Side-Channel Leakage of Masked CMOS Gates*

Stefan Mangard[1], Thomas Popp[1], and Berndt M. Gammel[2]

[1] Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
{Stefan.Mangard,Thomas.Popp}@iaik.at
[2] Infineon Technolgies AG
St.-Martin-Straße 76, 81541 Munich, Germany
Berndt.Gammel@infineon.com

Abstract. There are many articles and patents on the masking of logic gates. However, the existing publications assume that a masked logic gate switches its output no more than once per clock cycle. Unfortunately, this assumption usually does not hold true in practice.
In this article, we show that glitches occurring in circuits of masked gates make these circuits susceptible to classical first-order DPA attacks. Besides a thorough theoretical analysis of the DPA-resistance of masked gates in the presence of glitches, we also provide simulation results that confirm the theoretical elaborations. Glitches occur in every CMOS circuit. Consequently, the currently known masking schemes for CMOS gates do not prevent DPA attacks.

Keywords: Power Analysis, DPA, Masking, Masked Digital Circuits, Masked Logic Gates

1 Introduction

During the last years, a lot of research has been conducted on differential power-analysis (DPA) attacks [11] and on corresponding countermeasures. DPA attacks exploit the fact that the power consumption of a device executing a cryptographic algorithm is correlated to intermediate results of the algorithm. This correlation between the intermediate results and the power consumption allows an attacker to reveal the secret key that is used by a device (see [11]).

Hence, the goal of countermeasures against DPA attacks is to completely remove or at least to reduce this correlation. Essentially, there exist two approaches to achieve this goal.

The first approach is to try to make the power consumption of a device independent of the data that is processed by the device. The countermeasures that are based on this approach are usually called hardware countermeasures. Typical examples of such countermeasures are detached power supplies [19], logic

* This work has been supported by the Austrian Science Fund (FWF Project Number P16110-N04) and by the European Commission under the Sixth Framework Programme (Project SCARD, Contract Number IST-2002-507270).

styles with a data-independent power consumption [20, 21], noise generators and the insertion of random delays [4, 12]. Each of these hardware countermeasures reduces the correlation between the data that is processed by the device and the power consumption. In practice, hardware countermeasures are typically combined. This can reduce the correlation down to a level that makes DPA attacks almost impossible in practice.

The second approach to counteract DPA attacks is to randomize the intermediate results occurring in a cryptographic algorithm. The motivation behind this approach is that the power consumption of a device processing randomized intermediate results is uncorrelated to the actual intermediate results. The randomization of intermediate results is usually called masking. Masking can be applied either at the algorithm level or at the gate level.

Applying masking at the algorithm level means that an algorithm is rewritten such that all intermediate results are randomized, while the input and the output of the algorithm are identical to those of the unmasked version. There are several publications that discuss how symmetric [1, 7, 8, 24] and asymmetric ciphers [5, 15] can be rewritten this way.

The alternative to masking at the algorithm level is the usage of masked logic gates for implementations of cryptographic algorithms. This leads to circuits where no wire stores a value that is correlated to an intermediate result of the algorithm. Clearly this approach is more generic. Masking at the gate level is independent of the implemented algorithm and in principle it can even be done completely automatically, *i.e.* a program can be used to convert a digital circuit into a circuit of masked gates. Throughout this article, we refer to such circuits as masked circuits.

The theory of masking at the gate level has been analyzed recently in [9]. An implementation of an AES co-processor that is based on masking at the gate level has been presented by Trichina and Korkishko in [22, 23]. Additionally, there exist several patents on masking at the gate level (see for example [10], [13] and [14]).

However, an important issue of masking at the gate level has not been considered until now. The security analyses that have been conducted so far assume that each gate in a masked circuit switches no more than once per clock cycle. However, this assumption does not hold true in general. The input signals of a gate in a digital circuit usually do not arrive at the same time. Therefore, the output of a gate potentially switches several times during one clock cycle.

The transitions at the output of a gate that occur before the gate switches to the correct output are called *glitches*. The fact that glitches occur in digital circuits is well known and it is extensively discussed in the literature on VLSI design (see for example [17]). Glitches contribute significantly to the power consumption of CMOS circuits and hence, they are very relevant for the DPA-resistance of these circuits.

