High-confidence Remote Power Analysis on Heterogeneous SoCs

Anis Fellah-Touta, Lilian Bossuet, Carlos Andres Lara-Nino

High-confidence Remote Power Analysis on Heterogeneous SoCs

Anis Fellah-Touta[1], Lilian Bossuet[1] and Carlos Andres Lara-Nino[2]

1. UJM St-Etienne, CNRS, LabHC UMR 5516, SAINT-ETIENNE, France.

2. Universitat Rovira i Virgili, DEIM, TARRAGONE, Spain.

Correspondence: anis.fellah.touta@univ-st-etienne.fr

January 7, 2025

Abstract

In recent years, significant advances have been made in side-channel analysis, particularly in the design of attack methodologies targeting SoC-FPGAs. These devices have become increasingly popular in cloud data centers thanks to their flexibility and efficiency. As a result, there has been a growing number of proposals for sharing FPGA fabrics among multiple users in cloud environments. However, even when logical isolation is used as a protective measure for each tenant, the presence of a multiple-tenants in the FPGA environment raises significant concerns about potential security threats. Recent works have revealed the possibility of power side-channel attacks being executed in a cloud-FPGA environment, even without direct physical access to the platform. These attacks can be carried out by a malicious user, leveraging delay sensors implemented using internal FPGA resources. These sensors have the ability to monitor the power consumption of a circuit, thus giving a malicious user insights into the internal operations of the SoC-FPGA and potentially enabling extraction of sensitive information. The primary challenge for successful remote power analysis lies in accurately cutting and aligning power traces. Secondary digital channels with trigger information are typically used for this purpose. This paper presents a novel method that simplifies the conditions necessary for a remote power attack. The approach mitigates the need to connect digital triggers to the remote sensor, thereby reducing the complexity of the attack setup. To validate the efficacy of the proposed method, a successful key recovery was performed on a hardware implementation of an AES cipher.

Keywords: FPGA, internal sensor, AES, RPA, ring oscillator, TDC.

Please cite as:

```
@article{BFL24c,
title = {{High-confidence Remote Power Analysis on Heterogeneous SoCs}},
author = {Fellah-Touta, Anis and Bossuet, Lilian and Lara-Nino, Carlos Andres},
journal = {J. Hardware Syst.  Secur.},
numpages = {16},
volume = {9},
number = {4},
year = {2024},
doi = {10.1007/s41635-024-00155-5}}
```

1 Introduction

Field-programmable gate arrays (FPGAs) play a crucial role in modern computing. They showcase the importance of adaptability in computational hardware, and represent a significant advance in semiconductor technology. In contrast to Application-Specific Integrated Circuits (ASICs), which are carefully designed for specific tasks, FPGAs provide a dynamic platform where logic gates and connections can be configured and reconfigured to suit the specific requirements of the application. The flexibility of FPGAs allows them to play a variety of roles in a wide range of computing applications. They are especially powerful in cloud computing [LHZ21]. In this context, the adaptability of FPGAs is a real advantage. It means that computer resources can be adjusted on the fly to match the specific needs of different tasks. This ability to change things quickly has been very useful in situations where tasks change frequently or when specific processing capabilities are crucial. Additionally, FPGAs are known for their efficiency, making them even more valuable in scenarios where resource optimization is crucial [Qas+19]. The integration of FPGAs into cloud data centers has brought about a significant change in how computational resources are provided. Conventionally, resource allocation was constrained, specific tasks were often assigned to individual computational units. The arrival of FPGAs brought in an era of remarkable flexibility, allowing multiple users to use FPGA resources at the same time [Asi+17; YB17; FVS15]. To a great extent, this major change was made possible, by the careful implementation of logical isolation measures. These measures act like digital walls that separate FPGA resources for each user. This creates a sense of independence in computing, ensuring that one user's actions do not affect anyone else's. At the same time, these measures act as strong guards, protecting the security and privacy of important data in this shared environment.

Side-channel attacks involve the exploitation of unintended information leakage, such as power consumption patterns or electromagnetic emissions, during the operation of a system. Fault injection attacks represent another threat vector. This method involves deliberately inducing errors or faults in a system's operation. Adversaries can tamper with FPGA operation by injecting faults into the logic gates or interconnections that potentially enable the extraction of highly sensitive information.

