IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2020, No. 4, pp. 416–442. DOI:10.13154/tches.v2020.i4.416-442

Unrolled Cryptography on Silicon

A Physical Security Analysis

Thorben Moos

Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
thorben.moos@rub.de

Abstract. Cryptographic primitives with low-latency performance have gained momentum lately due to an increased demand for real-time applications. Block ciphers such as PRINCE enable data encryption (resp. decryption) within a single clock cycle at a moderately high operating frequency when implemented in a fully-unrolled fashion. Unsurprisingly, many typical environments for unrolled ciphers require protection against physical adversaries as well. Yet, recent works suggest that most common SCA countermeasures are hard to apply to low-latency circuits. Hardware masking, for example, requires register stages to offer resistance, thus adding delay and defeating the purpose of unrolling. On another note, it has been indicated that unrolled primitives *without* any additional means of protection offer an intrinsic resistance to SCA attacks due to their parallelism, asynchronicity and speed of execution. In this work, we take a closer look at the physical security properties provided by unrolled cryptographic IC implementations. We are able to confirm that the nature of unrolling indeed bears the potential to decrease the susceptibility of cipher implementations significantly when reset methods are applied. With respect to certain adversarial models, e.g., ciphertext-only access, an amazingly high level of protection can be achieved. While this seems to be a great result for cryptographic hardware engineers, there is an attack vector hidden in plain sight which still threatens the security of unrolled implementations remarkably – namely the static power consumption of CMOS-based circuits. We point out that essentially all reasons which make it hard to extract meaningful information from the dynamic behavior of unrolled primitives are not an issue when exploiting the static currents for key recovery. Our evaluation is based on real-silicon measurements of an unrolled PRINCE core in a custom 40 nm ASIC. The presented results serve as a neat educational case study to demonstrate the broad differences between dynamic and static power information leakage in the light of technological advancement.

Keywords: Unrolled Cryptography · Low-Latency Cryptography · PRINCE · Side-Channel Analysis · Static Power SCA · SPSCA

1 Introduction

Physical security becomes a concern whenever cryptography is deployed in a field that puts the hardware responsible for executing cryptographic primitives in a potentially hostile environment. Years of academic and industrial research have revealed the unpleasant truth that no universal solution exists to protect cryptographic devices from key recovery attacks when they are forced to operate under permanent physical exposure to untrusted parties. Although significant advances have been made in developing dedicated protection mechanisms against this threat, there is still neither one guaranteeing full resistance, nor any that is universally applicable to all hardware and software implementations alike (without significant adjustments).

Received: 2020-04-15 Accepted: 2020-06-15 Published: 2020-08-26

Glitch-Resistant Masking.

With respect to the protection of hardware implementations against passive and non-invasive physical attacks, glitch-resistant masking (a.k.a. hardware-based masking or hardware masking) has become one of the most promising research directions. This particular field was sparked by the introduction of threshold implementations (TIs) in 2006 [NRR06] and has been complemented by a number of further schemes (e.g., [RBN+15, CRB+16, GMK16, GMK17, GM17, GM18, GIB18, FGP+18]) that are summarized and analyzed in [MMSS19]. The common denominator between all of them is the one vital ingredient strictly required to achieve provable security in the presence of glitches, namely the correct instantiation of register stages (see [MMSS19]). Registers are a fundamental part of the approach as they prevent the propagation of glitches between combinatorial (sub-)circuits. Naturally, such a concept is not applicable to implementations where the inclusion of clocked memory elements contradicts a certain design goal, as it is the case for fully-unrolled low-latency circuits. In contrast to other common block cipher implementations strategies (e.g., round-based or serialized), an unrolled circuit is a fully combinatorial representation of the whole encryption (resp. decryption) function without any memory elements incorporated[1]. Clearly, such a design strategy has significantly higher demands in terms of area usage, as no part of the cipher, e.g., a substitution box (Sbox) or a round function, may be reused during the cryptographic operation. Yet, the unrolled implementation style enables the fastest possible execution as it avoids the additional delay to store or synchronize intermediate results. In summary, there is an inherent conflict between state-of-the-art hardware-based masking and the desired high-speed single-cycle execution property of low-latency ciphers. The difficulty of combining low-latency performance with glitch-resistant masking has been extensively discussed at Asiacrypt 2016 for FPGA platforms [MS16a]. Schneider *et al.* attempt to balance the trade-off between physical security and speed of execution in multiple different case studies. One of the considered variants is to mask only the outer rounds of a block cipher and leave the rounds in the middle unrolled and unprotected. Another is the realization of implementations equipped with hardware masking (e.g., TI) as asynchronous circuits, i.e., regular register stages are included but controlled in a clock-less and self-timed fashion. Yet, none of these options fully preserves the desired low-latency characteristic. The only scheme that enables secure masking in the presence of glitches without strictly requiring synchronization stages (and the ensuing latency penalty) has been proposed by Gross *et al.* at CHES 2018 [GIB18]. The general concept is based on the observation that no register stages are needed between consecutive masked non-linear operations, when skipping the share compression that is usually performed to reduce the number of output shares to its minimum. Yet, by applying this technique to mask a certain function, the number of shares required per intermediate result (and therefore the size of the circuit) grows exponentially in the number of subsequent non-linear operations [GIB18]. Naturally, a full cipher instantiation contains a large number of non-linear operations, making the technique rather impractical for a fully-unrolled block cipher. Without integrating at least a couple of resharing and compression stages, containing registers and demanding the addition of fresh randomness, the circuit size and the number of output shares would simply explode.

Unrolling as an Implicit Countermeasure.

We summarize that applying hardware masking to unrolled ciphers is neither trivial nor cheap. None of the available options actually preserves the low-latency property at a reasonable price. However, it has been argued before that such concepts may not even be required in order to achieve a proper level of resistance against side-channel attacks. In

[1] The concept of unrolled hardware implementations should not be confused with the common optimization strategy of (loop-)unrolling in software implementations.

fact, even before dedicated low-latency primitives became prominent in cryptography, it was pointed out that the nature of unrolling itself may serve as a decent countermeasure against classical passive attacks, especially when certain usage and design principles are respected. The underlying observation is simply that the fast, asynchronous and highly-parallel execution actively prevents an adversary from capturing the dissipation of the target part of the circuit in sufficiently high quality. Bhasin *et al.* proposed at CT-RSA 2010 to unroll hardware implementations of cryptographic primitives in order to prevent adversaries from learning sensitive data through physical measurables [BGSD10]. The crucial prerequisite is hereby that the data path is cleared between each two consecutive encryptions, which obviously halves the available throughput. In such a scenario the adversary can not predict the Hamming distance (HD) between consecutively processed values in the first round(s) and learns less information. While first experiments focused on the DES block cipher [BGSD10], similar results showing the effectiveness of unrolling against SCA attacks have been demonstrated on the AES as well [MMP11].

In 2012 the first dedicated low-latency block cipher was introduced by the name of PRINCE [BCG+12]. This primitive has been specifically developed to be implemented in a fully-unrolled fashion in order to encrypt and decrypt data efficiently in a single clock cycle. Such a lightweight and high-speed encryption engine is a crucial component for the secure communication in pervasive computing environments with real-time security needs. The intrinsic SCA resistance attributed to the unrolled implementation style may not have been the primary objective during the design process, but it certainly is a welcome side effect as pervasive computing solutions are often threatened by physical adversaries as well. In order to assess the physical security properties of unrolled primitives, several articles have analyzed PRINCE implementations regarding their susceptibility to side-channel attacks, e.g., [YHA15], [MS16a], [YHA17a], [YHA17b] and [CSR+19]. The main focus of these works is finding efficient ways to exploit the observable leakage despite the challenges that the single-cycle execution presents in that regard. The applied techniques range from frequency analysis [CSR+19] to the smart selection of Points-of-Interest (PoI) [YHA15]. Others have pointed out that unrolled ciphers are particularly susceptible in chosen-input scenarios [YHA17a]. These attempts emphasize the additional exploitation effort that has to be invested in order to analyze the security of unrolled implementations, but also the importance of considering different adversary models.

Static Power Side-Channel Analysis (SPSCA).

The landscape of power analysis attacks has changed significantly over the years. The continuous down-scaling of circuit technology has led to a decline of the dynamic power consumption per individual logic unit due to smaller capacitances and supply voltages involved. As a result, it becomes increasingly difficult to target the dissipation of small parts of a circuit in divide and conquer based power analysis attacks. The progressive decrease of propagation delays only benefits this development. At the same time, the static power consumption intensifies in newer technology generations and reaches a significant magnitude in sub-100 nm complementary metal-oxide-semiconductor (CMOS) technology [Moo19]. Hence, it is fair to wonder whether such contrary trends may lead to a shift of the primarily targeted side channel when considering implementations in advanced technology nodes. Numerous advances have been made in recent literature towards a better understanding of the static power consumption of CMOS-based circuits as a source of information leakage. At CHES 2019 it was demonstrated that the difference in susceptibility between two successive CMOS technology nodes can be as large as a 10-fold increase [Moo19]. It was also shown that exponential dependencies of this side channel on the temperature and the supply voltage can be exploited by adversaries to escalate the leakage of information [Moo19, MMR20]. Finally, it was discovered that SPSCA attacks can be performed without obtaining control over the clock signal of the device under test (DUT) when sensitive intermediates remain in the circuit after cryptographic operations and are not subject to an immediate modification [Moo19]. This is particularly relevant for our work, as unrolled circuits, due to the nature of their usual applications, are commonly deployed without any reset signal or key-removal mechanisms, as high performance is often the main criterion. Unrolled circuits which are instantiated without any considerations of this issue, are a prime example of implementations where the full state, containing all sensitive intermediates, remains in the circuit between any two consecutive encryptions. Yet, this side channel has never been considered as a complementary attack vector when evaluating the SCA security of unrolled circuits.

FPGA vs. ASIC.

Latency-optimized ciphers are primarily attractive for ASIC platforms. A cryptographic primitive, like any kind of computation, can unfold its full potential with respect to execution speed when realized in an advanced IC technology node as a semi-custom (standard-cell-based) or full-custom design. Hence, a striking issue with all the previously listed works, analyzing the physical security of unrolled PRINCE implementations, is simply that all of them are based on FPGA case studies. At first glance, this may not appear to be an overly limiting factor for the general validity of the reported results. FPGA case studies are frequently used to make generalized statements about hardware implementations. However, specifically for exceptional implementation styles such as unrolling, it is not always possible to transfer conclusions from FPGA to ASIC platforms in a meaningful way. To illustrate the discrepancy between the two hardware platforms in more detail, we refer to the so-called *cost of programmability* [KR07]. According to the seminal work by Kuon *et al.* [KR07], a fully combinatorial representation of a function (such as unrolled PRINCE) requires about **35** times as much area on an FPGA as on a standard-cell-based ASIC, due to the structure of the programmable fabric. Clearly, such a significant increase in the number of gates involved in the computation leads to a much higher power consumption and delay as well. In particular, the authors observed that regular logic designs are more than **4** times slower on an FPGA, while consuming **14** times as much dynamic power as an equivalent ASIC design in the considered 90 nm reference technology [KR07]. Obviously, a 300% faster circuit which consumes 93% less dynamic power is significantly harder to exploit via side-channel analysis. In summary, without an ASIC-based case study, an important benchmark is missing in order to understand how susceptible low-latency cryptography is towards attacks when implemented in its predestined environment.