In this article, we analyze the effect of glitches on the DPA-resistance of masked gates. In fact we show that several masked CMOS implementations of nonlinear gates, such as AND and OR gates, are not resistant to DPA attacks.

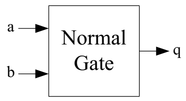

Fig. 1. The inputs and the output of a normal gate.

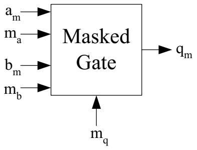

Fig. 2. The inputs and the output of a masked gate.

These implementations are susceptible to classical first-order DPA attacks. We show this fact theoretically and we also provide attack results based on SPICE simulations.

This article is organized as follows: Section 2 introduces the concept of masking at the gate level and discusses existing publications and patents on this countermeasure. In Section 3, we perform a thorough theoretical analysis of the DPA-resistance of masked gates in the presence of glitches. Furthermore, we discuss the causes of glitches and elaborate on the effort that is necessary to prevent them. Section 4 presents simulation results of DPA attacks that have been conducted on implementations of masked gates as they have been proposed in [14] and [22, 23]. We show that both approaches lead to gates that are susceptible to DPA attacks in practice. The conclusions of our findings are presented in Section 5.

2 Masking at the Gate Level

The basic idea of masking at the gate level is to represent each value a occurring in a circuit by two values a_m and m_a. m_a is a random mask that is statistically independent of a and uniformly distributed. The masked value a_m is calculated by adding a and m_a modulo two: $a_m = a \oplus m_a$.

In a masked digital circuit, logic gates take the tuple (a_m, m_a) instead of a as input. In fact, all inputs and the output of every logic gate are masked. The inputs and the output of a gate in a normal digital circuit are shown in Figure 1, while the inputs and the output of a gate in a masked digital circuit are shown in Figure 2.

In a normal digital circuit, a gate with two inputs calculates the output q based on the inputs a and b: $q = f(a, b)$. In a masked circuit, the inputs as well as the output are masked. This means that $a_m = a \oplus m_a$, $b_m = b \oplus m_b$ and $q_m = q \oplus m_q$, where m_a, m_b and m_q are randomly generated masks. The masked gate calculates the output q_m of the gate based on the inputs a_m, m_a, b_m, m_b and m_q: $q_m = \tilde{f}(a_m, m_a, b_m, m_b, m_q)$.

For the sake of readability, we only discuss gates with two masked inputs and one masked output. However, this restriction can be done without loss of generality. Our results also hold true for more complex gates. Another restriction we make in this article is that we only analyze masked circuits where one data bit is masked with one mask bit. We do not consider the general case where a value a is masked with several masks: $a = a_m \oplus m_1 \oplus m_2 \oplus \ldots \oplus m_n$.

Using more than one mask bit for one data bit, as for example proposed in [9], is not very practical. Already in the case where only one mask bit is required for each data bit, the generation and the distribution of the mask bits are challenging tasks for the designers of a circuit.

In practical (commercial) applications, area and power restrictions usually rule out the generation of a fresh mask for every data bit in every clock cycle. This approach would essentially mean that for every data bit, one (pseudo-) random number generator would be required. In practical applications, designers have to reuse the same mask for several data signals or they have to use the same mask for several clock cycles.

However, in the context of this article we do not elaborate on the issue of how masks can be generated or distributed. We simply use the best-case assumption concerning the generation and distribution of masks, *i.e.* fresh masks m_a, m_b and m_q can be generated for every gate in every clock cycle. We show that even using this ideal assumption, glitches in masked digital circuits make these circuits susceptible to DPA attacks.

2.1 The Theory Behind Masked Gates

So far, the masking of algorithms has received more attention than the masking of logic gates. For example, there are several publications on how to mask DES [1, 8] and AES [1, 2, 7, 24]. However, there also exist two publications [3, 9] that discuss masking in a more generic way. In particular, [9] discusses the theory of masked gates. In this article, masked circuits are referred to as private circuits. The goal of these circuits is to provide protection against an attacker that can probe a certain number of wires in a circuit. Power-analysis attacks are modelled as probing attacks because they allow the attacker to determine the value of a particular wire.