While the multi-tenant FPGA model holds significant promise, it also introduces its own set of challenges. Recent research brought to light a previously neglected vulnerability - a scenario characterized by power side-channel attacks within cloud-based FPGA systems [Sch+21; ZS18; Gra+20; Gla+20]. This is an important discovery since it suggests that the attacks can be executed remotely, eliminating the need for physical access to the target platform, as was the case in traditional side-channel attacks. In this new method of attack, malicious actors demonstrate a high level of expertise in manipulating internal FPGA resources, with a particular emphasis on using the resources themselves to construct delay sensors. In this way, they examine the circuit's power consumption patterns and provides them with crucial insights into the functioning of the FPGA, potentially enabling the extraction of highly sensitive information.

The use of internal sensors made of digital components is not a recent concept [Fra+10; Hen10]. Traditionally such sensors were used to monitor circuit operation and to check the system is functioning as intended. However, previously it

was not considered feasible that such sensors could be used to extract sensitive information from the platform. Recent advances challenge this belief, as they revealed that sensors based on time-to-digital converters (TDC) [Gra+21] and ring-oscillators (RO) [Gra+19] can in fact be used to perform power analysis with reasonable precision. Furthermore, the circuits can be implemented and operated remotely, thus eliminating the need for physical access to the target platform. This means that attackers no longer require physical proximity to the target, instead they can perform power analysis and potentially retrieve sensitive information remotely.

A major challenge to remote power analysis (RPA) is obtaining additional information needed to cut and align traces. In standard power analysis, if attackers have physical access to the platform, it is assumed that they can explore and locate a trigger signal that marks the beginning of a trace. This trigger signal helps synchronize and align the collected power traces. However, in the case of a remote attack model, this process becomes considerably more complex. We cannot assume that the target's *start* and *done* signals, which typically serve as clear markers, will be accessible to the sensor. Therefore, determining the precise point for cutting and aligning traces remotely becomes a significant and timely problem that is not properly addressed in the literature.

This paper presents an innovative solution to this significant challenge. Our approach is based on the use of a frequency-based covert channel, a method known for its ability to transmit information discreetly. By using this covert channel, we can accurately encode data that are essential for the synchronization of traces. Notably, this approach stands out for its efficiency in covertly transmitting information, thereby effectively bypassing conventional methods of detection.

To demonstrate the feasibility of the approach, the RPA of an unprotected implementation of AES-128 was performed. The results indicate that the proposed approach is feasible, as the power traces were aligned with high accuracy, and the secret key was successfully recovered.

The rest of the paper is organized as follows. Section 2 reviews side channel attacks and covert channels. Section 3 describes the methodology and the materials used in the experiment and provides a formal description of the proposed attack scenario. Section 4 presents the results and Section 5 discusses potential countermeasures. Finally, Section 6 summarizes the findings and draws a number of conclusions.

2 Literature review

2.1 SoC-FPGAs

A System-on-Chip Field-Programmable Gate Array (SoC-FPGA) is a cutting-edge integrated circuit that combines the capabilities of a FPGA with the integration features of an ASIC. The architecture of a SoC-FPGA encompasses reconfigurable FPGA fabric, comprising a matrix of configurable logic cells and interconnections for dynamic adaptability. This fabric is coupled with dedicated hard intellectual property (IP) blocks, including microprocessors, memory controllers, and peripherals, thus creating a heterogeneous computing environment. SoC-FPGAs are ideal for applications that require a balance between flexibil-

ity, performance, and power efficiency. Their ability to integrate dedicated processors with custom hardware accelerators makes them suitable for diverse applications.

In addition to their architectural dimension, the significance of the power distribution network (PDN) within SoC-FPGAs should not be underestimated. Efficient management of power is critical for SoC-FPGAs to guarantee optimal performance, energy efficiency, and thermal management. Typically, SoC-FPGAs incorporate a shared PDN that supplies power to both the programmable logic fabric and the hard IP blocks, including processors and peripherals. This shared PDN has several advantages, such as reduced overall power consumption and simpler design.

