IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2020, No. 3, pp. 121–146. DOI:10.13154/tches.v2020.i3.121-146

CPAmap: On the Complexity of Secure FPGA Virtualization, Multi-Tenancy, and Physical Design

Jonas Krautter, Dennis Gnad and Mehdi Tahoori

Karlsruhe Institute of Technology, Germany
{jonas.krautter,dennis.gnad,mehdi.tahoori}@kit.edu

Abstract. With virtualized Field Programmable Gate Arrays (FPGAs) on the verge of being deployed to the cloud computing domain, there is a rising interest in resolving recently identified security issues. Those issues result from different trusted and untrusted entities sharing the FPGA fabric and the Power Distribution Network. Researchers were able to perform both side-channel and fault attacks between logically isolated designs on the same FPGA fabric, compromising security of cryptographic modules and other critical implementations. Side-channel attacks specifically are enabled by the vast degree of freedom given to developers when making use of the basic FPGA resources. Both ring oscillators as well as long delay lines, implemented using low-level FPGA primitives, have been shown to provide sufficient data for simple or correlation-based power analysis attacks. In order to develop new or apply known countermeasures onto designs and implementations in a virtualized multi-tenant FPGA, we seek to fully understand the underlying mechanisms and dependencies of chip-internal side-channel attacks. Although the impact of process variation and other physical design parameters on side-channel vulnerability has been investigated in previous works, remote attacks between logically isolated partitions in multi-tenant FPGAs introduce new and unique challenges. Thus, we systematically analyze the impact of physical mapping of both attacker and victim design on the success of correlation power analysis attacks on the Advanced Encryption Standard (AES). We report our findings on a Xilinx Zynq 7000-based platform, which show that the effect of global and local placement as well as routing and process variation on the success of side-channel attacks almost exceeds the impact of hiding countermeasures. This result reveals fundamental challenges in secure virtualization of FPGAs, which have been mostly ignored so far. Eventually, our results may also help vendors and hypervisors in developing zero overhead side-channel countermeasures based on adequate global and local placement of isolated designs on a multi-tenant FPGA.

Keywords: FPGA · Virtual · Cloud · Multi-Tenant · Side-Channel Attack · Power Analysis · Internal Sensor

1 Introduction

Field Programmable Gate Arrays (FPGAs) have become an important component for all major cloud computing providers as a generic, versatile hardware accelerator, particularly in the area of artificial intelligence and machine learning [AWS19, Cor17, CLO19]. Currently, the accelerator hardware in public clouds is – to the best of our knowledge – not distributed to users on a finer granularity than allowing one user per chip. However, major operating systems already support partial reconfiguration [Lin18] and datacenter-specific FPGAs have been offered just recently [Xil19]. Virtualization and multi-tenancy will allow for maximum

Received: 2020-01-15 Accepted: 2019-03-15 Published: 2020-06-19

FPGA utilization, as widely discussed in recent publications [FVS15, BSB+14, KLP+18]. Unfortunately there still remain unresolved security issues [SGMT18b, KGT18].

Ever since the seminal work of Kocher et. al. in 1999 [KJJ99], power analysis side-channel attacks continue to thwart the security of implementations, that are otherwise secure against cryptanalysis. These attacks are based on the analysis of system and environment variables, which are influenced by the data and computations taking place in the system. Next to power, electromagnetic emanation (EM), supply voltage, heat or photon emission are just a few examples of characteristics, that can be observed by an attacker to learn about intermediate computations and data. Traditionally, power analysis attacks had to be carried out by a local attacker with extensive measurement equipment.

Through unintended use of basic FPGA primitives, researchers were recently able to carry out both fault and side-channel attacks between logically isolated designs on a single FPGA chip [SGMT18b, ZSZF13, ZS18, KGT18, RPD+18]. Those attacks become feasible due to all design components in the chip being connected through a shared Power Distribution Network (PDN), even if logically separated. In several publications [SGMT18b, ZSZF13, ZS18], the behaviour of Time-to-Digital Converter (TDC) sensors based on long delay lines or Ring Oscillators (ROs) was exploited to estimate voltage fluctuations on the PDN and successfully perform side-channel attacks on AES and RSA implementations.

Whereas fault attacks can at least be detected on both hypervisor and victim side, side-channel attacks impose a more challenging problem. As side-channel attacks are known since 1999, countermeasures such as hiding [MMMT09, GM11] and masking [CEM18] have been investigated and deployed in the field. These well known countermeasures can be applied to a multi-tenant FPGA scenario as well as FPGA-specific countermeasures based on reconfiguration [MGV08].

The impact of physical design space on the Side-Channel Analysis (SCA) vulnerability has been investigated in previous works [LB09, RSVC+11, JL09, MSM+14, CEM18, WYR+13]. Especially hiding countermeasures for power equalization struggle with Process Variation (PV) and hardware asymmetry, reducing the effectiveness of such countermeasures [WMG18]. In the new threat model of multi-tenant FPGAs, both the victim design, such as a cryptomodule, and attacker design, such as a TDC sensor, are subject to PV. Moreover, it is yet to be explored, how the sensitivity to voltage fluctuations and the ability to generate them depends on the asymmetric PDN design across the FPGA.

To close this gap, we provide a systematic analysis of the dependencies between physical design parameters and side-channel vulnerability in the context of internal attacks on FPGAs. This is critical for understanding the security vulnerabilities and challenges of multi-tenant FPGA virtualization.

In the following, we briefly summarize reasons why the attack scenario for internal on-chip attacks differs from that of a classical side-channel attacker. On one hand, physical access and external measurement equipment improves the attacker's capabilities:

- The sampling rate and precision of the measurements depend on the quality of the measurement equipment, which is usually only limited by the attacker's financial power.
- The attacker's influence on the victim design is minimal.
- Any side-channels such as power, EM or photon emission can be exploited externally.
- The measurement data corresponds to the actual observables, such as the actual supply voltage, whereas internal measurements can only give estimates.

On the other hand, an internal attacker can exploit the following circumstances:

- Sensors are directly connected to the chip-level PDN and not hindered by board-level decoupling capacitors or noise from other components.

- The internal attacker can acquire localized information, by placing multiple sensors in different regions of the FPGA.

Considering those differences between an attacker with physical access and an FPGA-internal attacker, we believe that a thorough evaluation of the internal attack capabilities is a necessity for the further development of effective countermeasures for this specific scenario, which is the aim of this work. More specifically, we investigate the interaction between logically isolated partitions and analyze the success rate of power analysis attacks in terms of the required amount of measurements for key recovery w.r.t. the following physical design and mapping parameters of attacker and victim modules:

- Global module placement, which includes intra-chip PV and the PDN asymmetry across the chip
- Local primitive placement within partitions
- Inter-chip PV, analyzing different boards of the same type
- Heuristic Place-and-Route algorithms, through recompilation of bitstreams

Identical switching on the logical level causes different voltage noise, depending on the physical mapping, which has been explored for internal measurements on FPGAs as well [GOKT18, ZAB+18]. This dependency is due to the PV [YXL10, ZH12, GNM+13] and the runtime variations in the PDN [GOKT18, ZAB+18] leading to differences in both how switching activity influences the supply voltage and how the impact is observed in specific locations.

Our results on Xilinx Zynq XC7Z020 FPGAs show that the success of the attack is very much dependent on the above parameters, with the amount of traces required varying between a few hundred measurements and entirely unsuccessful attacks with up to $100k$ traces. We confirm similar findings on a larger FPGA PCIe accelerator card based on the Xilinx Virtex-7 XC7VX690T-2 FPGA with up to $10M$ traces, by performing experiments on a subset of parameters. The analysis implies that these physical design parameters are just as critical to the design's side-channel attack resistance as actual countermeasures, as the increase in the amount of traces required is within what some actual simple side-channel countermeasures are able to achieve [GM11, KGS+19, LCL10].

To verify the importance of physical design parameters for side-channel countermeasures, we compare the side-channel vulnerability of a module protected by a hiding scheme based on a power noise generator. In this setup, we demonstrate that not only the vulnerability of an unprotected design but also the effectiveness of countermeasures highly depends on the physical design and mapping parameters.

Finally, we conclude that this work exposes the very complex dependencies between physical design and side-channel vulnerability on a multi-tenant FPGA through a systematic analysis. This work can also lay a foundation for possible effective countermeasures, possibly with zero overhead, based on specific restrictions for local and global placement of trusted and untrusted modules as well as routing constraints.

The structure of the rest of this paper is as follows: In the next section we enlist related work, then in Section 3 we explain the theoretical background behind FPGA-internal power analysis attacks and our evaluation methods. In Section 4 we detail our experiments and the hardware that has been used. Section 5 presents the results of evaluating power analysis attacks w.r.t the mentioned parameters, which are later discussed in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

In this section, we elaborate on a selection of related works, which significantly contributed to our choice of experiments and evaluation methods. First, we present recent works about FPGA-internal side-channel attacks on various platforms, which are based on different sensor implementations. Recently proposed countermeasures against such attacks are shown in the second subsection.

2.1 FPGA-internal Side-Channel Attacks

In 2013, Zick et. al. showed how a Time-to-Digital Converter (TDC) sensor [Wu10] can be used to sense nanosecond-scale voltage transients inside FPGAs [ZSZF13]. A TDC is based on a long delay chain, which is often realized using carry primitives in FPGAs, to estimate voltage fluctuations by measuring the voltage-dependent transistor delay. In [GOKT18], the effect of logic activity on TDC sensors was explored and also analyzed for spatial dependencies.

These initial works were followed by multiple demonstrations of attacks on logically isolated designs inside multi-tenant FPGAs [SGMT18b, ZS18] and even across chips on the same board [SGMT18a]. The threat model is usually constructed around the concept of a *multi-tenant* FPGA, which is a single FPGA chip used by multiple untrusted users at the same time. With the increasing size of modern FPGAs chips, the virtualization of FPGA fabric is especially attractive for cloud computing scenarios to maximize resource utilization [EV12, BSB$^+$14, FVS15]. The basic assumption is for attacker and victim designs to be placed on the same FPGA into different, spatially separated partitions without any direct connections between them. A victim design would be a security-critical module such as an encryption core or a random number generator, whereas the attacker attempts to measure or influence the supply voltage to extract information or induce faults.

All of the attacker designs exploit the dependency of the circuit delay on the supply voltage, which in turn is affected by the data-dependent transistor toggling of the victim logic. The circuit delay is measured either with the described TDC or a Ring Oscillator (RO) based counter. Successful attacks were demonstrated on RSA (unprotected square-and-multiply) as well as on AES implementations. In another line of FPGA-internal side-channel attacks, the crosstalk of co-residing interconnect wires is exploited [RPD$^+$18], which can to the best of our knowledge not be exploited over longer distances and is thus not further elaborated here.