An important assumption that is implicitly made in [9] is that every wire changes its voltage level no more than once per clock cycle. A digital circuit is modelled as a graph, where the nodes correspond to gates and the connections correspond to wires. The propagation delay of the gates is not considered and therefore, no glitches occur in this model. However, glitches occur in digital circuits in practice and they have a significant impact on the power consumption of a circuit. Therefore, the model proposed in [9] needs to be updated in order to be applicable for circuits as they are used in practice.

The model used in [3] to analyze the security of masking does also not consider the effect of glitches. Hence, also this model needs to be extended accordingly.

2.2 Building Masked Gates Based on Multiplexors

One of the first patents on masking at the gate level has been issued to Messerges, Dabbish, and Puhl in 2001 [14]. This patent describes how an arbitrary logical function can be masked based on multiplexors and crossbar switches. All inputs of the logical function as well as the output are masked. Therefore, the interfaces of masked gates implemented according to [14] correspond to the one shown in Figure 2.

Implementations of masked gates using this approach are relatively big in practice. For example, a 2-input gate consists of 3 multiplexors, 3 crossbar-switches and 4 XOR gates. Nevertheless, in [6] it has been proposed to use this approach to secure a data scrambling technique against power-analysis attacks.

In the current article, we show that masked gates based on multiplexors do not prevent DPA attacks, if glitches occur in the masked circuit.

2.3 Building Masked Gates Based on Correction Terms

In [22] and [23], an alternative approach for the implementation of masked logic gates has been proposed. The basic idea of this approach is to build masked gates based on normal (unmasked) gates.

For example, the masked AND gate that is used in [22] and [23] to implement a masked AES co-processor consists of 4 AND gates and 4 XOR gates. The interface of this AND gate also corresponds to the one shown in Figure 2.

A similar approach as the one presented by Trichina and Korkishko has been patented by Klug, Kniffler, and Gammel in [10]. The main difference between these two approaches is that in the patent, the same mask is used for the inputs and the output, *i.e.* $m_a = m_b = m_q$. This leads to significantly smaller implementations of masked gates.

However, all these approaches are vulnerable to DPA attacks in theory and practice, if glitches occur in the masked circuit.

3 Theoretical Security Analysis of Masked Gates

In digital circuits, logical values are usually represented by voltage levels of wires. The power consumption of a digital circuit is data-dependent because keeping a wire at a certain voltage level requires almost no energy, while the switching of a voltage level requires a significant amount of energy.

We denote the energy that is needed to switch a wire from the voltage representing the value 0 to the voltage representing the value 1 as $E_{0\to1}$. Accordingly, we denote the energy that is needed to perform a $(1 \to 0)$ transition as $E_{1\to0}$. In practice, these energies are usually different, *i.e.* $E_{0\to1} \neq E_{1\to0}$. Although keeping a wire at a certain voltage level requires almost no energy, we also introduce a notation for these energies. We refer to these energies as $E_{0\to0}$ and $E_{1\to1}$, respectively.

Besides a notation for the energy consumption, also certain assumptions about the data inputs of masked gates are required in order to perform an analysis of the DPA-resistance. In this article, we use the common assumption that the inputs of a gate in a digital circuit are statistically independent and

Table 1. The transitions a normal AND gate can perform during one clock cycle.

a	b	q	Energy
$0 \to 0$	$0 \to 0$	$0 \to 0$	$E_{0\to 0}$
$0 \to 0$	$0 \to 1$	$0 \to 0$	$E_{0\to 0}$
$0 \to 0$	$1 \to 0$	$0 \to 0$	$E_{0\to 0}$
$0 \to 0$	$1 \to 1$	$0 \to 0$	$E_{0\to 0}$
$0 \to 1$	$0 \to 0$	$0 \to 0$	$E_{0\to 0}$
$0 \to 1$	$0 \to 1$	$0 \to 1$	$E_{0\to 1}$
$0 \to 1$	$1 \to 0$	$0 \to 0$	$E_{0\to 0}$
$0 \to 1$	$1 \to 1$	$0 \to 1$	$E_{0\to 1}$
$1 \to 0$	$0 \to 0$	$0 \to 0$	$E_{0\to 0}$
$1 \to 0$	$0 \to 1$	$0 \to 0$	$E_{0\to 0}$
$1 \to 0$	$1 \to 0$	$1 \to 0$	$E_{1\to 0}$
$1 \to 0$	$1 \to 1$	$1 \to 0$	$E_{1\to 0}$
$1 \to 1$	$0 \to 0$	$0 \to 0$	$E_{0\to 0}$
$1 \to 1$	$0 \to 1$	$0 \to 1$	$E_{0\to 1}$
$1 \to 1$	$1 \to 0$	$1 \to 0$	$E_{1\to 0}$
$1 \to 1$	$1 \to 1$	$1 \to 1$	$E_{1\to 1}$