1.1 Our contribution

We present an extensive analysis of the physical security level that an unrolled, latency-optimized, cryptographic primitive can provide when implemented in state-of-the-art ASIC technology. Surprisingly, no similar case studies seem to exist in public literature, despite their importance for the cryptographic community as well as the industry sector, showcased by the deployment of such primitives in real-world security produts[2]. By performing our analysis we contribute and discover a variety of novelties. In summary, we find that a few comparably inexpensive usage principles can greatly reduce the information leakage of unrolled primitives through dynamic circuit emanations. The static leakage on the other hand, due its different nature, remains informative and requires special care. Our observations can be used to guide the secure (and low-cost) implementation of unrolled cryptography on silicon and even the protocol design surrounding it. Furthermore, our case study is of educational value as it highlights the broad conceptual differences between

[2] The LPC55S microcontroller series by NXP semiconductors for example deploys PRINCE for memory encryption.

static and dynamic power information leakage in a simple and vivid manner. We express all contributions of this work in more detail in the following sub-categories:

Effectiveness of Unrolling as a Countermeasure.

For the first time in public literature, we perform a physical security analysis of an unrolled ASIC implementation of the low-latency cipher PRINCE. Our experiments on the custom 40 nm ASIC confirm that it is straightforward to extract parts of the secret key through side-channel attacks when the adversary obtains knowledge of consecutively encrypted plaintexts under a fixed key. In such a case, the Hamming distance (HD) between consecutively processed first-round Sbox outputs serves as an efficient distinguisher. However, it is claimed in [BGSD10] that clearing the data path between each two encryptions is an effective protection against this threat. We evaluate whether this claim holds for our target. Since *clearing* the data path is a rather vague description of the action to be performed, we test 4 different reset (i.e., clearing) methods and evaluate their worth against their cost. Our analysis shows that setting the plaintext input of the unrolled circuit to a random state (unknown to the adversary) between each two encryptions, while leaving the key constant, is most cost-effective and delivers a high level of resistance.
We also argue that unrolled cryptography, by nature, is extremely resistant to attacks when considering an adversary model with ciphertext-only access. In particular, it is very hard to exploit an unrolled implementation from the ciphertext side (i.e., targeting the last round(s)) when using the dynamic power consumption as a source of information leakage. The signal-to-noise ratio (SNR) degrades very quickly after the first round(s) of the implementation and the asynchronicity of signals in later rounds grows significantly. This resistance may be exploited by designers in such a way that unrolled primitives are used in protocols or modes of operation where an adversary may obtain the ciphertexts but not the plaintexts. Intuitively, such a scenario appears reasonably often in real-world applications, since the ciphertext is commonly transmitted over an insecure channel while the plaintext is often kept secret.

Impact of Static Power Leakage on Unrolled Crypto.

Both observations, the effectiveness of resetting the circuit to a random state between encryptions and the difficulty to exploit the later rounds, give reason for optimism regarding the SCA security of unrolled cryptography. It appears that unrolled circuits, if carefully used, can provide a high level of protection against common dynamic power SCA attacks at a comparably low cost. However, we demonstrate that this is not the case when analyzing the static power for key recovery. Obviously, the static power consumption does not leak information about a transition between consecutive states, but only about the current state of the circuit. Hence, any reset method is conceptually ineffective, as the previous state of the circuit is no part of the leakage function and does not have any impact on the static power consumption[3]. Additionally, a static power adversary can target each round of the unrolled block cipher with approximately the same effort due to the value-based information leakage. All logic gates leak at the same time about their inputs and the intensity of their leakage does not depend on their position in the circuit (e.g., how close to the input or output they are located). Accordingly, attacks with ciphertext access on the last round are not expected to be any more complex than attacks with plaintext access on the first round. This is a valuable asset for an adversary, as a common first-round attack on PRINCE can recover at most 64 bits of information about the 128-bit key. To retrieve more information, either a deeper hypothesis into the second round needs to be

[3] Reset methods can be effective it the adversary can *not* control the clock. This is discussed later in more detail.

made, or the last round has to be targeted. While this is not problematic for a static power adversary, an attacker exploiting the dynamic currents might struggle significantly.

Dynamic vs. Static Comparison.

We provide a detailed comparison between the two essential power consumption side channels, dynamic and static, with respect to a cryptographic primitive realized in 40 nm ASIC technology. Earlier comparisons between both side channels exist in the literature [PSKM15, MMR17]. However, our results are based on a more recent semiconductor technology node and exploit thermal as well as voltage dependencies for both, dynamic and static power attacks, to maximize the signal-to-noise ratio (SNR). Our observations clearly help to understand the current state of the technology-scaling-induced race between the two side channels.

Static Power Novelties.

This is the first work that presents static power side-channel attacks on an ASIC implementation of a cryptographic primitive in such an advanced technology node (40 nm). Previous results, which in part also exploit thermal and voltage dependencies, have been reported for ASICs manufactured in 65 nm [PSKM15, KMM19, Moo19], 90 nm [Moo19] and 150 nm [MMR20] technology. This is also the first work that reports SPSCA attacks on an unrolled cryptographic primitive on any platform.

2 Preliminaries

Before discussing the target and the results of our simulations and practical experiments, we shortly revisit a few concepts that are crucial for the understanding of this work. We conclude this section with a toy example for illustration purposes.

2.1 Useful and Useless Transitions in Logic Circuits

The main contributors to the dynamic power consumption of today's physical logic circuits are the charging, discharging and short-circuit currents which are consumed during the transition of a CMOS gate's output from low to high or vice versa [MOP07]. Such a transition of a logic gate's output, a.k.a. toggle, can either be of useful or useless nature. Useful transitions are required to ensure correct functionality, while useless ones are not. A sequence of two useless transitions, e.g., $0 \rightarrow 1 \rightarrow 0$ or $1 \rightarrow 0 \rightarrow 1$, is called a glitch. It has been known for a long time that such glitches are responsible for unnecessary energy loss in combinatorial logic circuits [LvMJ95]. The concrete number of glitches occurring in the evaluation of a combinatorial circuit mainly depends on the logic depth of the circuit, the fanout of each gate and how balanced the propagation delays are. In logic circuits with a large logic depth and a significant fanout per gate the power consumption caused by glitches can be immense. In 1995 already, the authors of [LvMJ95] have shown an example where 60% of the switching activity in an 8x8 array multiplier is caused by glitches and therefore unnecessary. In a 16x16 array multiplier even 77% of switching activity corresponds to glitches [LvMJ95]. In our unrolled PRINCE circuit, glitches account for 96% of all gate toggles on average when both inputs, key and plaintext, make a random transition. They still account for almost 92% on average when the key remains fixed and only the plaintext is changed from one random value to another.

Table 1: Input-dependent leakage current of a 2-input NAND gate in 45 nm technology taken from [AO14].

Input	Leakage Current [nA]
0,0	57.63
0,1	38.55
1,0	72.27
1,1	107.07

2.2 Data-Dependent Static Power Consumption

The static power consumption of CMOS-based circuits has become a relevant source of energy dissipation in more recent years due to the semiconductor technology moving towards nanometer dimensions [AO14]. It is well-known that this share of a circuit's power consumption depends on the logical values that are applied to the inputs of logic gates. In [KMM19] it is described how the structure of CMOS gates contributes to the severe input dependency of their leakage currents. To gain an impression of the magnitude and data dependency of the leakage currents conducted by individual gates we refer the reader to [AO14], where typical values for common gates in different nanometer-scaled technologies are presented. As an example, we provide the input-dependent leakage currents for a 2-input NAND gate in 45 nm technology in Table 1. It can be observed that a NAND gate conducts an almost three times larger current in a stable state for the most leaking input combination (1,1) than for the least leaky one (0,1). Such a difference can indeed be significant enough, especially when accumulating over multiple gates, to be exploited by adversaries to break cryptographic implementations. The practicality of such attacks has been demonstrated multiple times in literature [Mor14, PSKM15, MMR17, BCS$^+$17, Moo19, MMR20].

2.3 A Toy Example

In order to emphasize the broad differences between the nature of static and dynamic power information leakage we have constructed a toy encryption circuit and evaluate its behavior for two exemplary input transitions. The circuit is depicted in Figure 1. It receives two plaintext inputs p_0, p_1, two key inputs k_0, k_1 and calculates two ciphertext outputs c_0, c_1. The numbers denoted inside each logic gate correspond to their propagation delays in time delay units (tdu). Table 2 presents the input-dependent leakage currents for all types of logic gates in the circuit.

For the first exemplary input transition, we assume that input vector $(p_0, k_0, p_1, k_1) = (0, 1, 0, 0)$ has fully propagated through the circuit and resulted in output $(c_0, c_1) = (0, 1)$. In that case, the circuit idles in the state that is illustrated at the top of Figure 2, where the green color corresponds to a signal value of '1' and the red color corresponds to '0'. As a

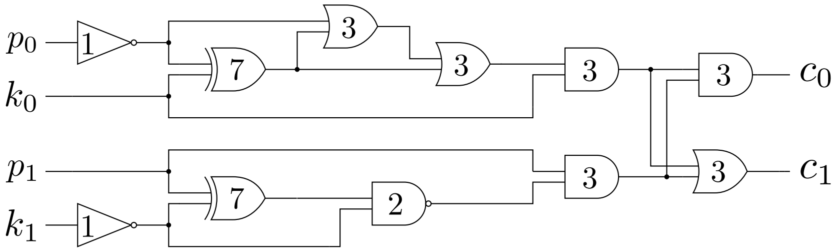


Figure 1: Toy encryption circuit with two plaintext inputs p_0, p_1, two key inputs k_0, k_1 and two ciphertext outputs c_0, c_1. The numbers within the logic gates denote their propagation delay in time delay units (tdu).

Table 2: Fictional input-dependent leakage currents of different logic gates.

Input	I_{INV} [nA]	I_{XOR} [nA]	I_{OR} [nA]	I_{NAND} [nA]	I_{AND} [nA]
0,0	33.5	278.1	156.2	54.6	98.5
0,1	39.8	239.2	155.8	41.5	81.1
1,0	-	239.5	87.9	75.3	112.6
1,1	-	287.8	68.3	112.1	143.4

next step, we assume that another input vector, namely $(p_0, k_0, p_1, k_1) = (1, 1, 0, 0)$, arrives at the input and propagates through the circuit which still holds the state corresponding to the previous input. Only one bit, namely p_0, makes a transition, while the remaining values are identical in both of the consecutive input vectors. This single-bit transition causes the sequence of gate toggles that is depicted by the timeline at the bottom of Figure 2. The timeline can be interpreted as a depiction of the dynamic power consumption caused by gate toggles. For simplicity it does not distinguish between $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions. As listed in Table 3 the input transition causes a total of 10 gate toggles, whereby 8 of them are useless, i.e., 4 glitches occur. The time to propagate the correct values to the output and keep them stable is 17 time delay units.

