possible to calculate one byte of the AES state of round nine based on one byte of the ciphertext and one byte of round key ten. We have exploited this property to successfully mount a DPA attack on the unmasked implementation of AES.

We have revealed round key number ten by attacking one byte of this key after the other. The DPA attack was done by formulating hypotheses about the number of transitions that occur at the output of the register "AES state" at the moment of time when the ciphertext is stored. The correlation between the hypotheses and the power consumption of the chip was measured using Pearson's correlation coefficient. The cipher key of the unmasked implementation was found based on $120,000$ measurements.

After the successful DPA attack on the unmasked implementation, we have performed the same attack on the masked ones. However, using the same hypotheses as in the unmasked case, it was not possible to reveal the cipher key of the masked implementations. Not even an attack based on one millon measurements was successful. This is actually an expected result. The hypotheses of the attacker do not correlate with the power consumption because the content of "AES state" register is masked. Table 1 shows a summary of our attacks on the "AES state" register.

Table 1. Summary of the attacks on the register storing the (masked) AES state

Attacked AES implementation	Number of needed measurements
S-Box	120,000
MOS-Box	not possible with 1,000,000
MAS-Box	not possible with 1,000,000

4 DPA Attacks on Logic Gates

In hardware implementations of cryptographic algorithms, many intermediate results that can be used for DPA attacks are usually not stored in registers. Our implementations for example do not store the output of the S-Box operations. We only store the AES state after each round.

In this section, we discuss DPA attacks on intermediate results that occur at the output of logic gates. Attacks of this kind cannot be conducted as easily as attacks on registers. The reason for this is that the transitions occurring at the output of logic gates are very hard to predict for an attacker. Registers switch their output only once per clock cycle. This transition of the output value leads to the power consumption that is attacked. Logic gates in CMOS circuits however, switch their output potentially several times per clock cycle—there occur glitches. This is a consequence of the timing properties of logic gates and the interconnection delays. Information about glitches in CMOS circuits can for example be found in [14].

We discuss the challenges of performing DPA attacks on logic gates based on our unmasked AES implementation in Section 4.1. DPA attacks on the masked implementations of AES are subsequently presented in Sections 4.2 and 4.3.

4.1 Attacks on the Unmasked Implementation

The output of the S-Box operation in the first round is an ideally suited target for a DPA attack on logic gates. This intermediate result is not directly stored in a register and it can be calculated based on one byte of plaintext and one byte of the cipher key. All attacks we discuss in this section are targeted at the logic gates computing this intermediate result. However, before discussing the attacks on the actual chip, we analyze and attack the power consumption characteristics of an unmasked S-Box based on simulations.

Attacking an Unmasked S-Box Based on Simulations The S-Boxes used in our architecture have been implemented as proposed by Wolkerstorfer *et al.* in [19]. A block diagram of this implementation is shown in Figure 2. In order to analyze the power consumption of S-Box 1 of our unmasked implementation, we have performed simulations based on a back-annotated netlist of this S-Box.

Simulations of this kind can be used to determine the number of transitions that occur at the nodes of the S-Box circuit upon a change of the S-Box input. Figure 3 for example shows how the output bits of the S-Box change, if the

input switches from 10_{hex} to FF_{hex}. This transition at the input leads to many transitions at the output during a time frame of more than 2 ns. As it can be seen in Figure 3, many glitches occur in our S-Box implementation.

In addition to the transitions at the output of the S-Box, there also occur many transitions at the internal nodes. In order to assess the overall power consumption of the combinational circuit implementing the S-Box, we have performed simulations for all possible input transitions ($2^8 * 2^8 = 2^{16}$ simulations). During each simulation, we have counted the number of transitions that occur at the nodes of the S-Box circuit. This counting was done based on an in-house tool that has been developed to analyze the switching activity of nodes in combinational circuits. Using the output of this tool, we have calculated the average number of transitions that occur for each of the 256 possible S-Box outputs.