uniformly distributed. Based on this assumption and the notation for the energy consumption, we analyze the DPA-resistance of different logic gates in the following subsections.

First, we analyze the DPA-resistance of normal (unmasked) gates in Subsection 3.1. This analysis is presented in order to provide a reference for the analysis of masked gates. Subsection 3.2 discusses why masked gates provide DPA-resistance, if no glitches occur in a digital circuit. This is essentially a short summary of the arguments that have been used so far to promote masked gates as a countermeasure against DPA attacks.

In Subsection 3.3, we argue why the assumption that there are no glitches in a digital circuit is typically wrong in practice. This subsection in particular also discusses the effort that is necessary to avoid glitches in digital circuits.

Finally, in Subsection 3.4 we show why masked CMOS gates do not prevent DPA attacks, if glitches occur in a digital circuit.

3.1 Analyzing the DPA-Resistance of Normal Gates

A 2-input AND gate takes the two values a and b as input to calculate $q = a \wedge b$. For our analysis, we assume that the inputs arrive at the same time and that they change their values no more than once per clock cycle. We do not need to consider glitches for our analysis of normal gates because these gates are susceptible to DPA-attacks even if no glitches occur.

Each input of the AND gate can perform one out of four transitions ($0 \to 0$, $0 \to 1$, $1 \to 0$, or $1 \to 1$) during a given clock cycle. Hence, in total there exist $4^2 = 16$ possible combinations of input transitions that can occur. These combinations of input transitions are listed in Table 1. In addition to the input transitions, Table 1 also shows the corresponding output transitions and the energy that is needed to perform these transitions. All 16 cases shown in this table have the same probability of occurrence because the inputs a and b are statistically independent and uniformly distributed.

In a DPA attack on an AND gate that is part of a digital circuit, the power consumption of the circuit is first recorded several times while the circuit performs a cryptographic operation with different inputs. Subsequently, the power measurements are split into two groups. The first group contains all measurements, where $q = 0$ at the end of the clock cycle and the second group contains all measurements, where $q = 1$.

Using the notation introduced in this section, this means that the first group contains the cases where the output performs a $(0 \rightarrow 0)$ or a $(1 \rightarrow 0)$ transition, while the second group contains the remaining cases. The attacker calculates the means of the energies of both groups and subtracts them from each other.

$$\frac{3E_{0\rightarrow 1} + E_{1\rightarrow 1}}{4} \neq \frac{3E_{1\rightarrow 0} + 9E_{0\rightarrow 0}}{12} \tag{1}$$

The expected values of these two means are in general not equal and hence, there is a leakage of side-channel information. The processing of $q = 0$ requires a different amount of energy than the processing of $q = 1$. In practice, the number of samples that is needed to exploit this energy difference essentially depends on the background noise, *e.g.* due to other circuit parts, and on the values $E_{0\rightarrow 0}$, $E_{0\rightarrow 1}$, $E_{1\rightarrow 1}$, and $E_{1\rightarrow 1}$.

The corresponding analysis can also easily be carried out for other logic gates, such as OR and XOR. All these gates are susceptible to DPA attacks.

Throughout this article we focus on the correlation between the power consumption of logic gates and the data that is processed by the gates. This correlation determines the number of samples that are needed in DPA attacks in practice (see [12] and [16]).

3.2 Analyzing the DPA-Resistance of Masked Gates in Circuits without Glitches

Assuming that no glitches occur in a digital circuit, it is relatively easy to proof that masked gates are resistant to DPA attacks. We present the basic idea of these proofs based on a masked 2-input AND gate.