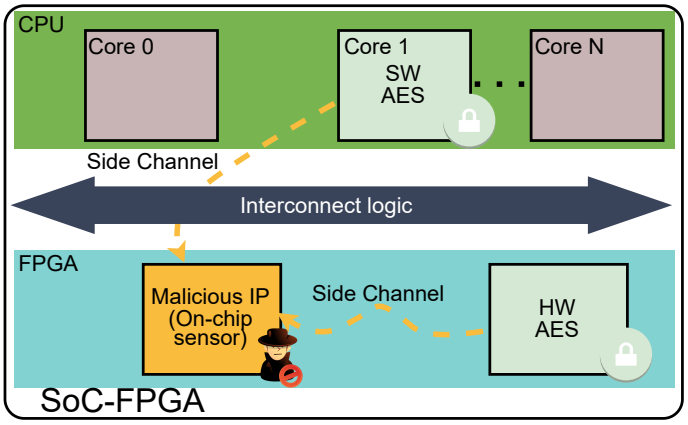
While SoC-FPGAs offer immense flexibility and versatility, they also involve challenges. Shared PDNs can cause variations in voltage and current, potentially affecting both stability and reliability. Additionally, the shared nature of the PDN can increase exposure to power-based side-channel attacks that require robust security measures. Our work further explores the reach and potential of this family of attacks.

2.2 Side channel attacks

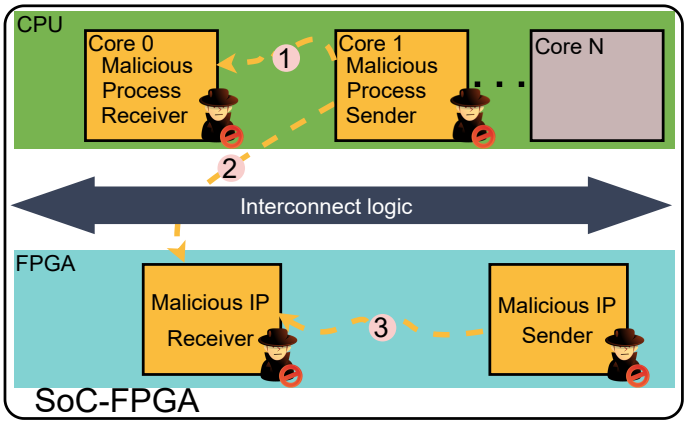
Side channel attacks refer to a class of attacks that exploit unintended information leakage from a system's physical implementation to compromise its security. Instead of directly targeting the underlying cryptographic algorithms, which are recognized as being secure from an algorithmic point of view, side channel attacks take advantage of observable side channel information such as power consumption [KJJ99], electromagnetic radiation [Agr+03], timing [Koc96], or acoustic emanations [HS15] during the execution of cryptographic operations.

In our work, we are interested in power analysis attacks that exploit variations in power consumption during encryption that originate from the switching activities within the device's components, such as logic gates and memory elements. The power consumption patterns can inadvertently leak information about the internal operations and, in turn, reveal sensitive data, such as cryptographic keys. By analyzing the resulting power traces and applying statistical techniques like Simple Power Analysis (SPA) [Koc96], Differential Power Analysis (DPA) [KJJ99] or Correlation Power Analysis (CPA) [BCO04], attackers can deduce the secret key and compromise the security of the system. In CPA, attackers typically collect multiple power traces while the cryptographic algorithm is being executed with varying inputs but the same secret key. They then perform statistical analyses, such as correlation coefficient computations, to identify patterns that align with the variations induced by specific intermediate values or computations. These patterns reveal information about the secret key and can be used to recover it.

In the context of SoC-FPGA, these power consumption measurements can be executed remotely and originate from within the circuit itself, challenging the previous assumption that physical access to the system was necessary. These attacks have been used to extract the secret key of an encryption module in scenarios where the attacker and victim use the same FPGA fabric but maintain logical isolation [Sch+21; Gra+19; ZS18]. Even when different FPGAs are used but share the same power supply [Sch+18], the potential for a power analysis attack remains. This vulnerability persists even when the victim is using a



(a) Remote power attacks with on-chip sensors

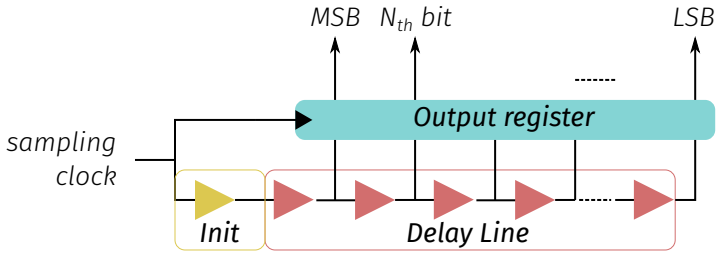


(b) Covert communication between CPU-CPU (1); CPU-FPGA (2); FPGA-FPGA (3)