Figure 4 shows the result of these simulations. The upper plot shows the average number of transitions occurring in the S-Box for each output value. The capacitive load of the wires in the S-Box do not differ significantly and hence, this transition count can be used as estimation for the actual power consumption of the S-Box. The lower plot in Figure 4 shows the Hamming weight of the output values. In many DPA attacks that have been published, hypotheses about the Hamming weight of an intermediate result have been used to perform an attack. Figure 4 however, indicates that the power consumption of our S-Box implementation is unrelated to the Hamming weight of the output value of the S-Box. The correlation between the two curves shown in Figure 4 is 0.035. Therefore, DPA attacks that are based on the Hamming weight model may not be successful. In order to verify this statement, we have performed DPA attacks based on our simulations.

For these attacks, we have first generated $100,000$ random plaintexts and we have randomly chosen a cipher key. Subsequently, we have determined the number of transitions that occur in the attacked S-Box during the encryption of the plaintexts. This number of transitions was used as estimation for the power consumption of the S-Box.

This estimated power consumption was attacked by predicting the Hamming weights of the intermediate results $i_1 \dots i_{12}$ (see Figure 2) and the Hamming weight of the S-Box output. However, none of these 13 DPA attacks was successful, *i.e.* the correct key did not lead to the highest correlation.

Subsequently, we have also performed attacks based on predicting each individual bit of the 4-bit intermediate results $i_1 \dots i_{12}$ and of the 8-bit S-Box output. Most of these 56 DPA attacks also failed. However, there were some intermediate results that lead to successful DPA attacks. For example, a DPA attack based on bit 2 of i_8 revealed the correct key based on $250,000$ measurements.

The value of this bit seems to match the power consumption of the S-Box quite well. However, we consider this property to be highly specific to our implementation and therefore we do not provide a detailed discussion about which bits lead to a successful attack and which did not. The transitions that occur in the circuit implementing the S-Box depend on many factors. The main factor is the HDL description of the S-Box. However, the transitions occurring in the

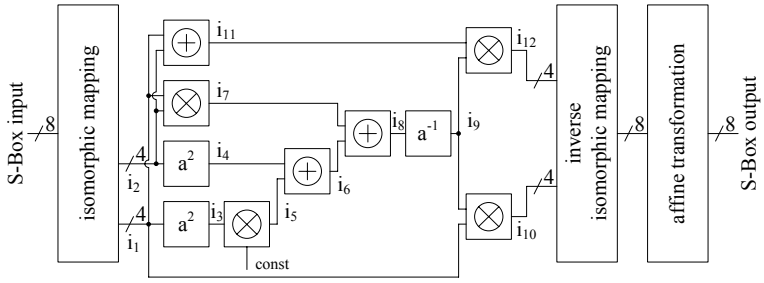

Fig. 2. Architecture of the unmasked S-Box implementation

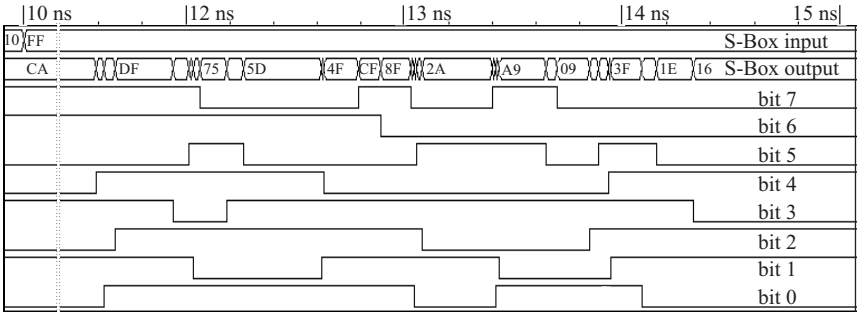

Fig. 3. The transitions that occur at the output of the unmasked S-Box, if the input changes from 10_{hex} to FF_{hex}