A masked 2-input AND gate takes five signals as input $(a_m, m_a, b_m, m_b, m_q)$ and calculates the output $q_m = ((a_m \oplus m_a) \wedge (b_m \oplus m_b)) \oplus m_q$. The assumption that there are no glitches in a digital circuit means that every input and output signal switches only once per clock cycle. Every input can perform one out of four transitions during a given clock cycle. Hence, there are $4^5 = 1024$ possible combinations of input transitions that can occur.

Like in the previous subsection, we have created a table containing all possible input transitions, the corresponding output transitions and the energies consumed by these output transitions. Based on this table it is possible to determine whether the processing of $q = 0$ and the processing of $q = 1$ require different amounts of energy or not.

In fact, it turns out that the expected value of the energy that is needed to process $q = 0$ and the corresponding expected value for the processing of $q = 1$ are identical. Furthermore, the table can be used to show that also DPA attacks on the inputs a and b are not possible. Assuming that there are no glitches in a digital circuit, the energy dissipation of a masked AND gate is indeed independent of the unmasked inputs and the unmasked output. Accordingly, it can be shown that implementations of other masked gates (OR, XOR, ...), as described in [10], [14], [22], and [23] are also resistant against first-order DPA attacks.

This fact has been used in the past to promote masked gates. However, in the following subsection, we discuss why the assumption that there occur no glitches in digital circuits usually does not hold true in practice.

3.3 Timing and Switching Characteristics of Digital Circuits

In practice, digital circuits are usually implemented based on CMOS (see [17]). Logical functions are realized by connecting multiple CMOS gates to each other. An important property of these gates is that they have a certain propagation delay, *i.e.* it takes a certain amount of time until the output of a gate reacts to a change at an input of the gate.

This property has a significant impact on the switching activity of a digital circuit. In such a circuit, the input signals of a gate are the outputs of different combinational paths. These paths do not necessarily have the same length. For example, it can happen that the input a of a gate always arrives earlier than the input b. The consequence of such a delay between the input signals is that the gate switches its output more than once per clock cycle. The output switches when the input a performs a transition and it switches again when the input b performs a transition. It is important to note that in the time span between the arrival of the two input signals, the output of the gate is switched to a "wrong" value. This "wrong" value is potentially the input of another logic gate. Of course, such a gate reacts to this transition at its input and changes its output based on the "wrong" input value. In this way, "wrong" values propagate through the circuit.

The consequence of all this is that a lot of unintended switching activity takes place before every wire in a combinational circuit settles to the final value. In practice, glitches account for a significant amount of the power consumption of a circuit. Hence, glitches cannot be neglected in a thorough analysis of the DPA-resistance of masked gates. In the following subsection, we show that glitches make masked gates susceptible to DPA attacks.

Glitches occur in classical CMOS circuits and of course they also occur in masked circuits that are based on CMOS. However, besides CMOS there are many other logic styles that can be used to implement digital circuits. Among them, there are actually some that prevent glitches.

Glitches do not occur in so-called domino logic styles, such as for example pre-charged NMOS [17], DCVSL [17] or SABL [20]. However, pre-charged circuits are usually bigger than corresponding CMOS circuits. Another major disadvantage of these logic styles compared to CMOS is the lack of automated off-the-shelf circuit synthesis tools.

The papers and patents that have been published so far on masking at the gate level do not address the problem of glitches. Therefore, readers of these publications might implicitly assume that masking can be implemented based on CMOS. However, as we point out in the following subsection, this is not the case. In order to be sure that masked circuits are DPA-resistant, a logic style that prevents glitches needs to be used. This significantly increases the implementation costs of masked circuits.

3.4 Analyzing the Effect of Glitches on the DPA-Resistance of Masked Gates

In digital circuits that are based on CMOS, the input signals of logic gates can arrive at different moments of time. Furthermore, these signals switch potentially several times during one clock cycle. We now analyze the impact of these facts on an implementation of a masked 2-input AND gate.

In order to simplify the analysis, we make certain assumptions about how the delays between the input signals look like and about how often the input signals switch per clock cycle. However, these assumptions do not mean a loss of generality.