In [SGMT18b], attacks on an unprotected AES module were successful after a few hundred measurement traces on the SAKURA-G board and Spartan-6 FPGA, dedicated for side-channel analysis. These results also correspond to our best-case attack success on the Xilinx Zynq XC7Z020 used in this work. Since the RO counter offers only limited sampling rates, we focus on TDC sensors in our work. We provide further details on our specific sensor implementation for this work in Section 3.

Likewise, fault attacks [BDL97] have also been demonstrated in multi-tenant FPGAs [KGT18], which are able to deduce secret information from injecting faults into specific computation steps of a cryptographic algorithm and observing the faulty output.

2.2 Existing Countermeasures against Power Analysis Attacks

In light of the mentioned attacks in the previous subsection, researchers have proposed defenses for the specific scenario of a virtualized FPGA. Despite being mostly developed with external attackers and physical access in mind, we also want to briefly mention the major categories of classical SCA countermeasures. Considering the secret data-dependent information within the measurement trace as the signal which the attacker wants to acquire, most countermeasures aim to decrease the Signal to Noise Ratio (SNR). This can be done by generating noise [KBG09] or equalizing the data-dependent power consumption of computations [MMMT09], thus *hiding* the sensitive information, or performing computations with randomized data on the algorithmic level, which classifies as *masking* [CEM18]. Hiding schemes often make use of Dual-Rail Precharge (DRP) logic to achieve power equalization [WMG18] through duplication of the original circuit. In the context of DRP countermeasures on FPGAs, the high impact of placement and routing variations is already well established albeit only with external attacks and the use of measurement equipment in mind. The vulnerability of DRP schemes against remote internal attacks is yet to be explored.

Specifically on FPGAs, researchers proposed to make use of Partial Reconfiguration (PR) to mitigate side-channel attacks through temporal jitter [MGV08] or, for example, interleaving implementations of an AES S-Box [BSP+19]. In the context of multi-tenant FPGAs in the cloud, bitstream checking has been proposed as a method to detect and block fault and side-channel attacks already on a hypervisor level, before the bitstream is being downloaded to the FPGA [KGT19, GRKT18]. Through analyzing reverse-engineered bitstreams as netlists with placement annotations, this method can effectively serve as something like an FPGA anti-virus, but the success depends highly on the provided signatures and requires the user to give up on bitstream confidentiality towards a possibly untrusted hypervisor.

There is also a recent countermeasure against on-chip SCA through power equalization with an RO-based active fence between attacker and victim [KGS+19]. The fence is implemented as a row-by-row RO array, with the row activation depending on either a sensor value for power equalization or a Pseudo-Random Number Generator (PRNG) for noise increase. This countermeasure is in line with previous works on randomly activated ROs for noise generation [LCL10] and will serve as an example for a generic hiding countermeasure to evaluate the effect of physical design parameters on protected designs in our work. We deploy RO arrays around the AES modules which are randomly activated. The noise increase makes the attack more difficult and increases the minimum amount of traces required. A more detailed description of this experiment is provided in Subsection 3.4.

Although we do not propose a new countermeasure in this work, we highlight the sensitivity of SCA attacks to various placement parameters for both trusted (victim) and untrusted (attacker) modules, which will be important for further development against side-channel leakage in the multi-tenant FPGA threat model. Our results show that noise generation as a hiding countermeasure can have as much impact as the exploration of the physical design space.

3 Theoretical Background and Methodology

For readers to understand our assessment methodology, we provide some necessary theoretical background information in this section. Details on FPGA-internal voltage sensing and noise generation are followed by a brief explanation of the Correlation Power Analysis (CPA) attack on the AES, which we used to evaluate attack success in terms of traces required for key recovery. In general, we perform two kinds of experiments: First we evaluate the general impact of a noise generation module based on ROs or toggling Flip Flops (FFs) on TDC sensors, then we assess the actual CPA attack success on AES modules. This approach allows us to analyze the general dependencies of module placement in the FPGA and SCA success before investigating the complexity of the actual attack w.r.t. all physical design parameters. In Section 5, we see how a simple correlation between the impact of noise generation modules on sensors and SCA success can not be established, but we still can infer a relation between the results from both types of experiments. Moreover, we show how a simple noise-generation countermeasure performs under varying conditions to

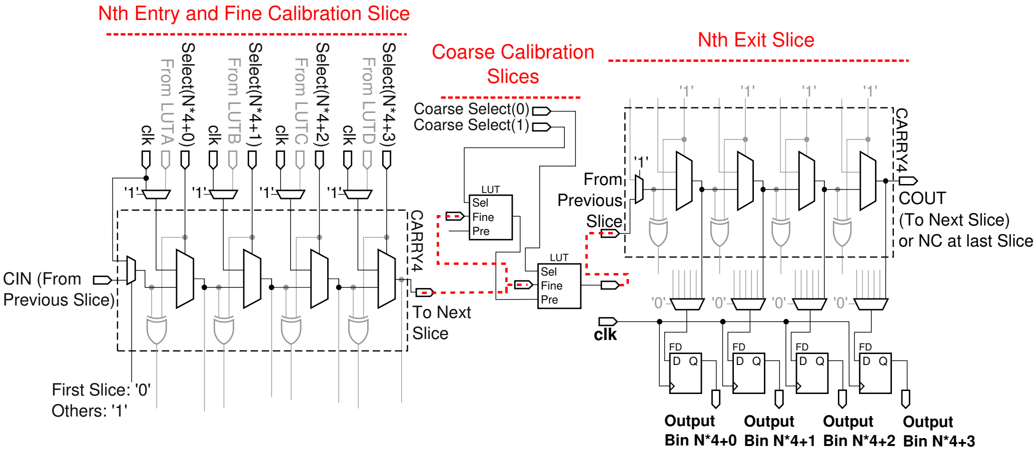


Figure 1: Principle of a delay line based on Xilinx LUT, FD, and CARRY4 elements [Xil16]. In *Fine Calibration*, selectable entry points of the clock *clk* are shown to allow for small adjustments of the total path length. The center slices use the output from the Fine Calibration as the input to *Coarse Calibration* slices based on LUTs. In *Exit* slices, multiple bins of the delay line are routed to Flip Flop (FF) (FD) primitives as the output values.

verify the importance of a thorough analysis for mitigating SCA attacks. This exemplary countermeasure is described in detail in the last subsection of this section.

3.1 Self-calibrating FPGA-internal Voltage Sensors

To perform estimated measurements of chip-internal supply voltage, we use sensors based on TDCs [Wu10], and introduced by Zick et al. [ZSZF13] for voltage measurement in FPGAs. These sensors are sensitive to process, voltage, and temperature (PVT) variations [GOKT18, ZSZF13], with voltage having the most influence during runtime. Thus, previous attacks manually calibrated for process variations offline, and CPA was still successful, despite small temperature changes [SGMT18b].

These sensors work by monitoring how fast a signal propagates through a long path by putting FFs at various depths alongside the path. When we connect the clock to the entry of the path and to the FFs, they will show every clock cycle how far it can propagate, which depends on the voltage level.

An issue with these kind of sensors motivates an adjusted design. It is usually not possible to know how long the path is supposed to be at design time, since both intra-die and inter-die manufacturing process variation can be significant. In previous works, the length of the total sensor needed to be adjusted in order to account for process variations, or operating frequency [SGMT18b]. We will show a design here which allows calibration at runtime, such that the same bitstream works on multiple boards, and can also be recalibrated to account for temperature changes. Here we specifically show how they are implemented in Xilinx slices, which is similar from at least their 5th Generation to Ultrascale+ line. The design is made out of three types of FPGA slices, that we show together in Figure 1.

In *Fine Calibration* slices, we allow the clock 'clk' to enter a Xilinx CARRY4 primitive (c.f. [Xil16]) at various depths of the delay line, by connecting it to all possible inputs that can be selected with multiplexers. Since the clock tree is balanced before the clock is connected to the CARRY4, the selection scheme allows for varying the path length in small steps. CARRY4 elements are used since they are known to allow for the smallest delays, which are about $12ps$ for one bit in a Xilinx 6-series FPGA [GOKT18]. Similar primitives are available in FPGAs of other vendors.

Coarse Calibration slices are fed with the output of *Fine Calibration* slices. In these slices, Look-Up Tables (LUTs) are configured in a similar cascaded way as the CARRY4 elements in Fine Calibration, but instead of the clock, they all get the output from fine calibration. Please note that this can lead to some non-linearities since the signal does not anymore benefit from the balanced clock tree, but needs to use standard FPGA interconnect resources.

Exit slices follow the design of related work [ZSZF13], in which CARRY4 primitives are used again for their small delays. However, here we add FFs between the multiplexers, which can then signal how far the clock could propagate through the path.

In the end, sufficient fine and coarse calibration slices need to be implemented, such that the first FF in the exit slices does fail timing requirements significantly, since typically high timing safety margins are applied in the timing analysis. The output of the sensor can sometimes show a zero in between a row of 1's, since the process variations between the FFs can exceed the delay of one bit. Thus so-called bubble detection has to be applied, which essentially switches the position of the highest '1' with the '0' that follows after it [Wu10]. During runtime, the sensor can be automatically calibrated by a state machine that checks the sensor output and adjusts coarse and fine calibration stages accordingly. The target is usually that the sensor is in the middle of the possible output range (e.g. 0–63), to be able to show both negative and positive voltage differences.

Please note that in [ZSZF13] it was suggested to use a phase-shifted clock at either entry or exit, to adjust the effective path length instead of the presented approach with calibration slices. We have also experimented with that, but got significant noise at idle. We think that noise comes from having two separated clock trees for the two clocks, and these clock trees getting affected with a different jitter.

3.2 Investigating the Impact of Global Module Placement

Initially, we want to evaluate the impact of switching voltage noise caused by modules placed in different regions of the FPGA, which is modulated on the PDN and observed by sensors in different locations. Thus, we employ variants of generic voltage noise generators using ROs or toggling FFs. ROs are implemented using a single inverter LUT with a feedback loop. To evaluate the effect of toggling FFs, we simply insert a register into the feedback loop, which is clocked by an arbitrary Phase Locked Loop (PLL) on the FPGA. The ROs or FFs are deployed as synchronously enabled grids on the FPGA roughly the size a regular AES module would occupy. A similar analysis has been done in [GOKT18], which already showed some spatial dependencies of sensor placement and switching activity, but without side-channel attacks in mind.