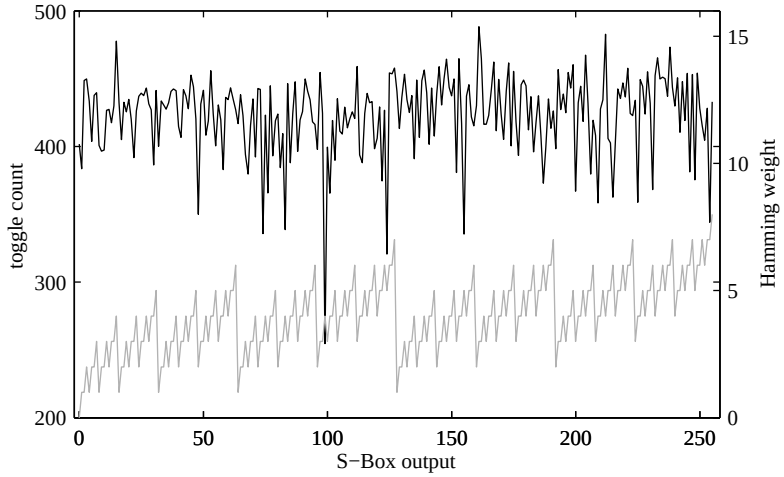

Fig. 4. The average number of transitions that occur in the unmasked S-Box for the 256 possible output values (upper plot); the lower plot shows the Hamming weight of the output values

S-Box also depend on the used cell library, the placement, the routing and of course on the way the synthesizer maps the HDL description of the S-Box to the cell library.

In our case, the power consumption characteristic of the S-Box shown in Figure 4 is correlated to bit 2 of i_8. Yet, there is no guarantee that this property is maintained if a different HDL description, synthesizer, placement tool, or cell library is used. In fact, it may turn out that in a different implementation, another bit of the intermediate results or even of the S-Box output are correlated to the power consumption of the S-Box.

The overall conclusion of our simulations is that DPA attacks based on simple power models, like the Hamming weight, work only for very few intermediate bits of our S-Box implementation. DPA attacks are only possible, if the power consumption values that are predicted by the attacker match the actual power consumption of the S-Box at least to some degree. In the following paragraphs, we empirically verify these results by performing the same attacks on the actual chip.

Attacking an Unmasked S-Box on the Actual Chip Using the setup described in Appendix A, we have encrypted one million random plaintexts with the unmasked AES implementation on our chip. During each encryption, the power consumption was recorded with a digital oscilloscope.

Based on these one million power traces, we have performed the same attacks as in the simulation. This means that we have mounted attacks based on the bits and the Hamming weights of the intermediate results $i_1 \dots i_{12}$ and of the S-Box output. The target of all our attacks was S-Box 1 in the first AES round.

We have measured a high correlation between at least one key hypothesis and the power consumption of the chip in all attacks. However, in almost all attacks it was not the correct key hypothesis that lead to the peak in the correlation trace.

Figure 5, for example, shows the result of a DPA attack based on one million measurements that was mounted on the least significant bit of the output of S-Box 1. The black trace shows the correlation we have measured for the correct key hypothesis, while the gray traces show the correlation for all other hypotheses. Some of the wrong key hypotheses lead to significant peaks. These peaks occur in the correct clock cycle, *i.e.* they occur in the clock cycle when the attacked S-Box operation is performed.

An attacker observing this result, can learn the moment of time when the S-Box operation is performed. However, the attacked byte of the cipher key is not revealed directly. This holds true for almost all attacks we have performed. In these attacks, it usually took not more than $250,000$ power measurements to determine in which clock cycle the S-Box operation is performed. However, in general the correct key could not be revealed—not even based on one million measurements.