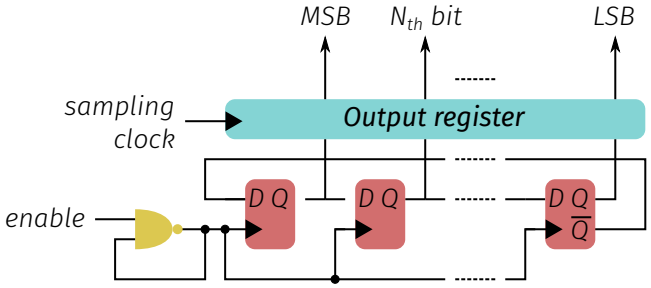
Figure 1: Two types of internal attacks on SoC-FPGAs

CPU while the attacker uses the FPGA embedded on the same SoC die [ZS18; Gra+20]. Fig. 1a provides a comprehensive overview of these remote attack scenarios based on on-chip FPGA sensors.

The possibility of conducting power attacks within the circuit arises from the inverse relationship between the supply voltage and the propagation delay of a CMOS inverter [ZMZ17]. Sampling the propagation delay makes it possible to discern the switching activity of the chip, and hence to distinguish between different computations. To measure this delay, a specialized circuit known as a delay sensor is required. This circuit is capable of measuring variations in the delay of an oscillatory digital signal. The propagation delay of an oscillator through digital components will fluctuate as a function of the temperature and voltage of the circuit. Consequently, as the chip performs different processing tasks, these actions will influence the delay propagation of the target oscillator. This delay will be measured and then sampled into a digital signal. Both TDCs and ROs can be used to implement delay sensors. The overall structure of these



(a) TDC-based sensor design



(b) RO-based sensor design

Figure 2: The architectures of the two main delay sensors used in RPA

two circuits is depicted in Fig. 2.

In the TDC-based sensor (see Fig. 2a), the initial delay block comprises a series of buffers that are crossed by a clock signal. The block has to be calibrated to be sure the edge of the input signal is properly recorded by the TDC-based sensor. Additionally, the delay line comprises a uniform sequence of buffers. The propagation of the clock signal within this line depends on the voltage of the chip. An increase in supply voltage reduces the propagation delay of this line and vice versa. To secure acquisition of each buffer's output, each output is linked to a latch activated by the clock signal. If the Hamming weight of the TDC-based sensor register escalates, this signals an increase in pulse propagation, which in turn, denotes a decrease in the propagation delay of the delay line, and conversely. The TDC-based sensor thus provides an accurate depiction of fluctuations in voltage within the circuit by converting the propagation delay into digital values that can be processed by a computer.

A ring oscillator is a specialized type of oscillator formed by an odd number of inverting gates interconnected within a closed-loop configuration. This arrangement produces continuous oscillation, in which the output feeds back into the input. The frequency of oscillation is directly determined by the propagation delay of the gates. The RO-based sensor in Fig. 2b is designed with a combinatorial loop featuring a NAND gate. The NAND gate's output serves as the clock for a counter that increments with each oscillation. The counter's value is sampled using a dedicated clock signal. Its value corresponds to the delay of the combinatorial loop. An increase indicates an increase in the loop's output frequency, and hence a decrease in propagation delay. Conversely, a decrease in frequency implies an increase in propagation delay. Deducing the output frequency thus provides valuable insights into voltage behavior within the circuit.

One of the main challenges of remote power analysis is aligning power traces. In a real scenario it is impractical, if not impossible, to capture a trigger signal that can be used to cut and align the traces. In [Sch+21], the authors describe a remote attack on AES-128 using TDCs as sensors. They suggest using the same sensors to create a trigger mechanism by enabling the start of sample storage when a large voltage drop is detected, for example, when the first round AES creates a significant variation in the output of the sensor. However, their technique is not completely reliable as it is susceptible to circuit noise that can prevent the mechanism from working properly. In our work, we use the same sensors as those used for RPA in previous studies [Sch+21; Gra+19; ZS18]. The attack model we consider is that the adversary only needs to retrieve a series of samples from the internal sensor to conduct RPA. Unlike in previous works [Sch+21], the dependence on the apparent voltage drop of the traces to perform the attack is eliminated. This feat was achieved by leveraging strategies previously used for covert-channel communications [BB18]. Consequently, a more generalized approach as chosen that remains effective even when the behavior of the target is not readily apparent. For instance, it can succeed in scenarios in which other operations are taking place on the same platform at the same time.