We outline the generic principle of our noise generation modules in Figure 2. These noise generators are intended to model the switching activity caused by the AES module, while eliminating the influence of local primitive placement. The amount of the switching activity as well as the local placement of all FPGA primitives within the noise generating modules is exactly the same for all modules. In addition, the local primitive placement of the self-calibrating TDC sensors is always identical. Therefore, any differences observed in the generated or sensed voltage noise can only result from inter- and intra-chip PV as well as variations in the PDN structure.

The impact of these noise generation modules and the sensitivity of sensors can now be analyzed by comparing sensor values with and without activated noise module. We compare both average and variance of sensor values during the respective period, which corresponds to 512 sensor samples each. In Figure 3 we show the impact of the FF based noise generation grid on 30 different sensors (grey) and the average of all 30 sensors (red) on our experimental platform. A significant drop in the average sensor value from $\mu_{[0,512)}$ to $\mu_{[512,1024)}$ is clearly visible after the FF toggling is activated at 512 samples. Albeit not

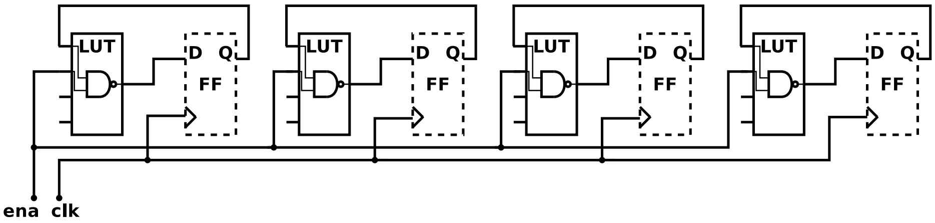


Figure 2: Generic schematic of an artificial noise generation grid based on ROs or toggling Flip Flops (FFs)

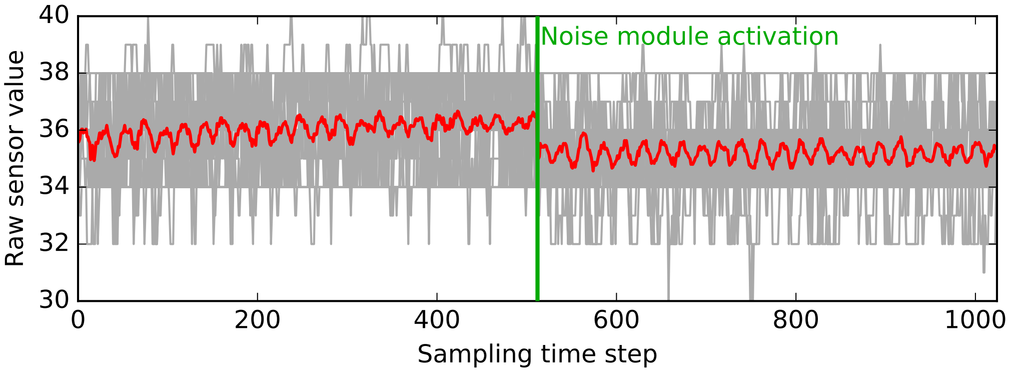


Figure 3: Raw traces (grey) and average (red) over 30 different sensors for 1, 024 sample points with noise module activation after 512 samples

clearly visible in the figure, a variance increase after toggle activation can be measured as well. We define a measure of impact or sensitivity (depending on the perspective of either noise generators or sensors) as the absolute of the subtraction $\delta^{\mu} = |\mu_{[0,512)} - \mu_{[512,1024)}|$ for the average or $\delta^{\sigma} = |\sigma_{[0,512)} - \sigma_{[512,1024)}|$ for variance respectively.

3.3 Correlation Power Analysis Attacks on AES

We evaluate the success of a classical CPA attack [BCO04] with a slight modification to the power model. CPA is based on correlating the actual measurements with a power model, which is based on a key-byte guess. Our power model for an attack on the last encryption round of AES is defined as $pm = \text{sbox}^{-1}(k_{\text{guess}} \oplus c) \wedge 2^b$, where k_{guess} is the key byte guess and c the corresponding ciphertext byte. With $b = \{0, 1, ..., 7\}$ we compute the correlation values per bit, since some bits correlate better than others. After evaluating a certain amount of traces, the correct key guess shows a higher correlation in one of the eight bits than the incorrect guesses, thus allowing key recovery.

To assess the attack success, we determine the minimum amount of traces required for recovering the first key byte of the last AES round key. If an attacker is able to recover only a single key byte or even 50% of the secret AES round key, the remaining key space is of course still too big for an exhaustive search. However, our goal is to analyze the general impact of design space parameters on side-channel vulnerability without considering a practical attack in a real-world scenario. For comparing the amounts, we define an exact measure for a successful key recovery: First, we determine the sampling point $t_{\max}$ with the highest overall correlation, which most likely corresponds to the point in time when the last encryption round is computed. We consider an attack successful, if the correlation for any of the 8 bits in $t_{\max}$ for the correct key guess is larger than 1.5 times the second

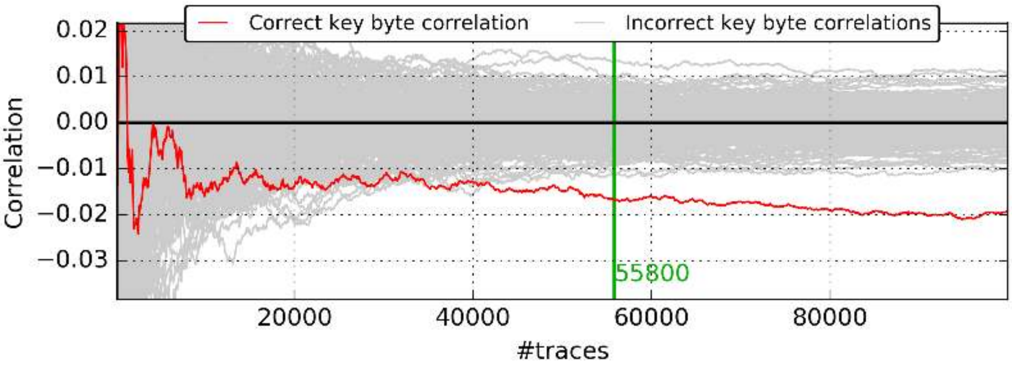


Figure 4: Example of a successful key recovery after 55, 800 traces, where the correlation with the correct key guess is less than 1.5 times the second lowest correlation value.

highest correlation value or smaller than 1.5 times the second smallest correlation value. As an example, the successful recovery according to our definition of a key byte after 55, 800 measurements is shown in Figure 4.

3.4 A Simple Noise-Generation Countermeasure

To evaluate the impact of physical mapping on SCA mitigation, we employ a hiding countermeasure based on previous works [LCL10, KGS+19], which artificially increases the circuit noise using ROs. An array of ROs is mapped layer-by-layer around the AES modules and the amount of activated layers is determined randomly during encryption. In Figure 5 we show the floorplan of the countermeasure around a single AES partition as seen in the Xilinx Vivado design software. Activating a higher amount of ROs causes the supply voltage to drop, whereas deactivation of RO layers raises the supply voltage. Thus, the RO array creates randomized voltage fluctuations over the fluctuations caused by the AES module. Consequently, the attacker needs a higher amount of traces to recover the secret key due to the worse SNR. In Section 5, we show how physical design mapping parameters impact this countermeasure to motivate their importance not only for future but also for existing attempts at mitigating SCA.

4 Experimental Setup

We performed two different kinds of experiments on our evaluation platform. In the first subsection, we provide details about the platform itself, whereas the experiments are explained in the following two subsections.

4.1 Platform

For our experiments, we use the Pynq-Z1 board from Digilent, which is based on a Xilinx Zynq XC7Z020-1CLG400C. This System on Chip (SoC) consists of Artix-7 based Programmable Logic (PL) together with a dual-core ARM Cortex-A9 processor. The board is capable of running a Linux system from an SD card on the ARM core, which allows easy interaction with the PL and fast sampling of sensor values through an AXI interface directly onto the SD card. We run our experiments on four different Pynq-Z1 boards to estimate the impact of inter-chip PV. During sampling, the boards are encapsulated in a metal casing inside an ordinary household fridge (c.f. Figure 6), to minimize environmental impact of temperature and electromagnetic radiation. Moreover,

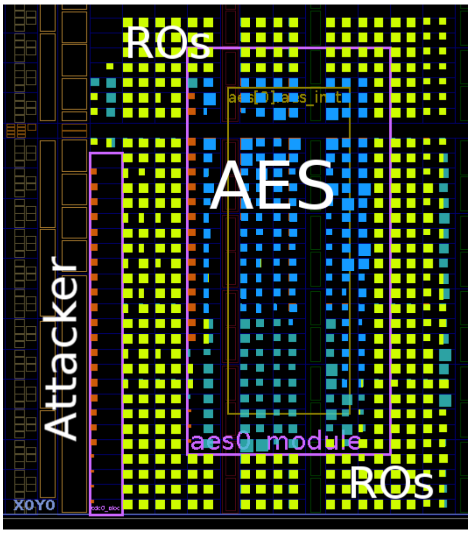


Figure 5: Floorplan of our rudimentary example countermeasure based on a randomly activated array of ROs (yellow) around the AES module (blue).