Unlike its dynamic counterpart, the static power consumption is not caused or affected by a transition between states. In fact, static power is consumed whenever the circuit's gates are connected to a power supply. Its magnitude, however, depends on the logic values applied to the inputs of such powered logic gates. Hence, two different stable leakage values can be observed, one before and one after the transition. These leakage values are calculated in Table 4.

While the dynamic power consumption is significant for this first input transition, due to the amount of gate toggles caused, the static power consumption shows only a small difference between both states as most of the transitions were caused by glitches and reverted back to their old state.

For the second exemplary input transition, we now assume that input vector $(p_0, k_0, p_1, k_1) = (1, 1, 1, 0)$ has fully propagated through the circuit and resulted in output $(c_0, c_1) = (1, 1)$. This state is depicted at the top of Figure 3.

Similar to the previous example, we now propagate a second input vector $(p_0, k_0, p_1, k_1) = (1, 1, 0, 0)$, which in fact is the same as before, through the circuit which still holds the previous state. As before, only one input bit makes a transition, but this time it is p_1. This

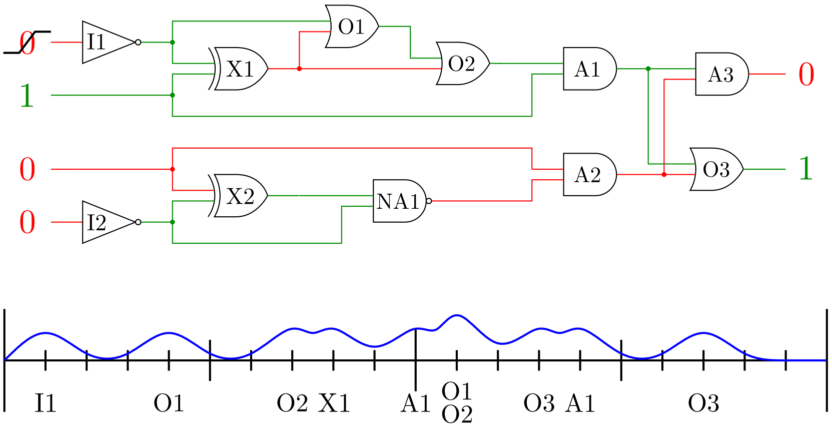


Figure 2: Circuit behavior for exemplary input transition $(p_0, k_0, p_1, k_1) = (0, 1, 0, 0) \rightarrow (1, 1, 0, 0)$. The timeline shows the occurrence of gate toggles over time.

Table 3: Number of gate toggles and glitches caused by the exemplary input transition and the total propagation time.

Input Trans. (p_0, k_0, p_1, k_1)	**Toggles**	**Glitches**	**Prop. time [tdu]**
$(0 \rightarrow 1, 1, 0, 0)$	10	4 (8 Trans.)	17

Table 4: Leakage currents exhibited by all circuit gates for the stable inputs before and after the input transition.

Input	$\mathbf{I_{I1}}$ **[nA]**	$\mathbf{I_{I2}}$ **[nA]**	$\mathbf{I_{X1}}$ **[nA]**	$\mathbf{I_{X2}}$ **[nA]**	$\mathbf{I_{O1}}$ **[nA]**	$\mathbf{I_{O2}}$ **[nA]**	$\mathbf{I_{O3}}$ **[nA]**	$\mathbf{I_{A1}}$ **[nA]**	$\mathbf{I_{A2}}$ **[nA]**	$\mathbf{I_{A3}}$ **[nA]**	$\mathbf{I_{NA1}}$ **[nA]**	$\mathbf{I_{SUM}}$ **[nA]**
(0, 1, 0, 0)	33.5	33.5	287.8	239.2	87.9	87.9	87.9	143.4	98.5	112.6	112.1	1324.3
(1, 1, 0, 0)	39.8	33.5	239.2	239.2	155.8	68.3	87.9	143.4	98.5	112.6	112.1	1330.3

toggle causes the sequence of gate transitions that is depicted at the bottom of Figure 3. The interesting observation is here that, although the same input is propagated through the same circuit, the behavior and dissipation of the circuit is vastly different in this second example. In particular, as listed in Table 5, only 4 gate toggles and 0 glitches occur. Furthermore, this second input transition leads to an execution time of only 6 *tdu* until the correct result is stable at the output (although further toggles occur in intermediate gates until 9 *tdu*). For the previous exemplary input transition, the output gates' last toggle occurred after 17 *tdu*.

This significant difference in the number of toggles and the execution time clearly highlights that the currently encrypted plaintext is only one part of the leakage function and can barely be correlated to the dynamic power dissipation of the circuit without knowledge of the initial state. Also, it underlines the fact that gates at similar stages of the circuit (e.g. directly at the output) are commonly evaluated at completely different moments in time depending on the transition at the input.

Finally, the static power consumption of the circuit before and after the transition is calculated in Table 6. In this scenario, the difference in the static power consumption is much larger than before, since more useful transitions occurred and affected the final state of the circuit. However, due to the lack of glitches the dynamic power consumption is much lower than for the first exemplary transition. In summary, the dynamic power

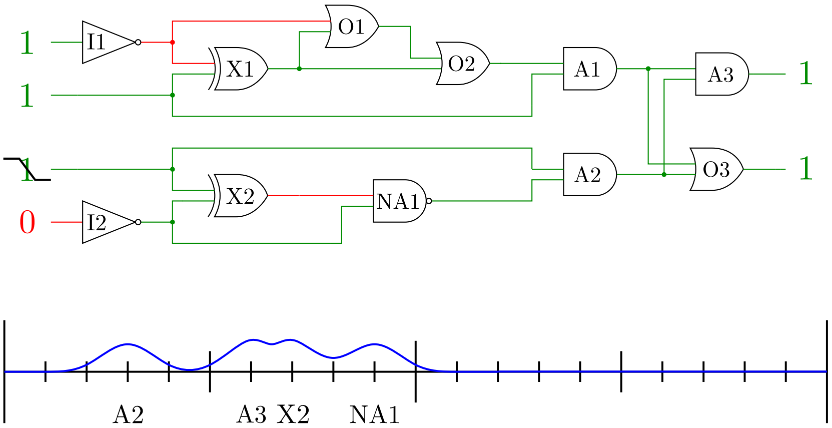


Figure 3: Circuit behavior for exemplary input transition $(p_0, k_0, p_1, k_1) = (1, 1, 1, 0) \rightarrow (1, 1, 0, 0)$. The timeline shows the occurrence of gate toggles over time.

Table 5: Number of gate toggles and glitches caused by the exemplary input transition and the total propagation time.

Input Trans. (p_0, k_0, p_1, k_1)	**Toggles**	**Glitches**	**Prop. time [tdu]**
$(1, 1, 1 \rightarrow 0, 0)$	4	0 (0 Trans.)	6

Table 6: Leakage currents exhibited by all circuit gates for the stable inputs before and after the input transition.

Input	$\mathbf{I_{I1}}$ **[nA]**	$\mathbf{I_{I2}}$ **[nA]**	$\mathbf{I_{X1}}$ **[nA]**	$\mathbf{I_{X2}}$ **[nA]**	$\mathbf{I_{O1}}$ **[nA]**	$\mathbf{I_{O2}}$ **[nA]**	$\mathbf{I_{O3}}$ **[nA]**	$\mathbf{I_{A1}}$ **[nA]**	$\mathbf{I_{A2}}$ **[nA]**	$\mathbf{I_{A3}}$ **[nA]**	$\mathbf{I_{NA1}}$ **[nA]**	$\mathbf{I_{SUM}}$ **[nA]**
(1, 1, 1, 0)	39.8	33.5	239.2	287.8	155.8	68.3	68.3	143.4	143.4	143.4	41.5	1364.4
(1, 1, 0, 0)	39.8	33.5	239.2	239.2	155.8	68.3	87.9	143.4	98.5	112.6	112.1	1330.3

consumption of an unrolled cryptographic primitive, like our toy cipher, is to a large part determined by the state of the combinatorial circuit before the actual input arrives. The following sections highlight how this situation can be exploited by designers to increase the side-channel security of unrolled circuits. The static power consumption, however, is not affected by any previous state and therefore inherently immune against countermeasures based on this fact.

3 Target

The target for our practical evaluations is a 40 nm CMOS ASIC prototype which is depicted in Figure 4. The chip is fully digital and 1.92 mm × 1.92 mm in size. It contains 8 metal layers for routing and has a nominal core voltage of 1.1V, as well as a nominal IO voltage of 2.5V. The ASIC has been developed for physical security evaluations and contains a number of different cipher cores which are all clock gated in order to make sure that they do not influence each other during evaluations. The position of the unrolled PRINCE implementation is indicated by the red circle on the left side of Figure 4. It occupies an area of 10 036 gate equivalents (GE), which corresponds to an on-chip-size of about 97.2 µm × 97.2 µm considering the utilization of about 75%. We have performed fully SDF annotated post-layout gate level simulations of the PRINCE core with picosecond accuracy at typical conditions. As stated in Section 2 already, the vast majority of gate toggles during the execution of the PRINCE core can be attributed to glitches. When both, the plaintext and the key are changed from one random value to another, the 9 169 logic gates in the circuit perform on average 114803 output transitions, while 96% of that are glitches. When only the plaintext is exchanged, 56920 gate transitions are caused, 92% of which correspond to glitches.