We assume that each input signal switches once per clock cycle and that at least one of the five input signals arrives at a different time than the other signals. Furthermore, if there is a difference between the arrival time of two signals, this difference is always assumed to be bigger than the propagation delay of the masked gate.

For the analysis of the susceptibility of the masked AND gate, we have used the same technique as in the previous subsections. We have created tables with the input transitions, the output transitions and the energy that is needed to perform the transitions.

First we have looked at the scenarios where only one of the five inputs arrives at a different moment of time than the remaining four inputs. There exist ten such scenarios. There are five input signals and each one of them can arrive either before or after the four other ones. One scenario is for example that m_q arrives first and that a_m, m_a, b_m and m_b arrive later.

Like in Subsection 3.2, in every scenario there exist $4^5 = 1024$ possible combinations of transitions that can occur at the inputs. However, in the ten scenarios where the inputs arrive at two different moments of time, the output of the masked AND gate performs two transitions instead of one. One transition is performed when the single input performs a transition and another one is performed when the other four input signals perform a transition.

We have analyzed whether the energy dissipation that is needed to perform these two transitions is correlated to $q = q_m \oplus m_q$ or not. This was done by calculating the expected value for the energy needed to process $q = 0$ and the corresponding expected value for $q = 1$ (see Subsection 3.1). The same has also been done for the unmasked inputs a and b. A masked gate is only resistant to DPA-attacks if the energy dissipation of the gate is uncorrelated to all unmasked inputs and outputs.

Unfortunately, it has turned out that in all ten scenarios the energy that is needed to perform the two transitions of the output is correlated to a, b or q. We have also investigated all remaining scenarios. These are for example the scenarios where two inputs arrive at separate moments of time either before or after the remaining three arrive. However, in the analysis of all scenarios, starting from the one where only one signal arrives at a different time to the scenario where all inputs arrive at separate moments of times, there has always been a correlation to a, b or q.

In practice, different arrival times are very common. In case of a masked gate, it is in particular very likely that the masks m_a, m_b and m_q arrive at different moments of time than the inputs a_m and b_m. The reason for this is that the masks are generated by a completely different part of the digital circuit.

Based on the scenarios we have analyzed in this section, we have to conclude that there exists no implementation of a masked AND gate based on CMOS that is resistant to DPA attacks. We have performed the same analysis as for the masked AND gate also for masked NAND, OR, NOR, XOR and XNOR gates.

It has turned out that masked nonlinear gates, such as AND, NAND, OR and NOR gates, are susceptible to DPA-attacks, while masked linear gates, such as XOR and XNOR gates, are resistant to DPA attacks. However, for implementations of operations like the AES S-Box, nonlinear operations are crucial (see [18] and [25]).

Therefore, the conclusion of our theoretical analysis is that all published gate-level masking schemes need to be implemented based on a logic style that prevents glitches.

4 The DPA-Resistance of Masked Gates in Practice

In order to empirically verify the results of the theoretical analysis presented in the previous section, we have performed simulated DPA attacks on implementations of masked 2-input AND gates. For this purpose, we have implemented the masked AND gate presented in [14] and a masked AND gate based on the approach described in [22, 23]. Both gates have been implemented using a CMOS standard cell library based on a 0.35 μm technology.

We have performed SPICE simulations of these gates for two scenarios. In the first scenario, all five inputs of the masked gates have arrived at the same time. In the second scenario, the output mask m_q has arrived first and the remaining inputs have arrived one nanosecond later. Like in the theoretical analysis, each input signal has only performed one transition per clock cycle.

We have simulated one power trace for each of the $4^5 = 1024$ combinations of input transitions that can occur. Subsequently, a DPA attack on q has been performed. The goal was to check whether the mean power consumption for $q = 0$ and the mean power consumption for $q = 1$ are indeed different or not. It is important to point out that the AND gates have been implemented exactly as described in [14] and [22, 23], respectively. Hence, there was no wire in the circuit that stored q directly. However, the glitches in the gates have lead to the fact that the power consumptions of the masked gates were correlated to q.