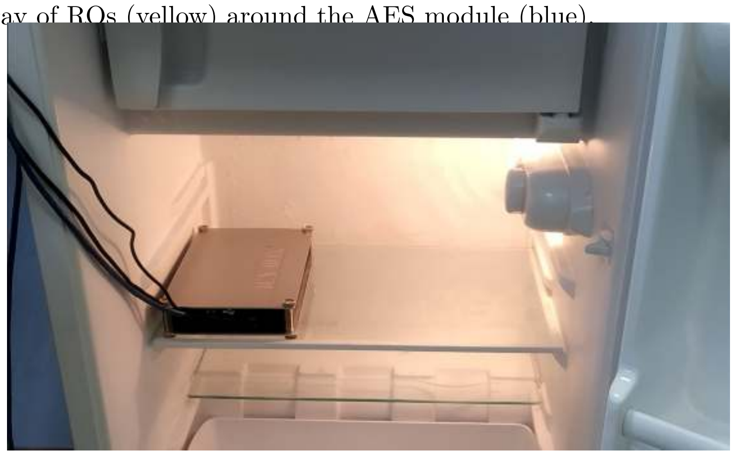
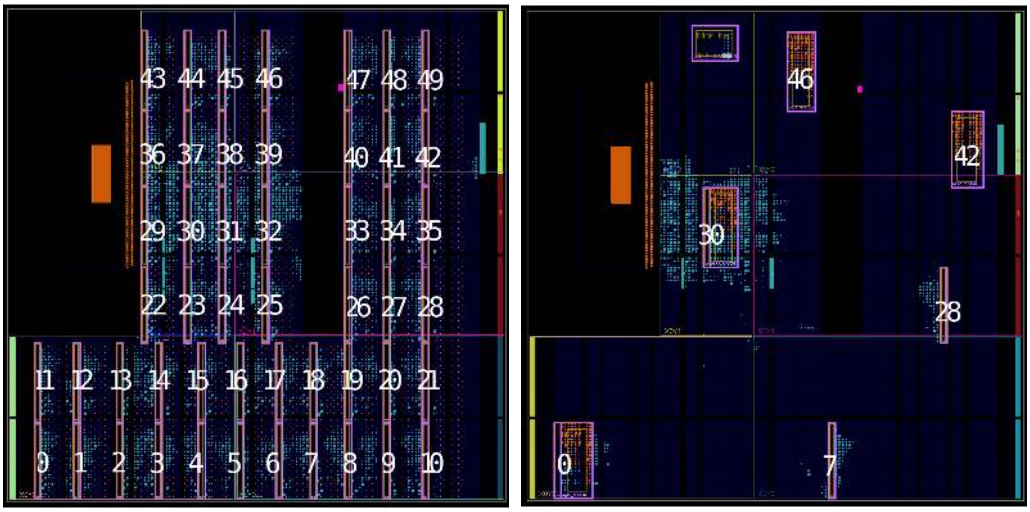


Figure 6: Picture of one of the four boards encapsulated in a metal case inside an ordinary household refrigerator. Three cables coming out of the case are USB, Ethernet, and Power.

we repeat any experiment twice and the first result is discarded using only the second result for evaluation to minimize the differences in actual chip temperature. We also perform experiments on the ADM-PCIE-7V3, a Virtex-7-based PCIe accelerator card, where sensor values are sampled through the PCIe interface onto the host computer directly. As our objective is to confirm the generality of trends, we only evaluate a subset of the previously elaborated design space parameters on this larger platform.

4.2 Evaluating the Effect of Global Placement

Initially, we attempt to investigate the general influence of the locations of both cryptomodule (victim) and sensor (attacker), by performing measurements as explained in Subsection 3.2 for 50 different locations. A floorplan of our evaluation design for the Pynq-Z1 board as seen in the Xilinx Vivado design software can be examined in Figure 7a. In each of the 50 designated locations, we place a self-calibrating TDC sensor next to an FF-based noise generation grid next to each other. The size of the grid is chosen to correspond to the size of the AES module implementation we use in later experiments. Several



(a) Floorplan of our placement impact evaluation design on the Pynq-Z1.

(b) Floorplan of our attack success evaluation design on the Pynq-Z1.

Figure 7: Floorplans of our evaluation designs as seen in the Xilinx Vivado Design software for analyzing the impact of global placement on the entire FPGA and assessing the attack success on four by four different sub-locations.

multiplexers, which are controlled by registers from the ARM core, allow the activation of a specific sensor/FF grid combination to evaluate all $2,500$ possible combinations. For a specific combination, we measure either $\delta^{\mu}_{s,n}$ or $\delta^{\sigma}_{s,n}$, where δ^{μ} and δ^{σ} are defined as in Subsection 3.2 and s and n correspond to the sensor and noise generator locations as presented in Figure 7a respectively. We examine which sensors show the most sensitivity to all noise generators as well as which grid modules generate the largest impact on all sensors.

Thus, all $2,500$ possible combinations are activated one by one and for each noise generator location $i = \{0, 1, ..., 49\}$ we compute the total impact caused by noise generators on either sensor average or variance, $I^{\mu}(i)$ or $I^{\sigma}(i)$ as the sum of all impact values caused by the FF grid in that location. Likewise, for each sensor location $i = \{0, 1, ..., 49\}$ we compute the total sensitivity based on sensor average or sensor variance, $S^{\mu}(i)$ or $S^{\sigma}(i)$, as the sum of the sensor's sensitivity in that location to all FF grids. More specifically, the total impact and total sensitivity values are defined as follows:

- Total sensitivity based on sensor average of a sensor at location i:
$$S^{\mu}(i) = \sum_{k=0}^{49} \delta^{\mu}_{i,k}$$
- Total impact based on sensor average caused by a noise generator at location i:
$$I^{\mu}(i) = \sum_{k=0}^{49} \delta^{\mu}_{k,i}$$
- Total sensitivity based on sensor variance of a sensor at location i:
$$S^{\sigma}(i) = \sum_{k=0}^{49} \delta^{\sigma}_{i,k}$$
- Total impact based on sensor variance caused by a noise generator at location i:
$$I^{\sigma}(i) = \sum_{k=0}^{49} \delta^{\sigma}_{k,i}$$

These four total impact and total sensitivity values $I^{\mu/\sigma}(i)$ and $S^{\mu/\sigma}(i)$ are measured 1000 times and averaged to finally acquire a map of locations on the FPGA fabric with their respective capability to sense or generate voltage noise.

4.3 Evaluating Attack Success

To assess the attack success on the Pynq-Z1 we evaluate CPA with up to $100k$ measurement traces as explained in Subsection 3.3. For evaluation of the parameter impact on a protected design, we collect up to $500k$ traces with the enabled hiding countermeasure described in Subsection 3.4. On the ADM-PCIE-7V3 the attack is more difficult in general, which is why we collect up to $10M$ traces for each parameter selection on that platform.

The AES core implements a block cipher encryption with a 128 bit key length and takes around 580 LUTs ($\approx 1\%$ of all LUTs on the Pynq-Z1) on our Xilinx 7-Series FPGAs. The module uses a 32 bit data-path, so 4 bytes are computed simultaneously and one round takes 5 clock cycles to complete. The four S-Boxes are implemented in logic and not as, for instance, a lookup table in BRAM. We always operate the AES encryption at a frequency of 25 MHz, whereas the attacker sensor samples at 100 MHz.

Since the amount of data we would require to evaluate all possible 50×50 locations from the initial impact experiments would exceed multiple terrabytes for only a single local placement or board, we restrict the attack evaluation to four different sensor and four different AES locations. This restriction results in four by four CPA results. The choice of locations is rather arbitrary but with the results of the previous experiments on global placement in mind, choosing sensor and AES locations which are more sensitive or have a larger impact on the voltage fluctuations. Figure 7b shows the selected locations on the FPGA floorplan for the Pynq-Z1.

We extend the experiment to four different boards and four different local placement strategies, to explore the effects of PV and local placement of the primitives within the AES module. The different placement strategies to explore the impact of local placement of FPGA primitives within each AES module are selected from the available options in the Xilinx Vivado design software:

- Default strategy
- Optimize for performance
- Optimize for area
- Optimize for power consumption

After compiling the design with a selected placement strategy, the placement constraints for one AES module are saved and replicated to the other four AES locations. Then, we recompile the design for evaluation with default settings but applying the placement constraints of one of the four strategies on the AES modules.

In total, this gives us 256 experiments, which we can then analyze w.r.t. the influence of each parameter on the success of the attack. An overview of the entire design space considered in this work is given in Table 1, with the global location identifiers corresponding to the assignments in Figure 7a. Additionally, we evaluate the effect of routing variations by recompiling the design with identical placement constraints on the AES modules. On the ADM-PCIE-7V3, we evaluate only a single board and the default local placement strategy, varying the global sensor and AES placement parameters, which leads to 16 experiments on that platform.

If not explicitly defined otherwise, we always attack the first round key byte of the last AES encryption round with the default key used in the examples in the appendix of the NIST AES specification [NIS01], where the correct hexadecimal value of the key byte is 0xD0. As an example, we also show results for the second round key byte on only a single

Table 1: Overview of our design space exploration, leading to 256 experiments on the success of CPA on AES.

Design parameter	Available parameter choices			
Inter-chip PV	Board A	Board B	Board C	Board D
Sensor location	0	7	30	28
AES location	0	30	42	46
Local primitive placement in AES modules	Default settings	Performance optimized	Area optimized	Power optimized

board in the appendix. The random plaintexts used in the attack are identical and issued in the same order for each experiment.

5 Results

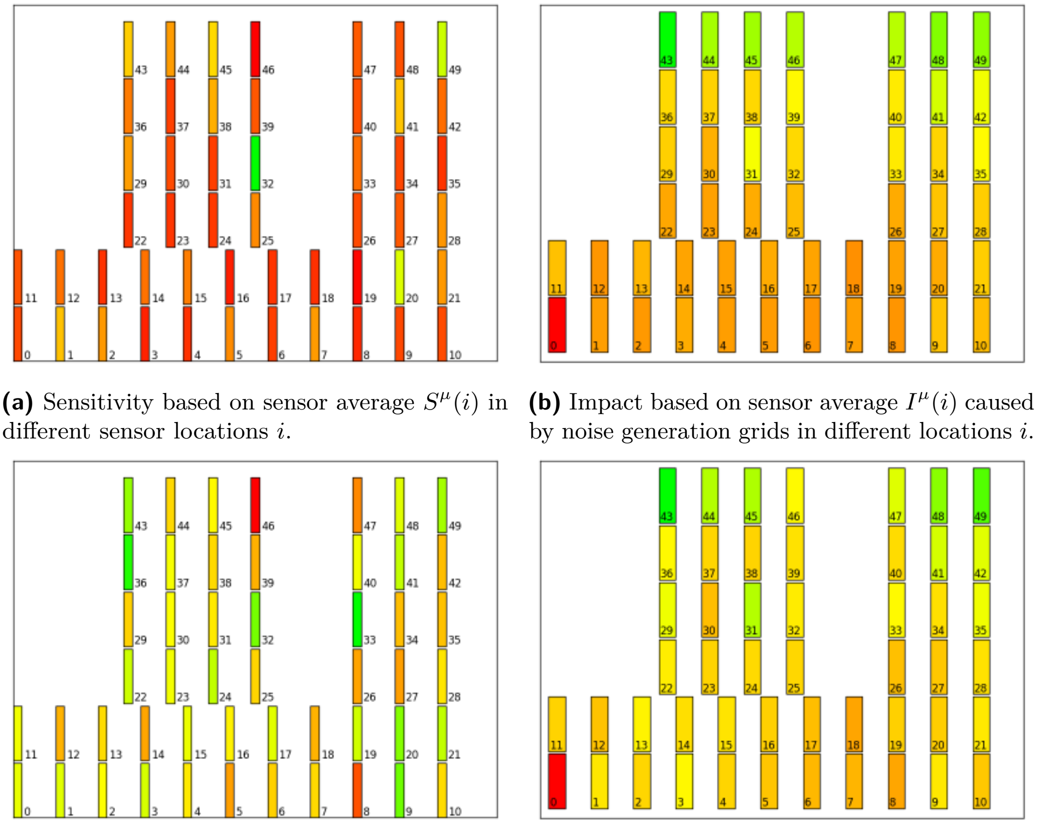
In this section, we present the results of our experiments, which are later discussed in Section 6. We start with presenting the measurements from evaluating the impact of noise modules on sensors in various respective locations, followed by the results of CPA on AES in different setups.