Our timing simulations further allow us to visualize the differences in execution time, defined as the time until the last output gate toggle occurs, for different input transitions and scenarios. Obviously such differences can not easily be observed in the power measurements presented in Section 4. Figure 5 shows the results of our simulations. We consider 2 different cases here and provide distributions for the occurrence of the first and the last output toggle respectively. We focus on the red distributions first. Here, we have simulated the unrolled PRINCE core for random plaintext transitions under a fixed key. In this scenario, the first output gates toggle after 2.1 to 2.4 nanoseconds. The final output toggle, before the output is stable and can be read from the crypto core, occurs after 4.2 to 4.5 nanoseconds. This already highlights the fact that gates corresponding to the last round of the cipher are evaluated at vastly different points in time, depending on the transition at

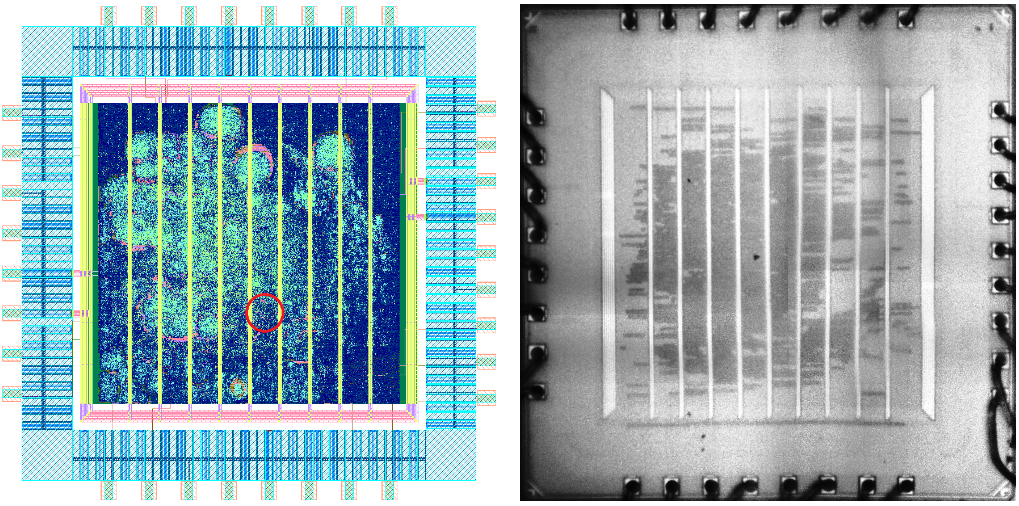


Figure 4: Layout schematic of the 40 nm ASIC on the left and microscope photography of the fabricated chip on the right.

the input. In that regard, it is important to note that differences of up to 2 nanoseconds can also be observed when monitoring the last toggle of a single output gate for different input transitions. Hence, dynamic power traces of unrolled primitives are inherently misaligned (asynchronous) at later stages of the circuit. It is also noteworthy that the quickest paths to the output require only half the amount of time that the longest ones require. The blue distributions in Figure 5 have been acquired when both, the plaintext *and* the key, are subject to random transitions. In that case the first toggles occur as early as 0.6 to 1.0 nanoseconds after the start. This is caused by the fact that the round key is propagated to all rounds at the same time and the path from the last round to the output is obviously shorter than the path from plaintext to output. The last toggle occurs later than for a fixed key due to the larger number of glitches that are caused when changing the key as well.

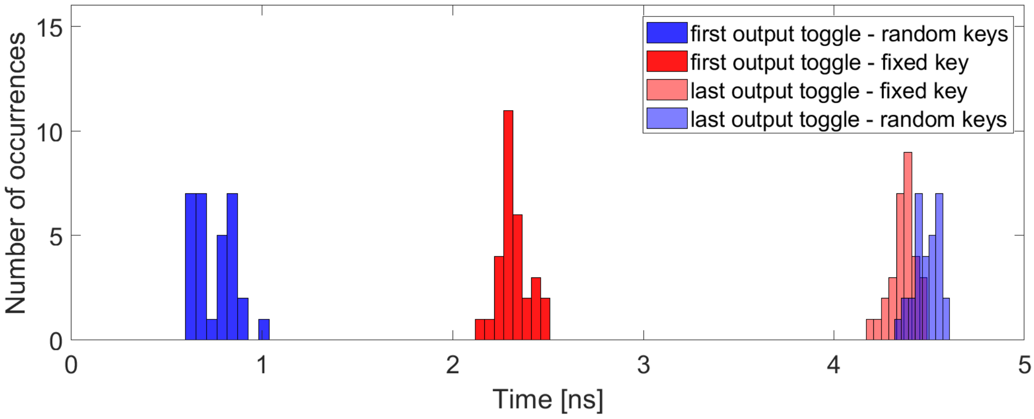


Figure 5: Timeline showing the distribution of first and last output toggles when simulating the unrolled PRINCE netlist for random plaintext transitions and either a fixed key (red) or random key transitions (blue).

4 Experimental Results

In this section we present our practical analysis of the physical security level of unrolled PRINCE implemented on a 40 nm ASIC prototype. At first, we analyze the dynamic dissipation of the circuit in 5 different usage scenarios. Then, we compare those scenarios against each other in terms of security provided and overhead spent. Finally, we present attack results exploiting the static power consumption of the circuit and discuss the dangers of ignoring this security threat.

4.1 Dynamic Power Attacks

Previous analyses of the SCA security provided by unrolled PRINCE implementations have targeted the dynamic power consumption on FPGA platforms. As described in Section 1, an ASIC implementation is expected to have a much higher speed of execution, a higher asynchronicity as well as lower power consumption footprint. All of those differences should increase the difficulty to perform attacks on the primitive, especially in an advanced semiconductor technology node due to the even smaller delays and even lower power consumption per logic unit. In the following experiments, we analyze the level of difficulty to mount successful attacks on our unrolled circuit, while taking different usage principles and adversary models into account. We distinguish a number of cases, as our conclusions about the security level of unrolled PRINCE on silicon entirely depend on how the primitive is instantiated and used.

4.1.1 Measurement Details.

In order to provide a meaningful and fair analysis, we have taken several measures to guarantee the highest possible quality of results in our trace acquisition. First of all, in order to capture the dissipation of a primitive whose execution takes only a couple of nanoseconds (see Figure 5), a high bandwidth and a high sampling rate are required from the oscilloscope that is utilized in the measurement process. In that regard we chose a *Teledyne LeCroy WaveRunner 8254M* [Wav20], which features a bandwidth of 2.5 GHz and a sampling rate of 40 GS/s. The vertical resolution of the scope is 8 bit in normal operation and up to 11 bit with enhanced resolution (ERES) which we used in our measurements. As a next step, we evaluated whether electromagnetic emanation (EM) or power measurements were favorable for the analysis. We found that power measurements led to a higher signal-to-noise ratio (SNR) than EM measurements which were recorded on the front side of the chip directly above the PRINCE core using a *Langer EMV ICR HH150-27* near-field probe with a bandwidth of up to 6 GHz. It may be counterintuitive that the power consumption of our target is supposed to be more informative than the electromagnetic radiation. Typically, when sampling a very short signal which potentially carries a lot of temporal information, EM measurements are the method of choice. Indeed, the precise point in time when a certain glitch occurs may carry valuable information for an adversary. Yet, we could not observe any benefit on our target when performing EM measurements and will shortly discuss the potential reasons for that. First of all, many logic gates in the underlying CMOS node have propagation delays as short as 10-20 picoseconds. Thus, one is naturally limited by the probe's and oscilloscope's bandwidth and sampling rate to adequately capture the timing of individual intermediate transitions. Bandwidths of 50 GHz and beyond would be required to enable the proper sampling of fast glitches in the output lines of such logic gates and common lab equipment rarely supports such high frequencies[4].

Additionally, we know from the simulations presented in Section 3 that up to 115 000

[4] Oscilloscopes with a bandwidth above 50 GHz do exist, e.g., the Teledyne LeCroy LabMaster 10 Zi-A series [Lab20], but come at the cost of multiple hundreds of thousands of dollars.

gate transitions occur in a span of less than 5 nanoseconds when executing the targeted PRINCE core. Thus, at each point of the execution a large number of gate transitions occurs simultaneously, making it difficult to effectively sense minor temporal differences related to the processing of a small part of the circuit. Finally, the active transistor area is covered by 8 metal layers (as seen from the front side of the die) which are densely utilized. While not all routed signals above the targeted area will transport information at the time of measurement, some certainly do. Others are responsible for supplying current to the core area of the chip. Hence, the same voltage fluctuations that are seen in the power measurements will also affect any EM measurements from the front side. In fact, from a purely visual standpoint EM measurements of this target's execution have a similar appearance as power traces and do not show any sharper distinction of different calculation parts. Measurements from the back side, targeting a thinned sample of the ASIC, may potentially improve the quality of results, but we consider this out of scope for this work. Hence, we concentrate on power analysis attacks in the following.

Since the static power results, presented later in this section, were acquired inside a climate chamber at an increased temperature and supply voltage, we examined whether those parameters have an impact on the dynamic power results as well. Our observations are that neither a constant nor a lower or higher temperature led to noticeably improved results, in terms of attack success and SNR. Thus, we recorded all dynamic power traces outside the chamber at room temperature. Increasing the supply voltage from 1.1V to 1.6V (45.45% overvoltage) on the other hand raised the first-round SNR by about 10-15%. Hence, we chose to exploit this effect in our trace acquisition. For completeness, we also evaluated a decreased supply voltage of 0.8V (27.27% undervoltage). A smaller supply voltage in theory leads to higher propagation delays and an overall slower execution of the primitive which may result in a better sampling of the signal. However, the results did not confirm any potential improvement for lower voltages, probably due to the generally decreased dissipation, but showed a reduced SNR by about 7% and a negative effect on the attack success.

4.1.2 No Reset (Highest Performance, Lowest SCA Security).

The most straightforward manner to implement unrolled PRINCE and similar cryptographic primitives does not include any data path reset or key-removal mechanism. Each plaintext or ciphertext input is processed as soon as it arrives and no cleanup of any kind is performed after the cryptographic operation. This is the desired mode of operation when high performance is the primary objective, since it guarantees that each PRINCE instance has a throughput of 64 bits per clock cycle and delivers the result of a new encryption after exactly one clock cycle. However, these deliverables demand that a key is constantly applied to the circuit, unless it is somehow possible to fetch it from a secure storage without any additional latency. Furthermore, it implies that the circuit is not reset to a determined state after an encryption, causing the final state of the previous encryption to remain in the cryptographic circuit until a new plaintext arrives. Hence, each new plaintext causes the circuit to transition from the previous input to the new one, and the dissipation depends on both of those values in the same manner. While an implementation of that kind delivers the overall highest performance, it is the weakest in terms of SCA protection.

In order to demonstrate this, we have measured 500 000 traces for random plaintexts processed by the unrolled PRINCE encryption on the 40 nm ASIC which took less than 15 minutes to acquire. Figure 6 depicts an overlay of 30 sample measurements on the top and the result of a Correlation Power Analysis (CPA) [BCO04] attack on the bottom. The leakage model for the CPA is the Hamming distance (HD) between two first-round Sbox outputs based on consecutively encrypted plaintexts. To be more precise, the model can be expressed as $\mathrm{HD}(\mathrm{S}(p_{i-1,j} \oplus \hat{k}_j), \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$, where $\mathrm{HD}(\cdot)$ is a function calculating the

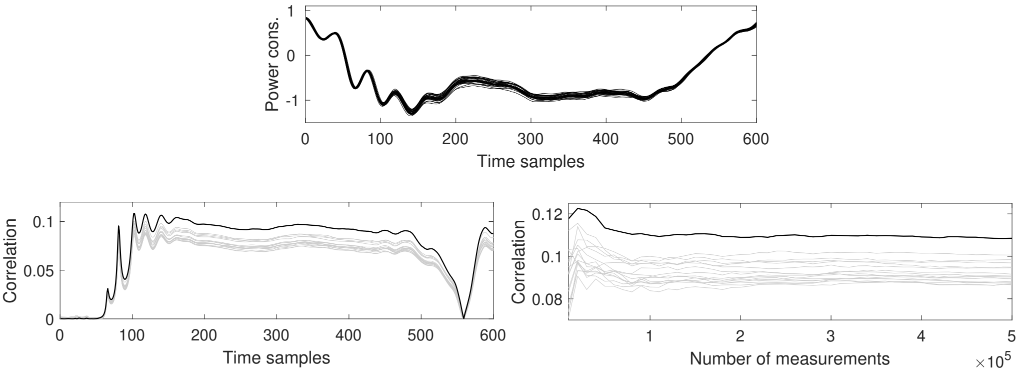


Figure 6: Overlay of 30 sample traces and a CPA attack using the Hamming distance of two first-round Sbox outputs based on consecutive known plaintexts.