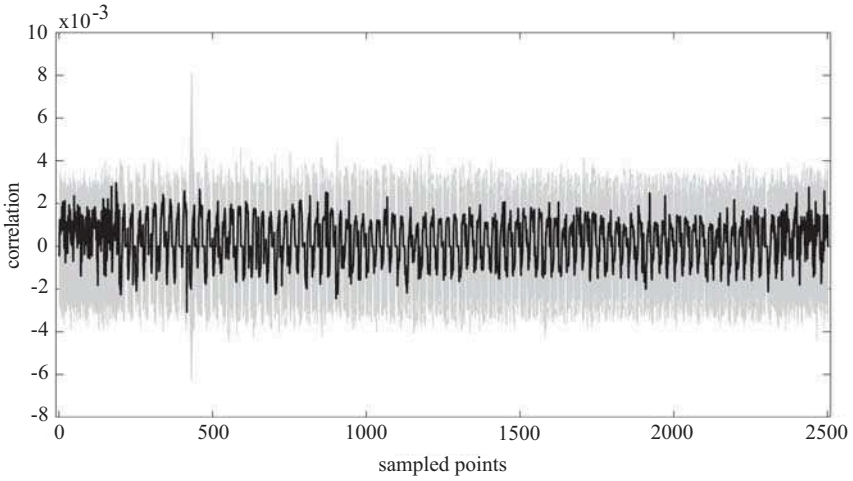

Fig. 5. The correlation for the correct key hypothesis (black) and the correlation for the wrong key hypotheses (gray) calculated based on one million measurements

The correct key could only be revealed by very few attacks. Like in the simulation, the attack on bit 2 of i_8 lead to the best results. 140,000 measurements were needed in order to successfully perform a DPA attack based on bit 2 of i_8.

In addition to the attacks on bits and Hamming weights, we have also performed a DPA attack using a more advanced power model of the S-Box. In fact, we have used the average transition count that is shown in Figure 4 as power model for our attack. This means that instead of predicting a bit of the S-Box output, we were predicting the number of transitions that occur in the S-Box. This approach is to some degree comparable to the template attacks described in [4].

The DPA attack we have performed based on predicting the number of transitions turned out to be very powerful. Only 25,000 measurements were needed in order to determine the attacked byte of the cipher key. This result confirms that it is legitimate to use the transition count as a model for the power consumption in the context of DPA attacks.

All in all, the results of the DPA attacks on the unmasked implementation have confirmed the results of our simulations. The big majority of DPA attacks using simple power models were not successful. Only those attacks using a power model that at least to a certain degree matches the actual power consumption of the S-Box, have been successful. The better the used power model was, the less measurements were needed for the attack. Our results have been confirmed by DPA attacks on all four unmasked S-Boxes of our chip.

4.2 Attacks on the Implementation of the Scheme by Oswald et al.

After the attacks on the unmasked S-Boxes of the chip, we have performed attacks on the implementation of the masked S-Boxes that are based on the approach of Oswald *et al.* (the MOS-Boxes).

Like in the unmasked case, we have first performed some simulations based on the back-annotated netlist of a MOS-Box. However, we have not performed simulated DPA attacks for the MOS-Box. Essentially, we have only derived a power model of the MOS-Box based on our simulations. This power model was created by counting the number of transitions occurring in the MOS-Box during the encryption of $100,000$ random plaintexts. For each of these encryptions, a randomly generated mask was used.

Nevertheless, it turned out that the power consumption of the MOS-Box depends on the data input of the MOS-Box. The 256 possible data inputs lead to different numbers of transitions in the MOS-Box. In fact, there were significant differences and hence, our MOS-Box implementation is leaking side-channel information.

This can be explained by glitches that occur in the MOS-Box. In [12], Oswald *et al.* prove that all intermediate results that occur in their masking scheme are independent of the plaintexts. However, this proof is done at the algorithm level. At this level, all the additional intermediate results that occur in an actual CMOS implementation due to glitches are not considered.