In order to provide a reference for the detected correlations, we have also performed a DPA attack based on simulated power traces on a normal AND

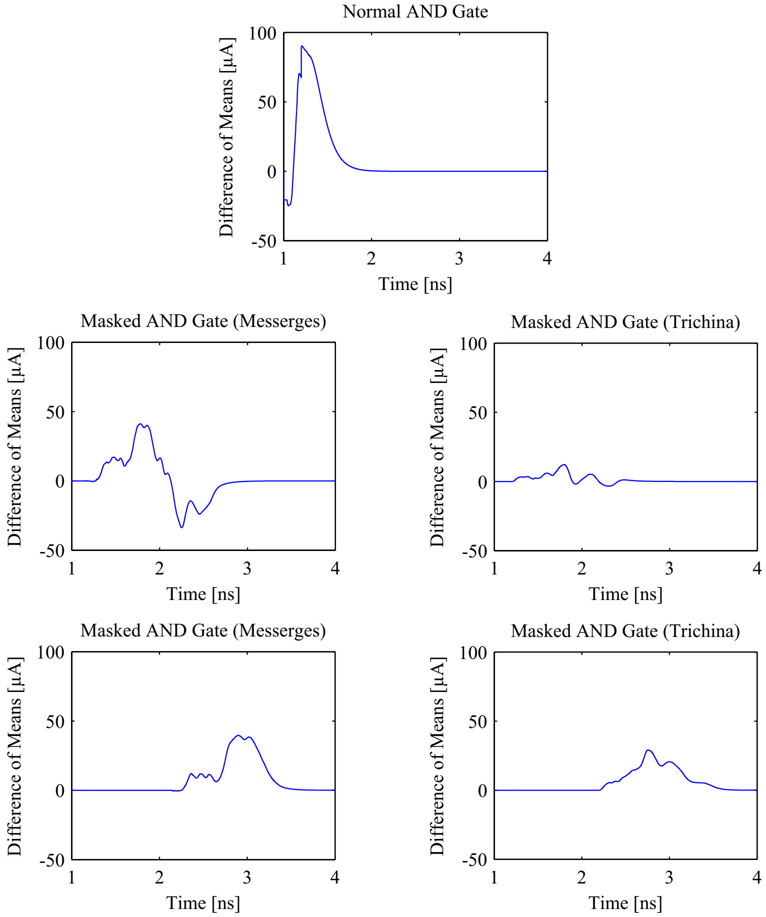

Fig. 3. The results of DPA attacks on a normal AND gate and of attacks on masked AND gates implemented according to [14] and [22, 23].

gate. The results of all attacks are shown in Figure 3. The first three plots are the result of attacks that are based on simulations where the inputs have arrived at the same time. Even in this scenario, the power consumptions of the masked gates are correlated to q. In fact, this is not surprising. The masked gates consist of unmasked CMOS gates. Consequently, even if the inputs arrive at the same time, glitches occur in the masked AND gates.

The last two plots show the results of attacks on implementations where m_q arrives one nanosecond before the other inputs. This time difference affects in particular the implementation according to [22, 23]. The time difference leads to a significant increase of the maximum of the DPA peak that occurs in the attack.

In the two scenarios we have analyzed for the masked gates, the DPA peaks that occur are obviously smaller than the peak that occurs in an attack on the normal AND gate. However, the two scenarios are just examples of attacks on the output q. We have also performed attacks on a and b and we have also looked at scenarios with other delays between the input signals. In fact, there are actually scenarios where peaks in the range of those of unmasked implementations occur.

In practice, it is extremely difficult to control the delay between the input signals of a gate. In the semi-custom design flows that are usually used to implement ICs, the designer has almost no control over these delays. Therefore, almost any delay scenario occurs in a big circuit in practice.

The goal of this article is to show that glitches are a problem for the DPA-resistance of masked CMOS circuits. We have not explicitly searched for the scenario that maximizes the DPA peak occurring in an attack on a particular implementation. Instead, we have presented two simple scenarios that should make our point clear. Already in these simple scenarios, the maxima of the DPA peaks are only a little bit more than halved by masking the gate. This is definitely less than one would expect from this countermeasure. A reduction of the DPA peak in this range can also be achieved by more inexpensive countermeasures such as the generation of noise [4, 12].