Hamming distance, S(·) is the substitution box (Sbox) of the PRINCE cipher and $p_{i,j}$ is the j-th plaintext nibble of the i-th plaintext that is encrypted. The term $\hat{k}_j$ corresponds to the j-th nibble of $\hat{k}$, with $\hat{k}$ being defined as $\hat{k} = k_0 \oplus k_1$. The whitening key k_0 and the round key k_1 are defined in [BCG+12]. This model requires knowledge of consecutively encrypted plaintexts under the same key. When an adversary possesses such knowledge, all key nibbles can be recovered with the available amount of traces. To be more precise, in our experiments, the lowest number of traces required to recover a key nibble was 3 000, the median was 43 000 and the highest number was 350 000. This is already a significant amount of samples required to perform a key recovery on an essentially unprotected implementation. Obviously there are multiple reasons for this observation, including the small power consumption footprint and high speed of the 40 nm ASIC technology, the highly-parallel implementation style and the asynchronicity of the signals. As we will see in the following, the exploitation effort is even much higher without knowledge of consecutively encrypted plaintexts under the same key.

Please note, that we do not consider chosen-input scenarios in this work. It is obvious and has been demonstrated in [YHA17a], that such a scenario allows to increase the signal-to-noise ratio (SNR), especially in the first round(s), significantly by toggling only the targeted input bits instead of encrypting random plaintexts. We also do not consider dedicated filtering and pre-processing approaches (as e.g. presented in [CSR+19]) or template attacks in this work. Instead, we concentrate on more generic evaluation metrics in this work in order to keep the analysis as universally valid as possible and make as few assumptions about an attackers capabilities as possible. We are aware that for any concrete implementation a highly specialized and optimized attack method will usually outperform such generic approaches. Yet, this is a valid statement for all attacks (dynamic and static) presented in this work and should not significantly impact the interpretation of our comparison. Also, we would like to stress that we did not perform a TVLA analysis [GJJR11, SM15] on the measurements in the *no reset* scenario, as this is not straightforward. Without a reset between encryptions, the dissipation depends not only on the currently encrypted plaintext, but also on the previous one. Since the measurements for fixed and random plaintexts in a non-specific t-test should be recorded in a randomly interleaved fashion [SM15], the following 4 transition groups would occur, (i) fix → fix, (ii) fix → ran, (iii) ran → fix, (iv) ran → ran. Hence, 4 different distributions would need to be distinguished instead of the usual 2 and the transition from fixed to fixed would always have zero dissipation, as no gate toggles are caused. Comparing only the two groups (iii) ran → fix and (iv) ran → ran is no solution for this problem either, as it would lead to a false sense of security. The random component in the (iii) ran → fix transition group is artificially introduced by the methodology and would not be a limiting factor for the adversary in an actual attack. For this reason, we refrain from defining a new TVLA methodology for this special case. However, for all experiments presented in the following we are able to provide TVLA results due to the reset methods applied.

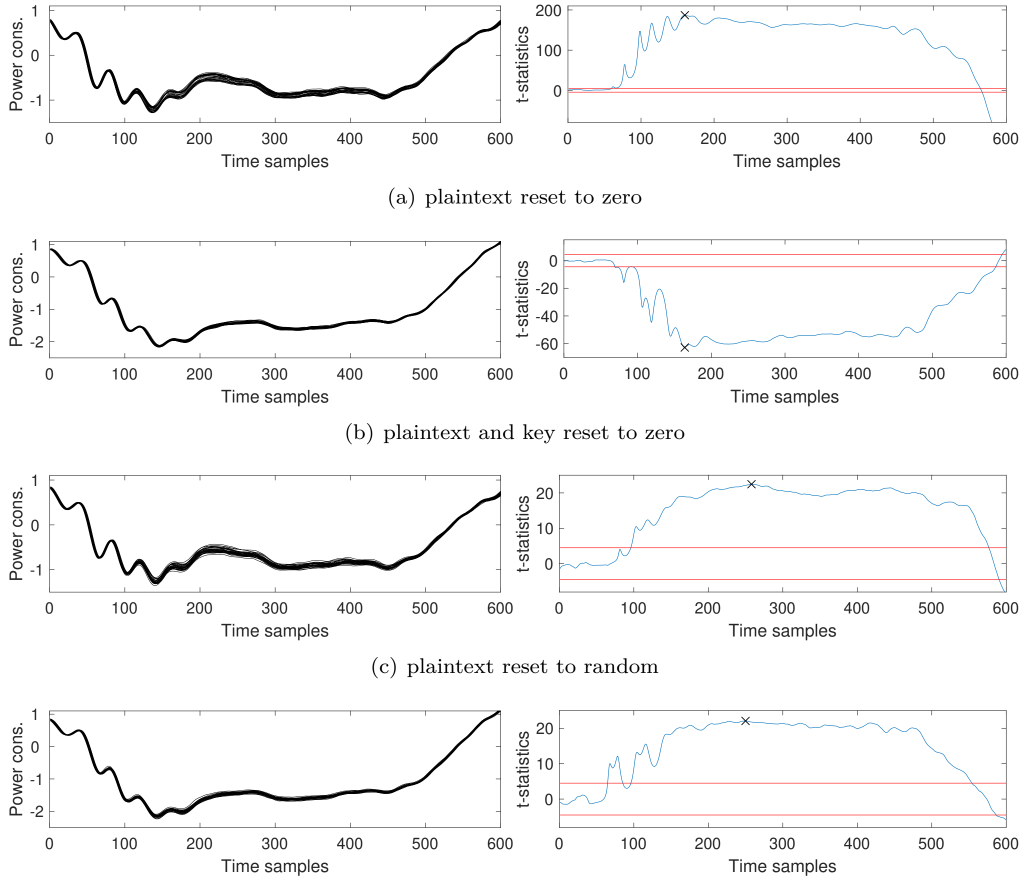
4.1.3 Reset Methods (Lower Performance, Higher SCA Security).

As our example on the toy cipher in Section 2 has demonstrated, the dynamic power consumption of an unrolled cipher during an encryption substantially depends on the initial state of the circuit. In particular, the leakage function of the circuit is mainly transitional. An attacker who obtains knowledge of consecutive inputs (processed under a fixed key) can easily correlate the dissipation of the circuit to his transitional hypothesis under the correct key guess. However, the authors of [BGSD10] claimed that *clearing* the data path between encryptions successfully prevents an adversary from performing such an attack. It is unclear from [BGSD10] how exactly such a reset (i.e., clearing) procedure should be executed, although it is described as *"propagating random values without interference from the key"*. In our experiments we test 4 different methods which come at different costs in terms of required randomness per cycle, total power consumption and delay. Yet, they all halve the available throughput, since useful data can only be encrypted every other clock cycle. Hence, in order to achieve the same performance as an unrolled primitive without reset method applied, twice as many unrolled PRINCE instances need to be implemented. This overhead is not unacceptable when compared to other side-channel countermeasures, such as dual-rail logic or masking. The 4 different reset strategies that we evaluate in the following are:

1. resetting the plaintext to zero
2. resetting the plaintext and the key to zero
3. resetting the plaintext to a random value
4. resetting the plaintext and the key to a random value

As described before, we are able to provide TVLA results in all 4 of these scenarios. Figure 7 shows an overlay of 30 sample traces on the left and results of a non-specific t-test for 10 000 randomly interleaved measurements for fixed and random inputs on the right. It can be observed that the voltage drop which is measured in the scenarios where both, the key and the plaintext, are reset is significantly larger. Figure 8 depicts the histograms of the fixed and random groups at the most leaking time sample (marked by an 'x' in Figure 7). It also shows the evolution of the maximum t-value over the number of traces on the right side for all 4 scenarios. Although the t-test reports leakage with a high confidence in all 4 scenarios, clear differences in the statistic's magnitude can be observed in Figure 7. Yet, the absolute magnitude of the t-value in a leakage detection scenario only expresses the confidence that the two input distributions can be distinguished (by their means in case of a first-order t-test) and can not be used as an assessment of the security level of an implementation. However, the histograms and the number of traces required to reach a certain t-value, as shown in Figure 8, confirm that the concrete fixed and random distributions shown here are more difficult to distinguish in case of the random reset scenarios. In order to asses the actual security level provided by the different usage scenarios we perform key recovery attacks in the following.

We chose to apply two different classes of key recovery attacks here, first Correlation Power Analysis (CPA) based on a power model and second collision-based SCA attacks which are independent of leakage models. As power models for the CPA we have tested the Hamming weight of Sbox outputs, the Hamming distance between Sbox outputs (of two consecutive encryptions) and all corresponding single-bit models (transition- and value-based). As



(a) plaintext reset to zero

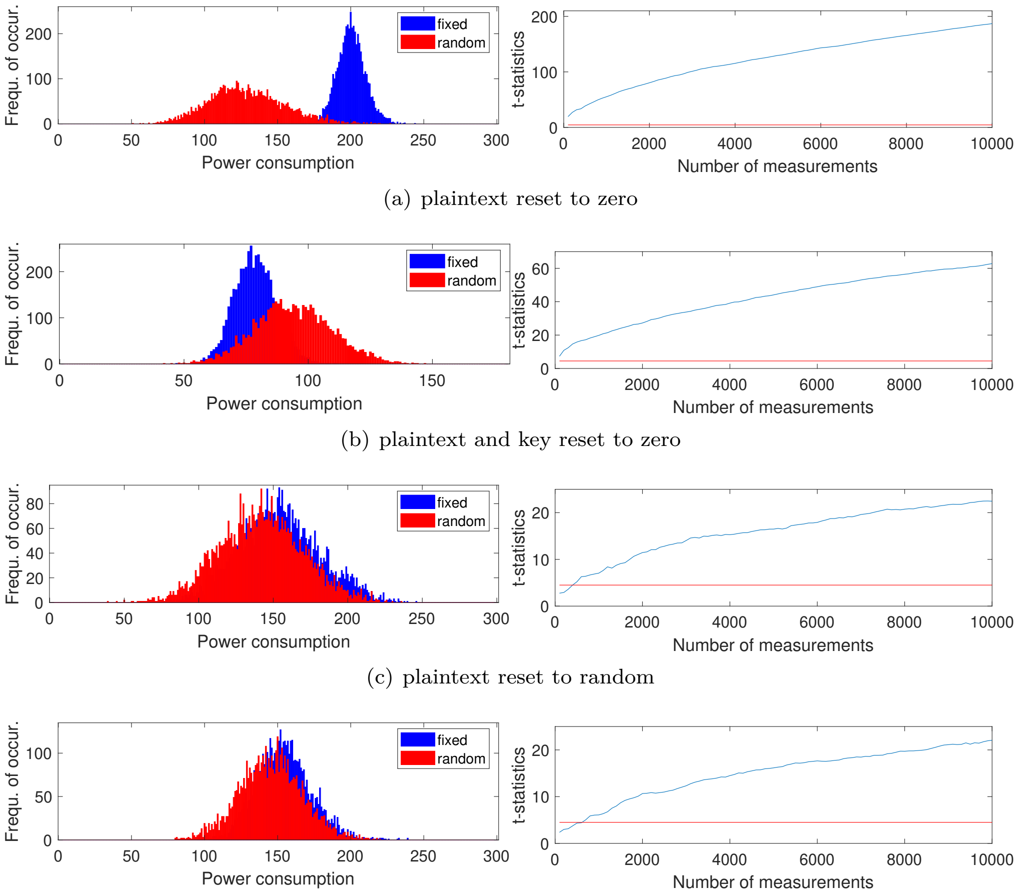
(b) plaintext and key reset to zero

(c) plaintext reset to random

(d) plaintext and key reset to random

Figure 7: Overlay of 30 sample traces and t-test results using 10 000 traces each for 4 different reset methods.