5.1 Effect of Global Placement

Here we show the results of performing measurements as described in Subsection 3.2 and Subsection 4.2 to evaluate the influence of voltage noise generators on voltage sensors in different respective locations on the chip. Through these measurements we intend to identify locations that generate a high amount of voltage fluctuations (as a source of information leakage) as well as locations that are more sensitive to them (as a sink, for a potential attacker sensor). The initial motivation is to be able to correlate the results with the CPA attack success evaluation, as presented in the next subsection. Although we are unable to actually present a direct correlation, we can still identify some connections between the results of the two experiments, which is why we present both of them in this work.

We measure all four variants of the total impact and total sensitivity $I^{\mu/\sigma}(i)$ and $S^{\mu/\sigma}(i)$ as defined in Subsection 4.2 of all 50 locations on two different boards. In Figure 8a, we see how all noise generation grids cause a specific distribution of sensitivity of the sensor average $S^{\mu}(i)$ for each sensor location i on the FPGA fabric. On the other hand, we show in Figure 8b the impact $I^{\mu}(i)$ on all sensor average values caused by each toggling FF grid location i. In all figures, the color scale, which is logarithmic, reflects the sensitivity of the sensor location or the impact of the noise generator location respectively: A higher sensitivity or higher impact is red, whereas lower sensitivity corresponds to a green color. The actual impact/sensitivity values have no specific meaning and are therefore omitted in the diagrams. The locations in the figures correspond to the respective locations as presented in Figure 7a in Subsection 4.2.

When considering the impact and sensitivity values $I^{\sigma}(i)$ and $S^{\sigma}(i)$ based on the variance of the sensor measurement presented in Figure 8c and Figure 8d, we observe almost no difference in the total impact $I^{\sigma}(i)$ caused by each noise module at location i,



(a) Sensitivity based on sensor average $S^{\mu}(i)$ in different sensor locations i.

(b) Impact based on sensor average $I^{\mu}(i)$ caused by noise generation grids in different locations i.

(c) Sensitivity based on sensor variance $S^{\sigma}(i)$ in different sensor locations i.

(d) Impact based on sensor variance $I^{\sigma}(i)$ caused by noise generation grids in different locations i.

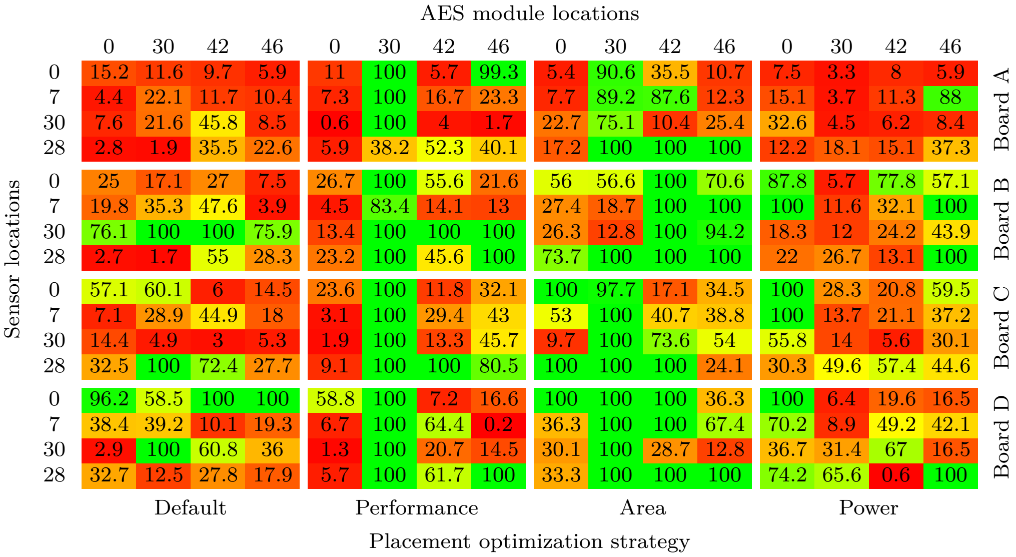
Figure 8: Influence of global placement of FF-based noise generators and TDC sensors evaluated and averaged over 1000 measurements on board A. Locations that are colored orange/red correspond to sensors that are more sensitive or noise generators that cause a higher impact respectively. Green colored locations are less sensitive or cause less impact.

but quite a few differences in the total sensitivity of each sensor to all noise generators $S^{\sigma}(i)$.

We also evaluate the impact and sensor sensitivity values based on sensor average and sensor variance on a second board of the same type. The results, which can be found in the appendix, are consistent with the ones obtained on the first board: Again, the spatial distribution of the total impact caused by each noise module is very similar to the results on the first board, whereas the distribution of the sensor sensitivity differs significantly.

From these initial experiments, we can identify sensor locations, that seem to be more sensitive to the voltage fluctuations in general, such as location 46. However, the location sensitivity is not fixed across boards, since locations 20 and 49 are rather insensitive on board A, but much more sensitive on board B. Moreover, sensitivity depends on the measurement method, as the impact on sensor average has a different distribution than the impact on sensor variance. On the other hand, we observe very similar distributions of the impact caused by noise generators in specific locations, where location 0 is the strongest across boards and even with different measurement approaches.

In conclusion, the sensitivity of specific sensors seems to be predominantly defined by inter- and intra-chip PV and the asymmetry of the PDN across the chip, whereas the



Board	Sensor	Default 0	Default 30	Default 42	Default 46	Performance 0	Performance 30	Performance 42	Performance 46	Area 0	Area 30	Area 42	Area 46	Power 0	Power 30	Power 42	Power 46
Board A	0	15.2	11.6	9.7	5.9	11	100	5.7	99.3	5.4	90.6	35.5	10.7	7.5	3.3	8	5.9
	7	4.4	22.1	11.7	10.4	7.3	100	16.7	23.3	7.7	89.2	87.6	12.3	15.1	3.7	11.3	88
	30	7.6	21.6	45.8	8.5	0.6	100	4	1.7	22.7	75.1	10.4	25.4	32.6	4.5	6.2	8.4
	28	2.8	1.9	35.5	22.6	5.9	38.2	52.3	40.1	17.2	100	100	100	12.2	18.1	15.1	37.3
Board B	0	25	17.1	27	7.5	26.7	100	55.6	21.6	56	56.6	100	70.6	87.8	5.7	77.8	57.1
	7	19.8	35.3	47.6	3.9	4.5	83.4	14.1	13	27.4	18.7	100	100	100	11.6	32.1	100
	30	76.1	100	100	75.9	13.4	100	100	100	26.3	12.8	100	94.2	18.3	12	24.2	43.9
	28	2.7	1.7	55	28.3	23.2	100	45.6	100	73.7	100	100	100	22	26.7	13.1	100
Board C	0	57.1	60.1	6	14.5	23.6	100	11.8	32.1	100	97.7	17.1	34.5	100	28.3	20.8	59.5
	7	7.1	28.9	44.9	18	3.1	100	29.4	43	53	100	40.7	38.8	100	13.7	21.1	37.2
	30	14.4	4.9	3	5.3	1.9	100	13.3	45.7	9.7	100	73.6	54	55.8	14	5.6	30.1
	28	32.5	100	72.4	27.7	9.1	100	100	80.5	100	100	100	24.1	30.3	49.6	57.4	44.6
Board D	0	96.2	58.5	100	100	58.8	100	7.2	16.6	100	100	100	36.3	100	6.4	19.6	16.5
	7	38.4	39.2	10.1	19.3	6.7	100	64.4	0.2	36.3	100	100	67.4	70.2	8.9	49.2	42.1
	30	2.9	100	60.8	36	1.3	100	20.7	14.5	30.1	100	28.7	12.8	36.7	31.4	67	16.5
	28	32.7	12.5	27.8	17.9	5.7	100	61.7	100	33.3	100	100	100	74.2	65.6	0.6	100

Figure 9: Minimum amount of traces ($\times 1,000$) required with different placement strategies for all possible combinations of AES and sensor location on four different Pynq-Z1 boards. Red colored cells correspond to combinations where an attack is easy, green colored to parameters resulting in a difficult attack. A value of $100k$ means that we were unable to recover the key with $100k$ traces.

impact of switching activity on the supply voltage depends mostly on the noise generator location.

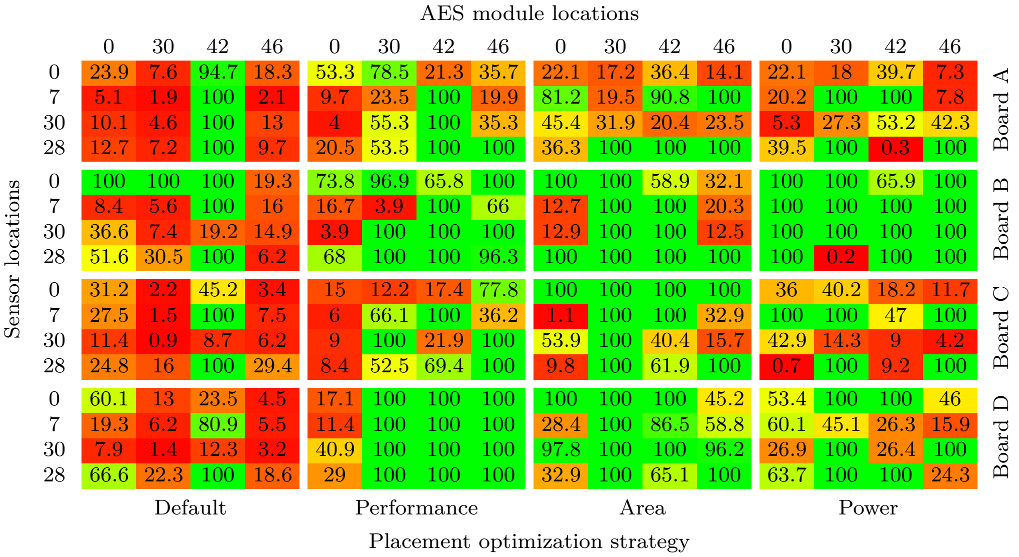
We present in the next subsection that CPA attack success seems to be correlating with neither sensor sensitivity distribution nor noise generator impact distribution. This is due to the high impact of local primitive placement when replacing the noise generators with the actual AES encryption modules. However, we are able to draw some conclusions from the attack success w.r.t the global AES module placement.