A last point that is important to mention is that the results of the simulated attacks presented in this section can not be compared directly with our theoretical analysis conducted before. The reason for this is the fact that the masked gates are built with unmasked CMOS gates. Hence, glitches occur not only outside the gates, but also inside the gates. The DPA peaks shown in Figure 3 are the result of the superposition of the effect of all kinds of glitches. However, as discussed in the theoretical analysis, masked gates are also susceptible to DPA attacks, if glitches occur only outside the masked gates.

5 Conclusions

There are several publications and patents on masking at the gate level. We have shown that all proposed implementations of masked gates based on CMOS are susceptible to DPA attacks because of glitches. Glitches have been completely ignored in previous analyses of masking at the gate level.

In this article, we have performed a theoretical analysis of the effect of glitches on masked gates. Furthermore, we have presented results of DPA attacks based on SPICE simulations of masked gates as they have been proposed in [14] and [22, 23]. Both approaches have turned out to be susceptible to DPA attacks.

Glitches in digital circuits can be prevented by using domino logic styles. However, implementations based on such logic styles are usually bigger than implementations based on CMOS. Also the design effort for circuits using domino logic styles is significantly higher than the one for corresponding CMOS circuits. This is a consequence of the fact that commercial synthesis tools for domino logic styles are currently not available. Hence, the protection of digital circuits against DPA attacks based on masked logic gates is very expensive in practice.

References

1. Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES, Secure against Some Attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *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.
2. Johannes Blömer, Jorge Guajardo Merchan, and Volker Krummel. Provably Secure Masking of AES. Cryptology ePrint Archive (`http://eprint.iacr.org/`), Report 2004/101, 2004.
3. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.
4. Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Çetin Kaya Koç and Christof Paar, editors, *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 252–263. Springer, 2000.
5. Jean-Sébastien Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.
6. Jovan D. Golić. DeKaRT: A New Paradigm for Key-Dependent Reversible Circuits. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 98–112. Springer, 2003.
7. Jovan D. Golić and Christophe Tymen. Multiplicative Masking and Power Analysis of AES. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *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.
8. Louis Goubin and Jacques Patarin. DES and Differential Power Analysis – The Duplication Method. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
9. Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
10. Franz Klug, Oliver Kniffler, and Berndt Gammel. Rechenwerk, Verfahren zum Ausführen einer Operation mit einem verschlüsselten Operanden, Carry-Select-Addierer und Kryptographieprozessor. German Patent DE 10201449 C1, January 2002.

11. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In Michael Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

12. Stefan Mangard. Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness. In Tatsuaki Okamoto, editor, *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

13. Renato Menicocci and Johan Pascal. Elaborazione Crittografica di Dati Digitali Mascherati. Italian Patent IT MI0020031375A, July 2003.

14. Thomas S. Messerges, Ezzy A. Dabbish, and Larry Puhl. Method and Apparatus for Preventing Information Leakage Attacks on a Microelectronic Assembly. US Patent 6,295,606, September 2001. Available online at http://www.uspto.gov/.

15. Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Power Analysis Attacks of Modular Exponentiation in Smartcards. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 1999.

16. Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, 51(5):541–552, January 2002.

17. Jan M. Rabaey. *Digital Integrated Circuits*. Prentice Hall, 1996. ISBN 0-13-178609-1.

18. Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A Compact Rijndael Hardware Architecture with S-Box Optimization. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 239–254. Springer, 2001.

19. Adi Shamir. Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies. In Çetin Kaya Koç and Christof Paar, editors, *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 71–77. Springer, 2000.

20. Kris Tiri and Ingrid Verbauwhede. Securing Encryption Algorithms against DPA at the Logic Level: Next Generation Smart Card Technology. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 137–151. Springer, 2003.

21. Kris Tiri and Ingrid Verbauwhede. A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In *2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004), 16-20 February 2004, Paris, France*, pages 246–251. IEEE Computer Society, 2004.

22. Elena Trichina. Combinational Logic Design for AES SubByte Transformation on Masked Data. Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2003/236, 2003.

23. 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.
24. Elena Trichina, Domenico De Seta, and Lucia Germani. Simplified Adaptive Multiplicative Masking for AES. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *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.
25. Johannes Wolkerstorfer, Elisabeth Oswald, and Mario Lamberger. An ASIC implementation of the AES SBoxes. In Bart Preneel, editor, *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.