collision-based SCA attacks we have evaluated the leakage-model-independent approaches introduced as Correlation-Enhanced Collision in [MME10] and Moments-Correlating DPA (MCDPA) in [MS16b]. Table 7 states the most successful of our tested attacks for all 5 scenarios (including *no reset*). The collision-based SCA attacks recovered less information about the key than simple CPA. This is reasonable, as collision-based SCA attacks are based on the assumption that a module is (time-)shared between multiple computations, or, in the parallel case, that multiple physical instances of a module have similar leakage characteristics. In our standard-cell-based unrolled PRINCE circuit, all Sboxes are realized as a unique composition of gates and therefore are expected to have different power characteristics and time of evaluation depending on the input transition. Thus, the circuit does not meet the requirement to successfully apply collision-based SCA, at least not to recover significant portions of the key. This is already a major difference compared to FPGA-based results, where collision-based attacks were shown to be effective [MMP11]. According to Table 7, the *plaintext reset to zero* already provides an increased security level compared to the *no reset* scenario. The leakage function depends on fewer variable inputs that are predictable for the adversary, which makes the attack less powerful. To be more precise, the power model reported in Table 7 for the *no reset* scenario depends on $p_{i-1,j}$ and $p_{i,j}$, while for the *plaintext reset to zero* case $p_{i-1,j}$ is replaced by constant 0. As there are 16 possible values each for $p_{i-1,j}$ and $p_{i,j}$, the adversary can distinguish



(a) plaintext reset to zero

(b) plaintext and key reset to zero

(c) plaintext reset to random

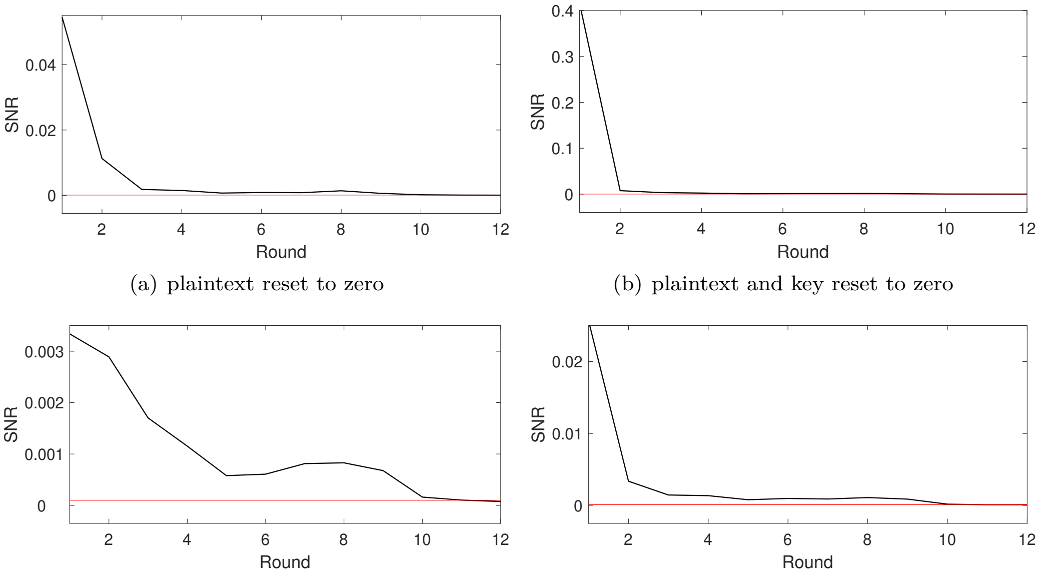
(d) plaintext and key reset to random

Figure 8: Histograms of the fixed and random groups at the most leaking time sample and development of the maximum t-value over the number of traces for 4 different reset methods.

$16 \cdot 16 = 256$ cases in the *no reset*, but only 16 in the *plaintext reset to zero* scenario. This is sufficient to reduce the number of recovered nibbles by more than the half. However, the encryption is still fully deterministic, as for any other common unprotected block cipher implementation.

The method where both, the key and the plaintext, are reset to zero apparently provides further security. This can be explained by the noise that is induced due to the fact that all rounds receive the round key at the same time at the start of the encryption. Hence, all rounds begin to toggle before the new plaintext even propagated that far (see Section 3). When the random resets are used, the encryption is non-deterministic, which results in a much higher noise level, as reflected in the t-test and attack results. Also, the adversary can not easily make transitional hypotheses anymore. In these cases, a significant part of the leakage function is unknown to the adversary. It still has some value-based leakage component, thus the detectable first-order leakage, but the evaluation shows that most key nibbles in the first round can not be recovered.

For a final comparison between the 4 different reset methods we evaluated the maximum signal-to-noise ratio (SNR) based on the 16 input nibbles of each of the 12 block cipher rounds. The results are depicted in Figure 9. The first observation that has to be made is that the SNR degrades very quickly after the first round for all 4 methods. This is caused



(a) plaintext reset to zero

(b) plaintext and key reset to zero

(c) plaintext reset to random

(d) plaintext and key reset to random

Figure 9: Maximum (nibble wise) signal-to-noise ratio (SNR) computed for all 12 round inputs for 4 different reset methods.

by the nature of unrolling. As already demonstrated by the toy example in Section 2 and the simulation results in Section 3, logic gates in later rounds are evaluated at completely different moments in time for different input transitions. The signals arrive at a different time at the gates depending on which path they have taken. This is what we call the asynchronicity of signals. Additionally, since the gates corresponding to the last two rounds are located towards the end of the circuit, their computational result does not affect as many further gates as the output of earlier rounds. The red line in Figure 9 is the border for statistical insignificance, which we experimentally determined as 0.0001. In all 4 cases the maximum SNR for round inputs 11 and 12 is below this threshold, indicating that an attack on the last two rounds is not expected to succeed given the available amount of traces. Nevertheless, we attempted different attacks from the ciphertext side, but indeed none were successful. Another interesting observation that can be made in Figure 9 is that the methods where the key is reset as well have a significantly higher first-round SNR. As reported in Section 3, the amount of gate toggles caused when changing the key in addition to the plaintext is more than twice as large. This can also be observed in the larger voltage drop in Figure 7. Taking all evaluation metrics into account (TVLA, CPA, SNR) we come to the conclusion that the *random plaintext reset* is the most preferable choice. It delivers the best SCA security and is cheaper than the *random key and plaintext*

Table 7: Summary of the optimal attack results (among all tested ones) for the 5 dynamic power scenarios respectively with a maximum of 500 000 traces.

Reset Type	Attack	Best Power Model Found	Rec. Nib.
no reset	CPA	$\mathrm{HD}(\mathrm{S}(p_{i-1,j} \oplus \hat{k}_j),\, \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	16/16
plain zero	CPA	$\mathrm{HD}(\mathrm{S}(0 \oplus \hat{k}_j),\, \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	7/16
plain and key zero	CPA	$\mathrm{HD}(\mathrm{S}(0 \oplus 0),\, \mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	5/16
plain random	CPA	$\mathrm{HW}(\mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	2/16
plain and key random	CPA	$\mathrm{HW}(\mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	3/16

reset, since it consumes a less power and requires only a third of the randomness per clock cycle (32 instead of 96 bit).

In order to provide a benchmark with respect to the security level that the *random plaintext reset* scenario provides we have measured 100 million traces which took about 24 hours. Then we conducted the same CPA attack that proved to be most successful for the smaller amount of traces (see Table 7) on all key nibbles and obtained the results depicted in the Appendix in Figure 13. The first row shows the results targeting key nibbles 0 to 4, while the last row corresponds to nibbles 12 to 15. Surprisingly, the attack with 100 million traces is not any more successful than with 500 000 traces, as also 2 key nibbles can be recovered (numbers 1 and 15)[5]. We also attempted single-bit models, but none of them succeeded on more than two nibbles either. For the sake of completeness we finally performed collision-based Moments-Correlating DPA (MCDPA) [MS16b] on the 100 million traces with an offset of 0. The results are depicted in the Appendix in Figure 14. Here, the first row shows the results targeting key differences 0-1 to 3-4, while the last row corresponds to differences 12-13 to 15-0. As shown in the figure, only 1 key difference can correctly be recovered[6].

In conclusion, the *random plaintext reset* is a viable and effective protection against side-channel attacks targeting the dynamic circuit emanations of unrolled primitives. In our case study it was not possible to extract a notable portion of the key even with a huge amount of available traces. Furthermore, even if a larger part of the key could be extracted from the first round, an attacker would still need to perform further attacks with a deeper hypothesis into the second round or target the last round. Both strategies have small likelihood of success based on the acquired SNR results. This analysis shows that unrolled cryptography on silicon can provide a high level of resistance against dynamic power SCA attacks at a low cost, if certain usage principles are carefully respected.

4.2 Static Power Attacks

As a next step, we analyze the susceptibility of unrolled ciphers in state-of-the-art ASIC technology towards attacks exploiting the static power consumption. Static power side-channel analysis (SPSCA) has been a growing field in recent years due to its emergence in nanometer-scaled CMOS technologies [Moo19]. Its nature is entirely different from dynamic power analysis, since it does not exploit a momentary transitional effect that can be observed for a finite period of time only. Instead, it is based on observing a static phenomenon that can be quantified for as long as no transition occurs in the targeted circuit part. As demonstrated on the toy example in Section 2, the static power consumption is fully independent of a potential previous state of the circuit and exhibits a deterministic leakage behavior for any given input. Hence, the aforementioned tricks and usage principles are not expected to be effective against this kind of adversary. In the end of our analysis, we discuss under what circumstances one source of information leakage is preferable over the other (from an adversarial standpoint) and which guidelines need to be observed to provide protection against both.