5.2 Attack Success Evaluation on an Unprotected Design

The main part of our experiments is the assessment of CPA success on AES in all 256 scenarios, as explained in Subsection 4.3. As mentioned in Subsection 4.3, we choose a subset of four locations, which are shown in Figure 7b for sensors and AES modules out of all 50 locations from the previous experiment on voltage noise impact. The choice of locations is partially based on the results of the previous experiments. We include locations of interest such as location 0, which caused the highest impact on all sensors in the experiments in Subsection 5.1, and select the rest arbitrarily, but further apart to reasonably evaluate the global mapping impact.

In Figure 9 we present all results of performing CPA on four different boards, four different local placement strategies and the four by four possible combinations of sensor and AES location. The table cells are colored according to the SCA success assessment: Less amount of traces required corresponds to a more vulnerable setup and is thus colored in a red tone, whereas a higher amount of traces is marked green, to indicate less vulnerability. We note again that only attacks with up to $100k$ traces have been evaluated, therefore all values of $100k$ indicate an unsuccessful attack with $100k$ measurements. First, we notice that despite attacking the same byte of the last AES round key, despite the identical random plaintexts and despite our extensive efforts to eliminate fluctuations in environmental parameters, the difference in the required amount of traces is remarkably up to 500×.

We note that the routing randomization impact is by no means a negligible effect



		Default				Performance				Area				Power			
		0	30	42	46	0	30	42	46	0	30	42	46	0	30	42	46
Board A	0	23.9	7.6	94.7	18.3	53.3	78.5	21.3	35.7	22.1	17.2	36.4	14.1	22.1	18	39.7	7.3
	7	5.1	1.9	100	2.1	9.7	23.5	100	19.9	81.2	19.5	90.8	100	20.2	100	100	7.8
	30	10.1	4.6	100	13	4	55.3	100	35.3	45.4	31.9	20.4	23.5	5.3	27.3	53.2	42.3
	28	12.7	7.2	100	9.7	20.5	53.5	100	100	36.3	100	100	100	39.5	100	0.3	100
Board B	0	100	100	100	19.3	73.8	96.9	65.8	100	100	100	58.9	32.1	100	100	65.9	100
	7	8.4	5.6	100	16	16.7	3.9	100	66	12.7	100	100	20.3	100	100	100	100
	30	36.6	7.4	19.2	14.9	3.9	100	100	100	12.9	100	100	12.5	100	100	100	100
	28	51.6	30.5	100	6.2	68	100	100	96.3	100	100	100	100	100	0.2	100	100
Board C	0	31.2	2.2	45.2	3.4	15	12.2	17.4	77.8	100	100	100	100	36	40.2	18.2	11.7
	7	27.5	1.5	100	7.5	6	66.1	100	36.2	1.1	100	100	32.9	100	100	47	100
	30	11.4	0.9	8.7	6.2	9	100	21.9	100	53.9	100	40.4	15.7	42.9	14.3	9	4.2
	28	24.8	16	100	29.4	8.4	52.5	69.4	100	9.8	100	61.9	100	0.7	100	9.2	100
Board D	0	60.1	13	23.5	4.5	17.1	100	100	100	100	100	100	45.2	53.4	100	100	46
	7	19.3	6.2	80.9	5.5	11.4	100	100	100	28.4	100	86.5	58.8	60.1	45.1	26.3	15.9
	30	7.9	1.4	12.3	3.2	40.9	100	100	100	97.8	100	100	96.2	26.9	100	26.4	100
	28	66.6	22.3	100	18.6	29	100	100	100	32.9	100	65.1	100	63.7	100	100	24.3

Figure 10: Minimum amount of traces ($\times 1,000$) required on recompiled designs to analyze the effect of routing randomization. Red colored cells correspond to combinations where an attack is easy, green colored to parameters resulting in a difficult attack. A value of $100k$ means that we were unable to recover the key with $100k$ traces.

from a security perspective. Since we are only able to fix the primitive placement in our implementations, a recompilation causes the Vivado Design software to re-route certain nets. Figure 10 presents the results of evaluating CPA on recompiled bitstreams under the same parameter variations as before. Although the previously discovered dependencies are still valid, the overall results differ quite significantly. These results show the significant impact of local placement and routing on the security of the module in terms of SCA vulnerability.

Next, we try to extract dependencies between the four given parameters (board, global placement of AES, global placement of the sensor, local placement strategy) and the attack success. Therefore, we also average the results across different dimensions and also across the results from original and recompiled bitstreams, which is presented in Table 2.

In the following, we summarize and discuss the major observations from the experiments:

Table 2: Averages over the minimum amounts of traces ($\times 1,000$) required for a successful attack across all other dimensions and both original and recompiled design, when considering only specific design space parameters.

Overall average	42.6			
Local placement strategy	Default 26.5	Perf. 47.1	Area 56.4	Power 40.2
Sensor location	0 51.5	7 48.1	30 44.4	28 60.1
AES location	0 37.4	30 59.9	42 58.7	46 48.1
Board	A 37.1	B 58.8	C 50.1	D 58.1
Attacker-victim distance	adjacent 57.8		non-adjacent 40.4	

- The *local placement strategy* of the AES modules seems to have the strongest influence on the attack success, although the difference in required traces is only a little more than 2× for default and area-optimized placement.

- The *board* parameter, which corresponds essentially to the inter-chip PV, shows that board A, which is also the oldest board, is the most vulnerable of the four.

- Considering the *global sensor location* on the FPGA fabric, we see that it seems to be the least important parameter. We are unable to deduct a board-independent sensor location that has the best or worst attack success, when averaging over all other parameters.

- Although the impact of the *global AES location* on the FPGA fabric is not strong either, we determine location 0 to be the most vulnerable one on average. This concurs with the results of the previous experiments in Subsection 5.1 on voltage noise impact depending on global module placement, from which location 0 is expected to be the most vulnerable. However, when applying the power-optimized local placement constraints, location 0 seems to be actually one of the less vulnerable ones.

- Regarding the *design distance*, attacks are not necessarily easier, when the attacker design is placed closer to the victim design. In the last row of Table 2, we compare the attack success for adjacently and non-adjacently placed AES modules and sensors. We see that in fact attacks are slightly easier on average between the non-adjacent locations.

As the differences in the independent results in Figure 9 and Figure 10 reach up to 500×, we conclude that combinations of multiple design space parameters need to be considered in order to eventually improve security in a multi-tenant FPGA. When considering the location of the AES module, we see that for a specific local primitive placement strategy, we are clearly able to infer a less vulnerable and a more vulnerable location across different boards. The most definite example would be the AES location 30, which is the least vulnerable for all boards when placed with the performance or area optimizing strategy. In the same way, we see how location 0 is most vulnerable, when using performance-optimized mapping.

In general, a dependence of CPA success on the global placement can only be identified w.r.t. a specific local placement strategy. However, we can draw some general conclusions from the evaluation of the 50×50 locations in the previous subsection. In those experiments, the sensitivity of a specific sensor location was very different across boards, whereas the impact of a specific noise generation module was very similar. This initial result is reflected in the results in Figure 9, Figure 10 and Table 2 where the location of the AES module is more critical to the attack success as, for example, the board-independent low vulnerability of the AES module in location 30 shows.

We want to mention that results are also different, if we attack another AES round key byte. This is expected in our setup, as the implemented AES module uses four independently placed S-Boxes. For the amount of traces required in a successful CPA, however, the placement of S-Box and state register is critical. Attacking a different byte is thus equivalent to attacking a different local placement. In the appendix, we exemplary show the results on attacking the second round key byte instead of the first one on only a single board.

Finally, we present results of evaluating attack success on the Virtex-7-based ADM-PCIE-7V3, to confirm our findings on design parameter impact on SCA vulnerability across different platforms.

The attack is generally harder on the ADM-PCIE-7V3 which could be due to various reasons, such as a more stable power supply or the hierarchical layout of the larger FPGA with different clock regions. Thus, we need to collect up to $10M$ traces for an

		AES module locations			
		0	30	42	46
Sensor locations	0	290	10,000	1,340	10,000
	7	4,560	10,000	780	2,210
	30	2,170	90	2,490	2,270
	28	5,210	10,000	250	2,060

Figure 11: Minimum amount of traces (×1, 000) required for all possible combinations of AES and sensor location on a Virtex-7-based ADM-PCIE-7V3 accelerator card. Here, a value of $10M$ means that we were unable to recover the key with $10M$ traces.

adequate comparison of the required amount of traces for a successful CPA. However, our key observation – the significant impact of design space parameters on side-channel vulnerability – is clearly visible in the results on this platform as well. In Figure 11, we see again differences in the minimum amount of traces required for a successful attack of more than 100×, depending on the global placement parameters only. An AES module in location 30 can be attacked by a sensor in location 30 with only $90k$ traces, whereas other combinations are not vulnerable with up to $10M$ traces.

5.3 Impact of Physical Design Parameters on SCA Countermeasures

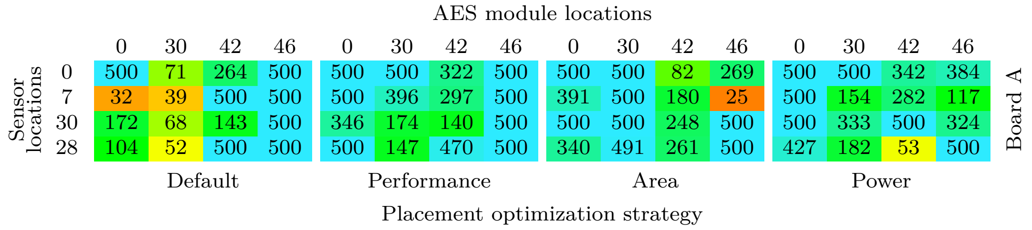
Last but not least, we implement a noise-generation-based hiding countermeasure, as explained in Subsection 3.4, and evaluate the success of attacks on a protected design on board A. To acquire comparable results, the countermeasure is implemented in a way, that allows to disable the noise-generating RO array without recompiling the bitstream. First, we need to evaluate the unprotected modules again as we cannot compare the protected implementation with the previous results due to the changes to local placement and routing introduced by the countermeasure. The results of the baseline evaluation are presented in Figure 12. As expected, the results differ from the previous experiments but are well within the expected range.