In order to verify that the side-channel leakage is indeed caused by glitches in our MOS-Box implementation, we have additionally performed a functional simulation of the MOS-Box circuit. For this simulation, we have used the same back-annotated netlist as in the previous simulations. However, we have ignored all the timing information and hence, no glitches occurred in the MOS-Box during the functional simulation. As expected, the number of transitions that occurred in the MOS-Box during the functional simulation did not depend on the input of the MOS-Box. During this simulation, only intermediate results occurred that have been proven to be independent of the data input of the MOS-Box.

However, unfortunately the timing characteristics of a circuit cannot be ignored in practice. The DPA attacks we have performed on the MOS-Box implementations on our chip confirm that the power consumption of the implementation with glitches actually leaks side-channel information. Like in the unmasked case, we have used one million measurements to perform the DPA attacks on our chip. The attacks we have performed first, were based on predicting individual bits and the Hamming weight of the output of MOS-Box 1 during round one. The predictions were only based on the plaintexts—the masks are unknown to the attacker.

Like in the attacks on the unmasked S-Box, it was not possible to determine byte one of the cipher key based on the attacks on the S-Box output—not even with one million measurements. However, in all attacks it was again possible to determine in which clock cycle the attacked MOS-Box operation is performed. Roughly $250,000$ measurements were needed to obtain this information.

In order to prove that it is possible to successfully attack the MOS-Box implementation with glitches, we have performed a DPA attack using the power model we had previously derived from the simulation with glitches based on the back-annotated netlist. Using this power model, it was possible to successfully attack the MOS-Box implementation based on $30,000$ measurements. Comparable results were also achieved when we targeted the remaining three MOS-Box implementations on our chip. Hence, DPA attacks on the MOS-Boxes of our chip can be performed successfully, if a reasonable power model is used.

4.3 Attacks on the Implementation of the Scheme by Akkar et al.

The implementation of the masked S-Boxes that are based on the approach of Akkar *et al.* (the MAS-Boxes) have been attacked in the same way as the MOS-Boxes in the previous subsection. This means that we have first derived a power model of a MAS-Box based on simulating its back-annotated netlist. Like before, this was done by counting the number of transitions occurring in the MAS-Box during $100,000$ masked encryptions of random plaintexts.

The number of transitions in the MAS-Box depends on the data input—just like in the case of the unmasked S-Box and the MOS-Box. It is important to point out that the observed dependency was not only caused by the zero-value problem [6] of the scheme of Akkar *et al.* Also non-zero data inputs lead to significantly different transition counts. These differences can again be explained by the glitches that occur in the circuit.

In order to verify that also our MAS-Box implementation can be attacked successfully in practice, we have measured the power consumption of our chip during one million masked encryptions. Using these power measurements, we have first performed attacks based on predicting the individual bits and the Hamming weight of the output of MAS-Box 1 in round one. The masks were again considered to be unknown to the attacker.

The attacks on the output of MAS-Box 1 have not been successful based on one million measurements. Yet, it was again possible to determine in which clock cycle the attacked MAS-Box operation was performed. Compared to the attacks on the other S-Box implementations, however, significantly more measurements were needed to obtain this information. It took $900,000$ measurements.

An intuitive argument for this big difference is that our MAS-Box implementation is significantly bigger than the S-Box or the MOS-Box implementation. Furthermore, roughly half of the operations in the MAS-Box operate on masks only. For an attacker, this part of the MAS-Box acts like a big noise engine. No glitches leading to a data-dependent power consumption can occur in this part of the MAS-Box. A data-dependent power consumption can only be caused by glitches in operations that involve the masked data.

Nevertheless, we have been able to successfully perform DPA attacks on the MAS-Boxes of our chip. Using the power model derived from the simulation of the MAS-Box, we have successfully attacked all four MAS-Boxes on our chip. For these attacks, $130,000$ measurements were needed.