4.2.1 Measurement Details.

Our setup for the static power side-channel attacks differs from the one used for the dynamic power experiments in several regards. Most notably, we have used a different oscilloscope. Since a high bandwidth and sampling rate are not primarily important for static power measurements, we rather chose a scope that has a high vertical resolution.

[5] Of course, it remains unclear whether these 2 successful recoveries are indeed reliable or simply a consequence of the probability of 1/16 for each key candidate to produce the highest correlation when no significant correlation is found for any candidate.

[6] As for the CPA, this may well be a statistical incident and no true recovery.

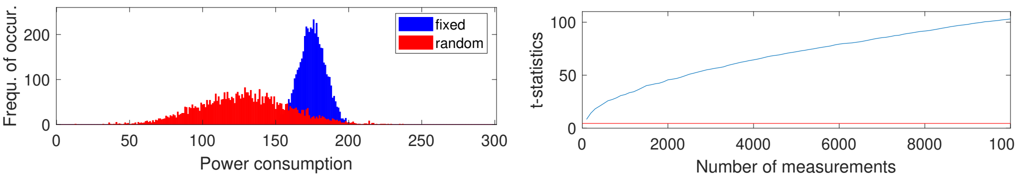


Figure 10: Histograms of the fixed and random groups and development of the maximum t-value over the number of traces for the static power measurements.

In particular, we have used a *Teledyne LeCroy HRO 66Zi* [HRO20], which features a bandwidth of 600 MHz, a sampling rate of 2 GS/s and a vertical resolution of 12 bit in normal operation and up to 15 bit with enhanced resolution (ERES). Furthermore, due to the leakage-enhancing effects of higher temperatures reported in [Moo19, MMR20], we have performed the measurements in a climate chamber at 90°C. The core supply voltage has been set to 1.6V instead of the nominal 1.1V (45.45% overvoltage), just as for the dynamic power measurements. Additionally, our setup utilizes a DC amplifier and a low-pass filter, similar to what has been suggested in [MMR20].

4.2.2 Results.

In the following, we apply the same evaluation metrics that have been introduced for the dynamic power analysis. First of all, we have measured the static power consumption of the circuit for a randomly interleaved sequence of 10 000 fixed and random inputs in order to perform a non-specific t-test. The results are shown as a histogram on the left and as t-value over the number traces on the right side in Figure 10. It is important to note that the measurements have been recorded while the global clock of the ASIC was active and other unrelated computations have been executed on the chip. In particular, an LFSR-based PRNG was running during the measurement and constantly computed on random values. We introduced this additional obstacle in order to clarify that there is not *necessarily* any relevant increase in the difficulty of measuring the static power consumption just because other parts of the circuit are consuming dynamic power at the same time[7]. This has already been observed in [Moo19] and was confirmed by our experiments. It is only required that the unrolled circuit itself is idle during the measurements and contains the sensitive intermediates. In this regard, it becomes clear that an unrolled circuit without a reset method or key-removal mechanism is susceptible to static power attacks, even if the adversary can not influence the clock signal of the circuit. Before moving to key recovery attacks, we take a look at the signal-to-noise ratio for all 12 round inputs. The result is compared to the dynamic power scenarios in Figure 11. It can be observed that the SNR does not degrade significantly after the first round, much unlike the dynamic scenarios. It stays approximately in the same range for all 12 round inputs. Furthermore, the SNR is consistently higher than in all dynamic power measurements for rounds 2-12. Even in the first round, the static power SNR is significantly larger than that for the *random plaintext reset* scenario. Clearly, the static power side-channel leakage contains more information about the later stages of the combinatorial circuit than the dynamic dissipation. In this regard we are able to perform attacks from the plaintext *and* the ciphertext side. Two exemplary results are depicted in Figure 12 and their success is compared in Table 8. Acquiring the 500 000 traces took approximately 70 hours. While the trace acquisition is orders of magnitude slower than for the dynamic power measurements in our laboratory experiments, the difference is expected to be significantly smaller when examining real

[7] Clearly, this does not mean that any kind of unrelated parallel workload is irrelevant for the attack success. Many counterexamples can be imagined. It simply means that it is no strict requirement that the whole chip is idle during measurement.

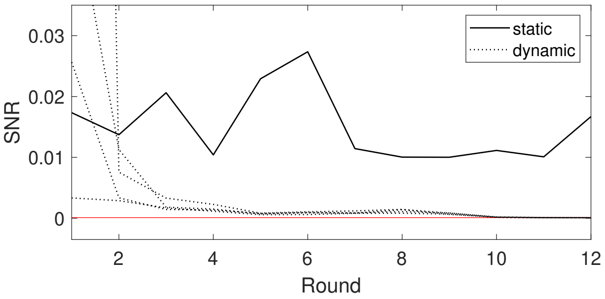
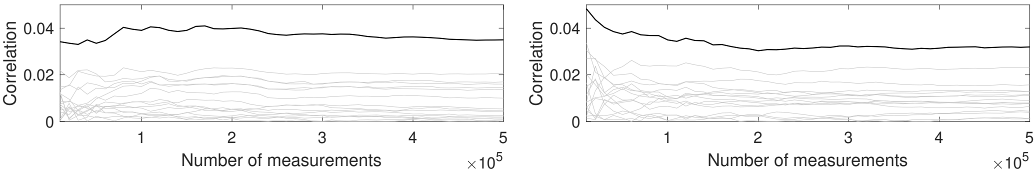


Figure 11: Maximum (nibble wise) signal-to-noise ratio (SNR) computed for all 12 round inputs for the static power measurements.



(a) first round

(b) last round

Figure 12: Two CPA attacks on the static power measurements, one on the first round of PRINCE using the LSB of the Sbox output and one on the last round of PRINCE using the LSB of the inverse Sbox input as a leakage model.

world devices. In our experience it is often not possible to acquire hundreds of millions of dynamic power traces per day when analyzing an actual product for its physical security. Additionally, we have not attempted to optimize the static power measurements in terms of measurement time in these experiments. Previous articles have outlined the trade-off between time interval spent and the quality of static power measurements [MMR20].
The CPA with ciphertext knowledge on the last round leads to the best results. The lowest number of traces required to recover a key nibble is 1 000, the median is 21 000 and the highest number is 420 000. 14 key nibbles can be recovered with less than 50 000 traces. In contrast to the dynamic power analysis, these numbers are independent of a potentially applied reset method and independent on having access to plaintexts. Hence, especially in the somewhat *protected* use cases, the static power consumption is a much more severe threat to the SCA security than the dynamic power. While a fully unprotected (i.e., *no reset*) unrolled implementation of PRINCE still provides high protection against dynamic power analysis attacks on the last rounds (with ciphertext-only knowledge), this is not true against static power adversaries, even if clock control is not an option[8]. Furthermore, in case clock control *is* obtained by a static power adversary, all reset methods are conceptually ineffective to thwart attacks. The adversary can simply stop the clock whenever a user-supplied input is applied to the combinatorial circuit (i.e., before the reset is performed) and measure its static leakage without any influence from a previous state. However, when the random reset of the circuit between each two encryptions is performed immediately in the next clock cycle after each valid encryption and the attacker can not influence the clock signal, SPSCA attacks are not informative and an adversary has to rely on the dynamic currents to extract information.

[8] Unless the circuit is never idle and encrypts data all the time. Then, any kind of SPSCA fails.

Table 8: Summary of the optimal attack results (among all tested ones) for the static power scenario with a maximum of 500 000 traces.

Round	Attack	Best Power Model Found	Rec. Nib.
first	CPA	$\mathrm{LSB}(\mathrm{S}(p_{i,j} \oplus \hat{k}_j))$	15/16
last	CPA	$\mathrm{LSB}(\mathrm{S}(c_{i,j} \oplus \hat{k}'_j))$	16/16

5 Conclusion

In this work we have analyzed the physical security level that can be provided by the unrolled low-latency cipher PRINCE when it is implemented in state-of-the-art semiconductor technology. We have realized the primitive in a fully round-unrolled fashion in a 40 nm CMOS node as a semi-custom standard-cell-based design with a latency of less than 5 nanoseconds. Our observations regarding its vulnerability are manifold. First of all, performing a *full* key-recovery attack (revealing all 128 key bits) on an unrolled ASIC implementation of PRINCE is always hard when observing its dynamic behavior. Even in the best case for the adversary it is difficult to extract more than 64 bit of information about the key. The extremely fast execution, the high level of parallelism and its asynchronicity make the life of adversaries difficult. Recovery of even small parts of the key requires a huge amount of observations when the adversary can not obtain the encrypted plaintexts, but rather is in possession of the ciphertexts only[9]. It is also extremely challenging to extract key parts when the state of the combinatorial encryption circuit is reset to a value unpredictable for the adversary between any two user-driven encryptions. This can be achieved by propagating a random plaintext through the circuit, while leaving the key constant, and simply ignoring the output of the computation. The cost of this method is obviously that the throughput is halved, which can be compensated by putting twice as many instances on the chip. Furthermore, it requires 64 bits of randomness every other clock cycle (i.e., 32 bits per cycle), which need to be generated by a PRNG. In such a case, only the currently encrypted plaintext is known to the attacker, while the previous state of the circuit can not be predicted. In that case, even encrypting the same plaintext twice leads to two vastly different power consumption (or electromagnetic emanation) footprints due to the difference in the initial state of the circuit. In other words, the encryption engine has a non-deterministic behavior and dissipation, which leads to a high level of protection. In our experiments, straightforward first-round attacks could not recover the key with up to 100 million traces.

However, there exists another attack vector with a remarkable impact on the physical security of unrolled cryptography on silicon, namely the static power consumption. Our results indicate that the static power side channel is a convenient source of information leakage for adversaries against unrolled cryptographic primitives in advanced technologies. Its independence of the execution speed, asynchronicity and glitching behavior of the circuit is a favorable advantage that leads to effective attacks. In our experiments, targeting the static power was clearly the best choice when trying to extract the full 128-bit key. Any round of the block cipher can be targeted with roughly the same effort and reset methods are useless if the adversary can stop the clock signal of the input register feeding the combinatorial circuit. The static dissipation is always deterministic, due to its independence of any previous state of the circuit. Hence, the increased security level achieved by certain usage principles does not translate to the static power side channel due to its different nature. The best chance to thwart this kind of attack is to ensure that the clock signal of the unrolled primitive is generated on silicon and can not be stopped without causing the circuit to lose its state. Yet, this is not always an option. Even without control over the

[9] This is only true when the adversary can not observe the physical leakage of the decryption of the ciphertext on the communication partner's side too.

clock, static power adversaries remain dangerous. To protect implementations one should always ensure that a reset of the full circuit is performed immediately in the next clock cycle after the result of a previous operation has been saved in order to not leave sensitive information behind. Only if that is guaranteed, the implementation can provide reasonable security against this type of attacker. Nevertheless, an adversary with full physical access to the target should never be underestimated. It is unclear whether possibilities exist to stop the propagation of the clock signal to a targeted cipher core, even when the oscillator is implemented on silicon and precautions are in place.