Next, the countermeasure is enabled and up to $500k$ traces collected to account for the increased attack difficulty. Figure 13 shows the attack success of CPA on the protected AES modules. We observe an expected significant increase in the minimum amount of traces required for a successful key recovery, in many setups the design cannot be attacked even with $500k$ measurements. However, whereas in some configurations the countermeasure can raise the amount of traces required by a factor of over 260×, in other settings the increase is only minimal. For a very small subset of the physical design parameters, the amount of traces even decreases with the enabled countermeasure.

These results clearly show that the effectiveness of SCA countermeasures are greatly influenced by physical design and mapping parameters. Although hiding countermeasures come with significant area and power overhead, by carefully considering the impact of local and global mapping, one can achieve similar level of protection with virtually zero-overhead. From a different perspective, neglecting the effects of physical design can significantly hinder the success of such explicit countermeasures.

		AES module locations																
		0	30	42	46	0	30	42	46	0	30	42	46	0	30	42	46	
Sensor locations	0	18.6	4.4	63.7	7.6	3.4	41.9	100	84.4	100	1.9	10.4	25.8	35.6	15.7	12.4	7.8	Board A
	7	10.3	2.7	24.4	15.2	88.9	10	32.3	53.9	94.2	100	31.3	58.6	100	11.5	21	9.5	
	30	1.9	1.8	30.1	10.2	4.8	16.3	52.5	54.1	37.1	71.7	26.8	24.7	17.8	14.6	17.9	7.3	
	28	11.6	2.7	57.2	40.8	10.7	1	100	100	31.4	100	79.5	44.2	100	30.5	74.6	30.2	
		Default				Performance				Area				Power				
		Placement optimization strategy																

Figure 12: Minimum amount of traces (×1, 000) required for a successful attack on board A without the RO-based countermeasure enabled.



	Default				Performance				Area				Power			
	0	30	42	46	0	30	42	46	0	30	42	46	0	30	42	46
0	500	71	264	500	500	500	322	500	500	500	82	269	500	500	342	384
7	32	39	500	500	500	396	297	500	391	500	180	25	500	154	282	117
30	172	68	143	500	346	174	140	500	500	500	248	500	500	333	500	324
28	104	52	500	500	500	147	470	500	340	491	261	500	427	182	53	500

Figure 13: Minimum amount of traces ($\times 1,000$) required for a successful attack with up to 500k traces on a design protected by our simple RO noise generator on board A. Blue cells reflect setups where up to 500k traces are required. Here, a value of 500k means that we were unable to recover the key with 500k traces.

6 Discussion

After reporting our results in the previous section, we would like to discuss them and carve out consequences for secure virtualization and multi-tenancy of FPGAs in this section.

6.1 Impact of Physical Design Parameters on Side-Channel Security

The most important conclusion to draw from our experiments is the importance of physical design parameters w.r.t side-channel vulnerability. We observe significant differences in the minimum amount of required measurements to successfully recover the secret AES key. On our experimental platforms, we are sometimes able to recover the key byte with only 200 traces, whereas other parameter settings prevent key recovery even with 100k traces. Those differences are well within the range of some side-channel countermeasures, making physical design mapping almost as critical.

Experimenting with an exemplary noise-based hiding scheme shows how critical the dependence on mapping parameters actually is. We reach an increase in the minimum amount of traces required for key recovery by more than 260× in some cases, whereas in other setups the countermeasure does not help at all to prevent side-channel leakage. Researchers already pointed out the importance of placement and routing for specific masking countermeasures [CEM18], but our systematic analysis shows its general impact, especially in the context of FPGA-internal attacks.

The presented results demonstrate the high influence of local placement and routing on the SCA vulnerability of victim modules. This is also the reason, why a surrogate model of the module activity based on the global placement on the FPGA fabric (represented by noise generating modules in the first set of our experiments) does not correlate with the success of CPA on AES modules for actual key recovery attacks. A challenge to the research community will be to investigate physical design and mapping from a side-channel security perspective.

Our evaluation of attack success on a data-center scale Virtex-7 based device proves the impact of design space parameters on larger FPGAs as well. Moreover, we observed how the layout and organization of the larger fabric further increases the design space and thus the variation in side-channel vulnerability. Modern high-end FPGAs are often composed of multiple dies, where side-channel attacks have been shown to be still possible [GRS19]. However, those multi-die chips add another design space parameter to be considered.

Determining the exact parameters and characteristics of the underlying FPGA architecture and hardware, which lead to the observed effects, goes beyond the scope of this paper. Nevertheless, we assume two causative factors: The PDN design is non-uniform across the chip, which leads to differences in the impact of current draw on the supply voltage, and the mapped user logic components are subject to intra-chip PV. Both factors, on one hand, impact the sensitivity of sensors in different locations and, on the other hand, lead to differences in the voltage traces caused by modules in different locations. However, a thorough knowledge of the architecture and underlying hardware is reserved to the vendors. With some FPGAs being actively reverse-engineered by a growing community, we may be able to unveil dependencies between specific placement and routing and side-channel vulnerability, despite the lack of knowledge about the underlying PDN architecture.

6.2 Possible Countermeasures based on Physical Design Parameters

Our results do not necessarily only introduce a new problem to the area of defending against on-chip side-channel attacks, they may be also useful in developing new countermeasures. We may be able to leverage the high influence of placement and routing to optimize algorithms for security.

Countermeasures based on design space parameters on both hypervisor and user side will require thorough, per-device analysis. This surely requires – depending on the device size and the approach – a significant amount of effort and time.

A possible approach could be built on three steps:

- First, a hypervisor generates multiple local placement mappings for a specific crypto-module.
- Then, for each mapping of the module, a global analysis determines regions, which are less vulnerable to side-channel attacks. Evaluating actual attacks for all possible combinations is hardly feasible. However, in future works we may be able to identify an adequate model similar to our noise generator approach in Subsection 4.2, which can assess side-channel vulnerability in less time than a full attack.
- Nevertheless, in the third and final step, using the results for improving side-channel security on a specific FPGA comes at zero overhead. On the hypervisor side, a global map of secure locations and precompiled cryptocores can be provisioned, which can be deployed by the user as a building block in security-critical applications.

This approach could improve security without requiring any additional area or resources of the FPGA and can be used in combination with other well-known SCA countermeasures [KBG09, MMMT09, CEM18, WMG18, MGV08, BSP$^+$19].

All in all, our work points out an important aspect of the development of countermeasures against side-channel attacks in virtualized multi-tenant FPGAs.

7 Conclusion

In this work, we thoroughly analyze the impact of physical design mapping parameters in multi-tenant FPGAs on the success of SCA attacks. The results of more than 256 experiments with CPA attacks on an AES FPGA implementation with up to 100k measurement traces reveal differences in the required amount of traces for key recovery of up to several hundred times. We show that the attack success depends on where attacker and victim modules are placed on the FPGA, how exactly the primitives within the module are locally placed, and on chip-to-chip variation.

Moreover, our experiments on modules, which are protected by a noise generator hiding scheme, prove that SCA countermeasures can be more or less effective under certain design mappings. Consequently, those parameters are of great importance when implementing protected cryptographic designs in multi-tenant FPGAs.

However, our analysis does not only introduce a new aspect to the application of SCA countermeasures on virtualized FPGAs, but may also be a starting point for future works on hiding leakage through design mapping only. Such countermeasures would come at zero overhead and would provide a valuable asset in securing multi-tenant FPGA access against internal side-channels. On the other hand, neglecting such device and physical design dependencies can compromise security of virtualized FPGAs in the cloud.

References

[AWS19] Amazon AWS. Amazon EC2 F1 instances, 2019. URL: https://aws.amazon.com/ec2/instance-types/f1/.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. *Cryptographic Hardware and Embedded Systems - CHES 2004*, 2004.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology — EUROCRYPT '97*. Springer Berlin Heidelberg, 1997.

[BSB+14] Stuart Byma, J Gregory Steffan, Hadi Bannazadeh, Alberto Leon Garcia, and Paul Chow. FPGAs in the cloud: Booting virtualized hardware accelerators with openstack. In *International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2014.

[BSP+19] Nahome Bete, Fareena Saqib, Chintan Patel, Ryan Robucci, and Jim Plusquellic. Side-channel power resistance for encryption algorithms using dynamic partial reconfiguration (SPREAD). *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2019.

[CEM18] Thomas De Cnudde, Maik Ender, and Amir Moradi. Hardware masking, revisited. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, Volume 2018, 2018.

[CLO19] HUAWEI CLOUD. FPGA-accelerated cloud server, 2019. URL: https://www.huaweicloud.com/en-us/product/fcs.html.

[Cor17] Intel Corporation. Intel FPGAs power acceleration-as-a-service for alibaba cloud | intel newsroom, 2017. URL: https://newsroom.intel.com/news/intel-fpgas-power-acceleration-as-a-service-alibaba-cloud.

[EV12] Ken Eguro and Ramarathnam Venkatesan. FPGAs for trusted cloud computing. In *International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2012.

[FVS15] Suhaib A. Fahmy, Kizheppatt Vipin, and Shanker Shreejith. Virtualized FPGA Accelerators for Efficient Cloud Computing. In *CloudCom*. IEEE, 2015.

[GM11] Tim Güneysu and Amir Moradi. Generic side-channel countermeasures for reconfigurable devices. In *Cryptographic Hardware and Embedded Systems – CHES 2011*. Springer Berlin Heidelberg, 2011.

[GNM+13] Benjamin Gojman, Sirisha Nalmela, Nikil Mehta, Nicholas Howarth, and André DeHon. GROK-LAB: Generating real on-chip knowledge for intra-cluster delays using timing extraction. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '13. ACM, 2013.

[GOKT18] Dennis R. E. Gnad, Fabian Oboril, Saman Kiamehr, and Mehdi B. Tahoori. An experimental evaluation and analysis of transient voltage fluctuations in FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(10), 2018.

[GRKT18] Dennis R. E. Gnad, Sascha Rapp, Jonas Krautter, and Mehdi B. Tahoori. Checking for electrical level security threats in bitstreams for multi-tenant FPGAs. In *2018 International Conference on Field-Programmable Technology (FPT)*. IEEE, 2018.

[GRS19] Ilias Giechaskiel, Kasper Rasmussen, and Jakub Szefer. Reading between the dies: Cross-SLR covert channels on multi-tenant cloud FPGAs. In *IEEE International Conference on Computer Design (ICCD)*, 2019.

[JL09] Jianwei Dai and Lei Wang. A study of side-channel effects in reliability-enhancing techniques. In *International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2009.

[KBG09] Najeh Kamoun, Lilian Bossuet, and Adel Ghazel. Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher. In *2009 3rd International Conference on Signals, Circuits and Systems (SCS)*. IEEE, 2009.