Table 2. Summary of the DPA attacks on the different S-Box implementations

S-Box Implementation	Number of measurements needed to determine clock cycle	Number of measurements needed to determine key (using power model)
Unmasked S-Box	220,000	25,000
MOS-Box	250,000	30,000
MAS-Box	900,000	130,000

5 Summary of the DPA Attacks on Logic Gates

In the previous section, we have discussed different DPA attacks on the unmasked and on the two masked AES implementations on our chip. The targets of these attacks were the S-Box operations in the first round of AES.

The main result of the attacks is that all three AES implementations leak side-channel information. CMOS implementations of the masking schemes proposed in [12] and [2] leak side-channel information due to glitches. We have analyzed this fact based on simulations of back-annotated netlists of all S-Box implementations. These simulations have shown that the number of transitions that occur in the S-Boxes depends on the S-Box input. Even in the masked cases, this dependency has been observed.

In addition to the simulations, we have performed DPA attacks on an actual chip. It has turned out that the attacks on the unmasked and the masked implementations lead to similar results. DPA attacks using simple power models, such as the Hamming weight or the value of a bit, were in general not successful. However, these attacks revealed in which clock cycle the attacked S-Box operation is performed. The number of measurements that were needed to obtain this information from the different AES implementations is shown in column two of Table 2.

All AES implementations have been successfully attacked using power models that have been derived based on simulations. The number of measurements that were needed to perform these attacks are shown in column three of Table 2. The attacks obviously pose a serious threat to unmasked as well as masked CMOS implementations of AES S-Boxes.

Designers of AES hardware implementations also need to be aware of the fact that their design might be susceptible to DPA attacks using simple power models. In our experiments, an attack on bit 2 of i_8 of the unmasked S-Box was successful. Actually, there is no guarantee that the power consumption of a masked AES hardware implementation is in general uncorrelated to similar hypotheses of an attacker. Depending on the implementation, it might also happen that the power consumption of a masked AES hardware implementation is correlated to the Hamming weight of the S-Box output.

6 Future Work and Conclusion

In this article, we have shown that it is possible to mount successful first-order DPA attacks on masked ASIC implementations of AES. The attacks we have presented are based on power models that have been derived from simulations of back-annotated netlists.

However, an attacker usually does not have easy access to the back-annotated netlist of a product. This is why we are currently closely analyzing the characteristics of the side-channel leakage that is caused by glitches. Our goal is to determine whether or not there exists a general power model that can be used to attack masked AES S-Boxes. In this context, we also plan to analyze in detail why our implementation of the MOS-Boxes is not more secure than our implementation of the MAS-Boxes.

Although, these questions remain unanswered at this time, our experiments clearly show that masked hardware implementations are not as secure in practice as one might have expected. We have shown that there is a side-channel leakage of masked CMOS implementations due to glitches. We have observed this side-channel leakage in simulations based on back-annotated netlists as well as in power measurements of an ASIC implementation.

The conclusion of this article is that it is crucial for the DPA resistance of a design to think about glitches when masking schemes are implemented. Glitches should either be completely avoided [13] or the used masking scheme needs to be adapted in a way that it also works in the presence of glitches [5].

Acknowledgements

The analyzed chip with the unmasked and the two masked AES-128 encryption engines has been designed and implemented in cooperation with the Integrated Systems Laboratory at the Swiss Federal Institute of Technology Zurich. We would like to thank Frank K. Gürkaynak and Simon Häne for their generous support.

References

1. Mehdi-Laurent Akkar, Régis Bevan, and Louis Goubin. Two Power Analysis Attacks against One-Mask Methods. In *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 332–347. Springer, 2004.
2. Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES, Secure against Some Attacks. In *Cryptographic Hardware and Embedded Systems – CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.
3. Johannes Blömer, Jorge Guajardo, and Volker Krummel. Provably Secure Masking of AES. In *Selected Areas in Cryptography, 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers*, volume 3357 of *Lecture Notes in Computer Science*, pages 69–83. Springer, 2005.

4. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2535 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2003.
5. Wieland Fischer and Berndt M. Gammel. Secure Masking in the Presence of Glitches. In *Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, Scotland, August 29 - September 1, 2005, Proceedings*, Lecture Notes in Computer Science. Springer, 2005.
6. Jovan D. Golić and Christophe Tymen. Multiplicative Masking and Power Analysis of AES. In *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2535 of *Lecture Notes in Computer Science*, pages 198–212. Springer, 2003.
7. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
8. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.
9. Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In *Cryptographic Hardware and Embedded Systems – CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251. Springer, 2000.
10. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001. Available online at `http://www.itl.nist.gov/fipspubs/`.
11. Siddika Berna Örs, Frank K. Gürkaynak, Elisabeth Oswald, and Bart Preneel. Power-Analysis Attack on an ASIC AES Implementation. In *Proccedings International Conference on Information Technology - ITCC 2004, Las Vegas, USA, Proceedings*, 2004.
12. Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A Side-Channel Analysis Resistant Description of the AES S-box. In *Fast Software Encryption, 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Proceedings*, volume 3557 of *Lecture Notes in Computer Science*, Springer, 2005.
13. Thomas Popp and Stefan Mangard. Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints. In *Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, Scotland, August 29 - September 1, 2005, Proceedings*, Lecture Notes in Computer Science. Springer, 2005.
14. Jan M. Rabaey. *Digital Integrated Circuits*. Prentice Hall, 1996. ISBN 0-13-178609-1.
15. Daisuke Suzuki, Minoru Saeki, and Tetsuya Ichikawa. Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2004/346, 2004.

16. Elena Trichina and Tymur Korkishko. Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results. In *Proceedings of the Fourth Conference on the Advanced Encryption Standard (AES)*, 2004.
17. Elena Trichina, Domenico De Seta, and Lucia Germani. Simplified Adaptive Multiplicative Masking for AES. In *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2535 of *Lecture Notes in Computer Science*, pages 187–197. Springer, 2003.
18. Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In *Cryptographic Hardware and Embedded Systems – CHES 2004, 6th International Workshop, Cambridge, MA, USA, August 11-13, 2004, Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2004.
19. Johannes Wolkerstorfer, Elisabeth Oswald, and Mario Lamberger. An ASIC implementation of the AES SBoxes. In *Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference 2002, San Jose, CA, USA, February 18-22, 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 67–78. Springer, 2002.

A Measurement Setup

A dedicated printed circuit board has been developed for mounting the DPA attacks on our chip (see Figure 6). We use an FPGA as interface between a standard PC and the chip. The communication between the PC and the FPGA is performed via an optically decoupled parallel interface.

Measurements are performed as follows. First, the input data of the chip is loaded into the FPGA via the parallel port. Then, the FPGA loads the data into the chip and starts an unmasked or masked AES encryption. The chip triggers a digital oscilloscope, which records the power consumption of the chip during the encryption.

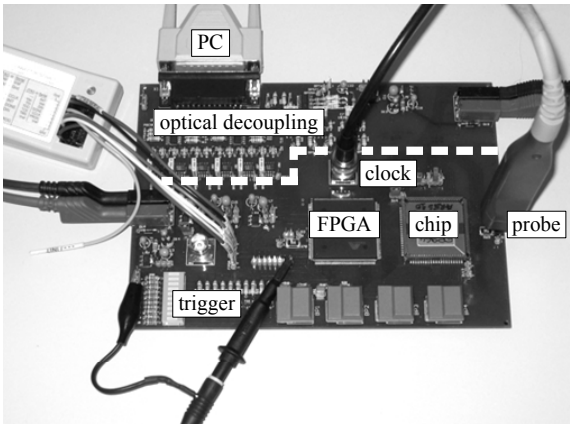

Fig. 6. Measurement setup for performing DPA attacks on the AES chip