Acknowledgments

The author would like to thank the TCHES 2020 reviewers for their constructive feedback and Amir Moradi for his input and support. The work described in this paper has been supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972 and through the project 271752544 "NaSCA: Nano-Scale Side-Channel Analysis".

References

[AO14] Zia Abbas and Mauro Olivieri. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. *Microelectronics Journal*, 45(2):179–195, 2014.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 208–225. Springer, 2012.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *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 16–29. Springer, 2004.

[BCS+17] Davide Bellizia, Danilo Cellucci, Valerio Di Stefano, Giuseppe Scotti, and Alessandro Trifiletti. Novel measurements setup for attacks exploiting static power using DC pico-ammeter. In *2017 European Conference on Circuit Theory and Design, ECCTD 2017, Catania, Italy, September 4-6, 2017*, pages 1–4. IEEE, 2017.

[BGSD10] Shivam Bhasin, Sylvain Guilley, Laurent Sauvage, and Jean-Luc Danger. Unrolling cryptographic circuits: A simple countermeasure against side-channel attacks. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 195–207. Springer, 2010.

[CRB+16] Thomas De Cnudde, Oscar Reparaz, Begül Bilgin, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Masking AES with d+1 shares in hardware. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 194–212. Springer, 2016.

[CSR+19] Nikhil Chawla, Arvind Singh, Nael Mizanur Rahman, Monodeep Kar, and Saibal Mukhopadhyay. Extracting side-channel leakage from round unrolled implementations of lightweight ciphers. In *IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2019, McLean, VA, USA, May 5-10, 2019*, pages 31–40. IEEE, 2019.

[FGP+18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable masking schemes in the presence of physical defaults & the robust probing model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):89–120, 2018.

[GIB18] Hannes Groß, Rinat Iusupov, and Roderick Bloem. Generic low-latency masking in hardware. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(2):1–21, 2018.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for sidechannel resistance validation. In *NIST non-invasive attack testing workshop*, 2011. https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/08_goodwill.pdf.

[GM17] Hannes Groß and Stefan Mangard. Reconciling d+1 masking in hardware and software. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 115–136. Springer, 2017.

[GM18] Hannes Groß and Stefan Mangard. A unified masking approach. *J. Cryptographic Engineering*, 8(2):109–124, 2018.

[GMK16] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TISCCS 2016 Vienna, Austria, October, 2016*, page 3. ACM, 2016.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In Helena Handschuh, editor, *Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Francisco, CA, USA, February 14-17, 2017, Proceedings*, volume 10159 of *Lecture Notes in Computer Science*, pages 95–112. Springer, 2017.

[HRO20] *Teledyne LeCroy HRO Series Data Sheet*. http://cdn.teledynelecroy.com/files/pdf/hro-12bit_datasheet.pdf, accessed July 5th, 2020.

[KMM19] Naghmeh Karimi, Thorben Moos, and Amir Moradi. Exploring the effect of device aging on static power analysis attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):233–256, 2019.

[KR07] Ian Kuon and Jonathan Rose. Measuring the gap between fpgas and asics. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 26(2):203–215, 2007.

[Lab20] *Teledyne LeCroy HRO Series Data Sheet.* http://cdn.teledynelecroy.com/files/pdf/labmaster-10zi-a-datasheet.pdf, accessed July 5th, 2020.

[LvMJ95] Jeroen A. J. Leijten, Jef L. van Meerbergen, and Jochen A. G. Jess. Analysis and reduction of glitches in synchronous networks. In *1995 European Design and Test Conference, ED&TC 1995, Paris, France, March 6-9, 1995*, pages 398–403. IEEE Computer Society, 1995.

[MME10] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2010.

[MMP11] Amir Moradi, Oliver Mischke, and Christof Paar. Practical evaluation of DPA countermeasures on reconfigurable hardware. In *HOST 2011, Proceedings of the 2011 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 5-6 June 2011, San Diego, California, USA*, pages 154–160. IEEE Computer Society, 2011.

[MMR17] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis of a threshold implementation prototype chip. In David Atienza and Giorgio Di Natale, editors, *Design, Automation & Test in Europe Conference & Exhibition, DATE 2017, Lausanne, Switzerland, March 27-31, 2017*, pages 1324–1329. IEEE, 2017.

[MMR20] Thorben Moos, Amir Moradi, and Bastian Richter. Static power side-channel analysis - an investigation of measurement factors. *IEEE Trans. VLSI Syst.*, 28(2):376–389, 2020.

[MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-resistant masking revisited or why proofs in the robust probing model are needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019.

[Moo19] Thorben Moos. Static power SCA of sub-100 nm CMOS asics and the insecurity of masking schemes in low-noise environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards.* Springer, 2007.

[Mor14] Amir Moradi. Side-channel leakage through static power - should we care about in practice? In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, volume 8731 of *Lecture Notes in Computer Science*, pages 562–579. Springer, 2014.

[MS16a] Amir Moradi and Tobias Schneider. Side-channel analysis protection and low-latency in action - - case study of PRINCE and midori -. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 517–547, 2016.

[MS16b] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016*, pages 5–15. ACM, 2016.

[NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[PSKM15] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: when should we care? In Wolfgang Nebel and David Atienza, editors, *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition, DATE 2015, Grenoble, France, March 9-13, 2015*, pages 145–150. ACM, 2015.

[RBN$^+$15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating masking schemes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology - A clear roadmap for side-channel evaluations. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.

[Wav20] *Teledyne LeCroy WaveRunner 8000 Series Data Sheet.* http://cdn.teledynelecroy.com/files/pdf/waverunner8000-datasheet.pdf, accessed July 5th, 2020.

[YHA15] Ville Yli-Mäyry, Naofumi Homma, and Takafumi Aoki. Improved power analysis on unrolled architecture and its application to PRINCE block cipher. In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*, volume 9542 of *Lecture Notes in Computer Science*, pages 148–163. Springer, 2015.

[YHA17a] Ville Yli-Mäyry, Naofumi Homma, and Takafumi Aoki. Chosen-input side-channel analysis on unrolled light-weight cryptographic hardware. In *18th International Symposium on Quality Electronic Design, ISQED 2017, Santa Clara, CA, USA, March 14-15, 2017*, pages 301–306. IEEE, 2017.

[YHA17b] Ville Yli-Mäyry, Naofumi Homma, and Takafumi Aoki. Power analysis on unrolled architecture with points-of-interest search and its application to PRINCE block cipher. *IEICE Transactions*, 100-A(1):149–157, 2017.

A Appendix

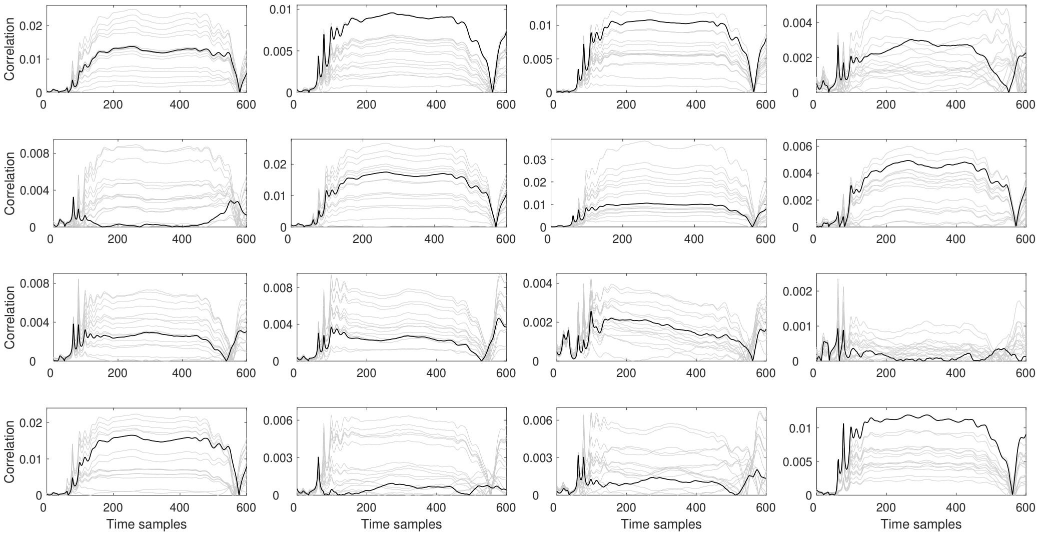


Figure 13: CPA attack on unrolled PRINCE using 100 million traces when the plaintext is reset to a random state between encryptions. Results for all 16 key nibbles are presented and the power model is the Hamming weight of the first-round Sbox output.

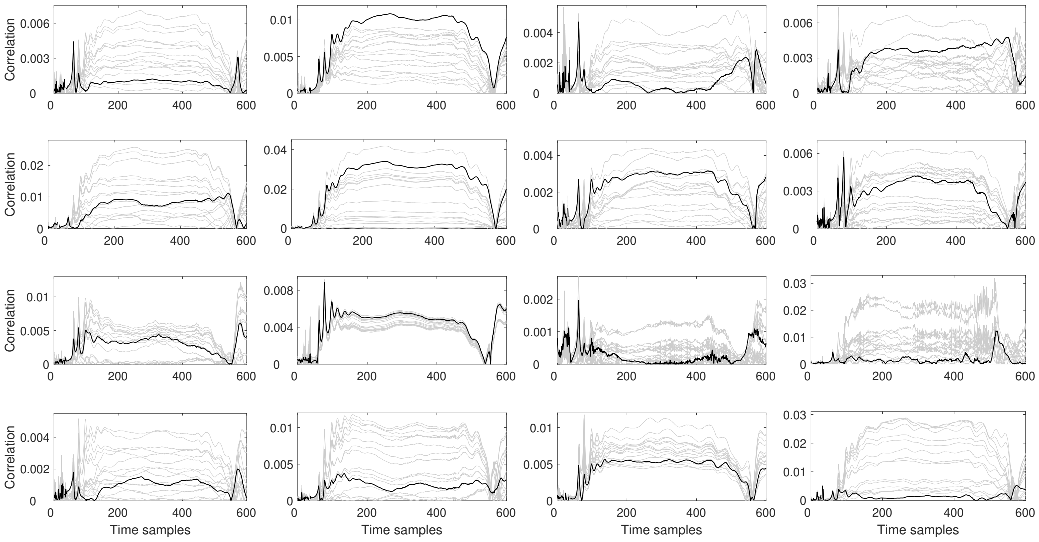


Figure 14: MCDPA attack on unrolled PRINCE using 100 million traces when the plaintext is reset to a random state between encryptions. Results for 16 key differences (0-1, 1-2, ..., 14-15, 15-0) are presented.