[KGS+19] Jonas Krautter, Dennis R.E. Gnad, Falk Schellenberg, Amir Moradi, and Mehdi B. Tahoori. Active Fences against Voltage-based Side Channels in Multi-Tenant FPGAs. Cryptology ePrint Archive, Report 2019/1152, 2019.

[KGT18] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. FPGAhammer: remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2018(3), 2018.

[KGT19] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. Mitigating electrical-level attacks towards secure multi-tenant FPGAs in the cloud. *ACM Transactions on Reconfigurable Technology and Systems*, 12(3), 2019.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology — CRYPTO' 99*. Springer Berlin Heidelberg, 1999.

[KLP+18] Ahmed Khawaja, Joshua Landgraf, Rohith Prakash, Michael Wei, Eric Schkufza, and Christopher J Rossbach. Sharing, protection, and compatibility for reconfigurable fabric with amorphos. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2018.

[LB09] Lang Lin and Wayne Burleson. Analysis and mitigation of process variation impacts on power-attack tolerance. In *Proceedings of the Design Automation Conference (DAC)*, DAC '09. ACM, 2009.

[LCL10] Po-Chun Liu, Hsie-Chia Chang, and Chen-Yi Lee. A low overhead DPA countermeasure circuit based on ring oscillators. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(7), 2010.

[Lin18] FPGA device feature list (DFL) device drivers, 2018. URL: https://lwn.net/Articles/757283/.

[MGV08] Nele Mentens, Benedikt Gierlichs, and Ingrid Verbauwhede. Power and fault analysis resistance in hardware through dynamic reconfiguration. In *Cryptographic Hardware and Embedded Systems – CHES 2008*. Springer Berlin Heidelberg, 2008.

[MMMT09] Robert P. McEvoy, Colin C. Murphy, William P. Marnane, and Michael Tunstall. Isolated WDDL: A hiding countermeasure for differential power analysis on FPGAs. *ACM Transactions on Reconfigurable Technology and Systems*, 2(1), 2009.

[MSM+14] Massimo Alioto, Simone Bongiovanni, Milena Djukanovic, Giuseppe Scotti, and Alessandro Trifiletti. Effectiveness of Leakage Power Analysis Attacks on DPA-Resistant Logic Styles Under Process Variations. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(2), 2014.

[NIS01] NIST. Advanced encryption standard (AES). Technical report, 2001.

[RPD+18] Chethan Ramesh, Shivukumar B Patil, Siva Nishok Dhanuskodi, George Provelengios, Sebastien Pillement, Daniel Holcomb, and Russell Tessier. FPGA side channel attacks without physical access. In *International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2018.

[RSVC+11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2011.

[SGMT18a] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi, and Mehdi B. Tahoori. Remote inter-chip power analysis side-channel attacks at board-level. In *Proceedings of the International Conference on Computer-Aided Design - ICCAD 2018*. ACM Press, 2018.

[SGMT18b] Falk Schellenberg, Dennis R.E. Gnad, Amir Moradi, and Mehdi B. Tahoori. An inside job: Remote power analysis attacks on FPGAs. In *Proceedings of Design, Automation & Test in Europe (DATE)*, 2018.

[WMG18] Alexander Wild, Amir Moradi, and Tim Guneysu. GliFreD: Glitch-free duplication towards power-equalized circuits on FPGAs. *IEEE Transactions on Computers*, 67(3), 2018.

[Wu10] Jinyuan Wu. Several Key Issues on Implementing Delay Line Based TDCs Using FPGAs. *IEEE Trans. Nucl. Sci.*, 57(3):1543–1548, 2010. doi:10.1109/TNS.2010.2045901.

[WYR+13] Xinmu Wang, Wen Yueh, Debapriya Basu Roy, Seetharam Narasimhan, Yu Zheng, Saibal Mukhopadhyay, Debdeep Mukhopadhyay, and Swarup Bhunia. Role of power grid in side channel attack and power-grid-aware secure design. In *Proceedings of the 50th Annual Design Automation Conference on - DAC 2013*. ACM Press, 2013.

[Xil16] Xilinx. 7 series FPGAs configurable logic block - user guide (v1.8), 2016.

[Xil19] Xilinx. Alveo U280 data center accelerator card data sheet, 2019.

[YXL10] Haile Yu, Qiang Xu, and Philip H W Leong. Fine-grained characterization of process variation in FPGAs. In *International Conference on Field-Programmable Technology (FPT)*. IEEE, 2010.

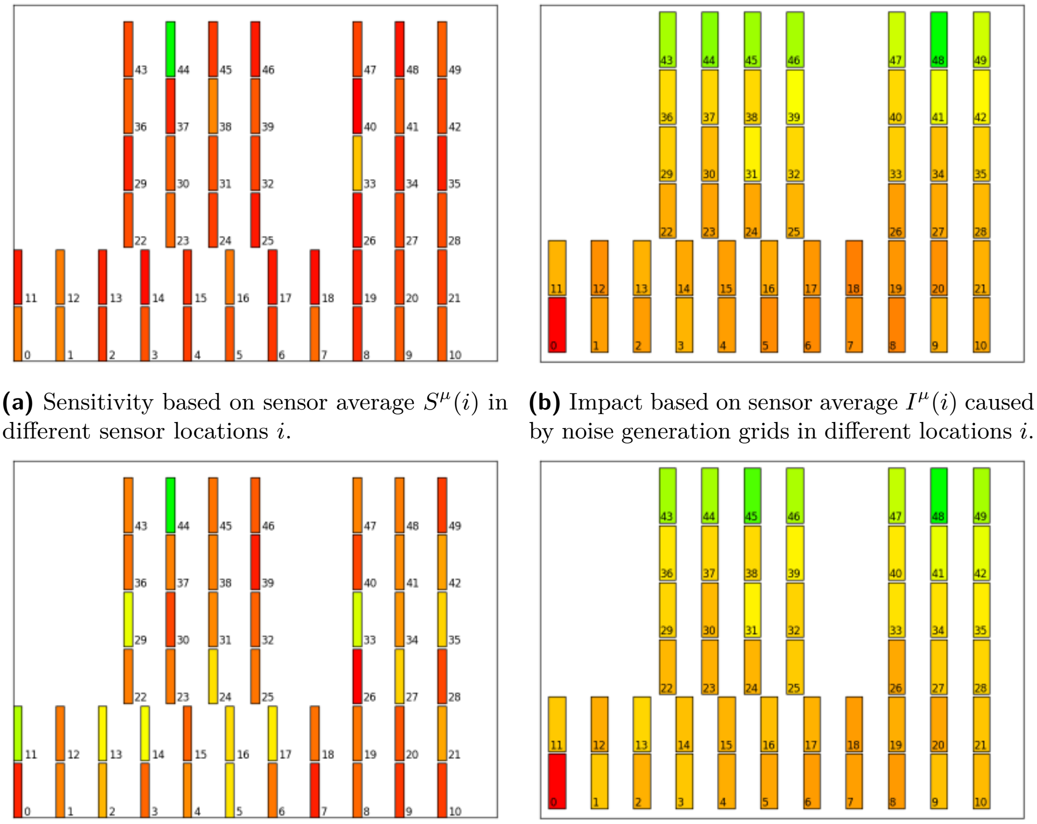
[ZAB+18] Shuze Zhao, Ibrahim Ahmed, Vaughn Betz, Ashraf Lotfi, and Olivier Trescases. Frequency-Domain Power Delivery Network Self-Characterization in FPGAs for Improved System Reliability. *IEEE Transactions on Industrial Electronics*, 2018.

[ZH12] Kenneth M Zick and John P Hayes. Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2012.

[ZS18] Mark Zhao and G. Edward Suh. FPGA-based remote power side-channel attacks. In *Symposium on Security and Privacy (S&P)*, 2018.

[ZSZF13] Kenneth M. Zick, Meeta Srivastav, Wei Zhang, and Matthew French. Sensing nanosecond-scale voltage attacks and natural transients in FPGAs. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays (FPGA)*. ACM Press, 2013.

A Influence of Global Placement on Board B



(a) Sensitivity based on sensor average $S^{\mu}(i)$ in different sensor locations i.

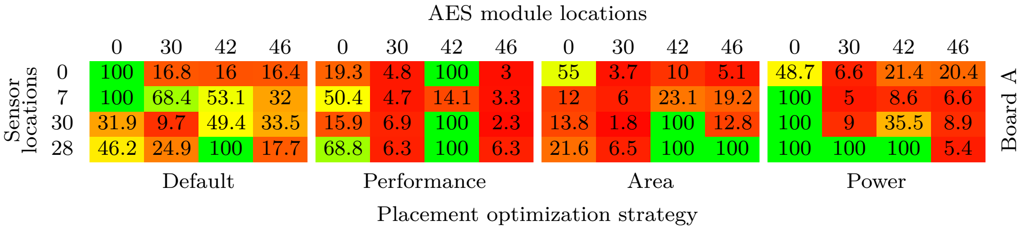
(b) Impact based on sensor average $I^{\mu}(i)$ caused by noise generation grids in different locations i.

(c) Sensitivity based on sensor variance $S^{\sigma}(i)$ in different sensor locations i.

(d) Impact based on sensor variance $I^{\sigma}(i)$ caused by noise generation grids in different locations i.

Figure 14: Influence of global placement of FF-based noise generators and TDC sensors evaluated and averaged over 1000 measurements on board B. Locations that are colored orange/red correspond to sensors that are more sensitive or noise generators that cause a higher impact respectively. Green colored locations are less sensitive or cause less impact.

B Evaluating Attacks on the second AES Key Byte on Board A



Sensor locations	Default 0	Default 30	Default 42	Default 46	Performance 0	Performance 30	Performance 42	Performance 46	Area 0	Area 30	Area 42	Area 46	Power 0	Power 30	Power 42	Power 46
0	100	16.8	16	16.4	19.3	4.8	100	3	55	3.7	10	5.1	48.7	6.6	21.4	20.4
7	100	68.4	53.1	32	50.4	4.7	14.1	3.3	12	6	23.1	19.2	100	5	8.6	6.6
30	31.9	9.7	49.4	33.5	15.9	6.9	100	2.3	13.8	1.8	100	12.8	100	9	35.5	8.9
28	46.2	24.9	100	17.7	68.8	6.3	100	6.3	21.6	6.5	100	100	100	100	100	5.4

Figure 15: Minimum amount of traces (×1, 000) required to attack the second round key byte on board A. Red colored cells correspond to combinations where an attack is easy, green colored to parameters resulting in a difficult attack. A value of $100k$ means that we were unable to recover the key with $100k$ traces.