2.3 Covert channels

A covert channel is a communication mechanism that allows information to be transmitted between two entities in a way that bypasses normal channels or security mechanisms. Covert channels operate by exploiting unintended or hidden communication channels within a system, often leveraging resources or behaviors not intended for information transfer.

Various techniques for establishing covert channels are described in the literature, each of which relies on a distinct physical parameter. In a SoC-FPGA configuration, covert channels can emerge through different entities, facilitating hidden communication between the CPU and the FPGA fabric. Fig. 1b is a conceptual representation of the different scenarios that can give rise to covert channels. In the figure, Scenario "1" denotes covert communication between malicious processes on CPUs. Scenario "2" denotes covert communication between a malicious process on the CPU and a malicious IP implemented in the FPGA fabric. Scenario "3" denotes covert communication that can occur between different malicious IPs.

In their work, as detailed in [BB21], Benhani et al. explored the use of the time-sharing cache memory, a technique known as a cache-based covert channel. This method enables unauthorized communication within a SoC. Basically, it leverages the cache memory's time-sharing properties to establish a covert channel, thereby allowing concealed information to be exchanged between different components of the SoC.

In [GER19], the authors demonstrate the exploitation of cross-talk phenomena in long wires, specifically within Arria 10 SoC-FPGAs. Their work focused on showing that cross-talk effects are possible in various long wires present in these platform architectures. The basic idea behind this type of attack is that the information transmitted through a long wire across the fabric noticeably influences delays in neighboring long wires. These long wires are typically used to enhance routing efficiency and the overall performance of reconfigurable

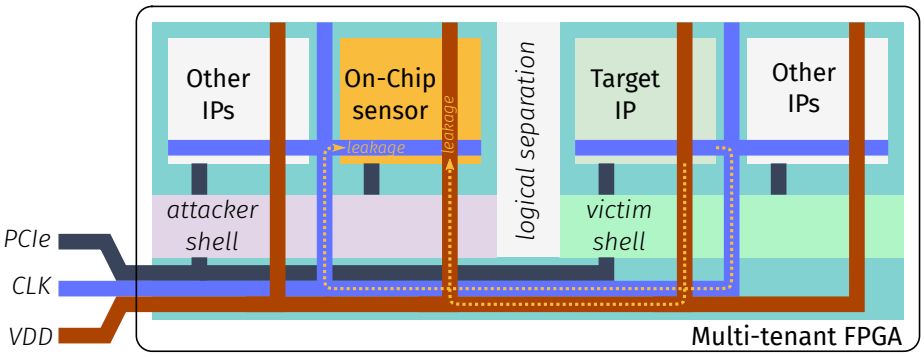


Figure 3: In the shared FPGA model, multiple users are logically isolated by fences, yet they share the same PDN through which power leakage propagates.

designs. However, when cross-talk occurs between the wires, it can result in unintended information leakage from one wire to another, particularly if the wires are in close proximity. This information leakage becomes a significant concern when the source wire is part of a secure region within the SoC, and the receiving wire connects to a potentially malicious IP. In such scenarios, it can lead to unauthorized information disclosure. Alternatively, if the transmitting entity is malicious and aims to transmit data to a different IP, the cross-talk between long wires can be exploited to establish covert communication channels. In [Ram+18], the former approach was demonstrated to launch a side-channel attack, specifically targeting the AES architecture. This indicates that the approach was not only capable of covert communication but also had practical applications in compromising security. To address the security vulnerabilities introduced by these attacks, potential countermeasures were introduced as described in [SMS20]. These countermeasures aimed to mitigate the risks associated with cross-talk attacks by isolating sensitive network connections from the other components of the FPGA architecture.

Another physical channel that can be exploited as a covert way of communication is the thermal covert channel. Masti et al. investigated the feasibility of using thermal sensors embedded in processor cores to establish covert communication between processes running on separate cores of the same processor [Mas+15]. Their approach centered on using core temperatures as a covert channel for discreetly conveying information. To send a logical 1, their malicious process would deliberately overload the core, causing the processor to overheat. Conversely, to transmit a logical 0, the malicious process would reduce the core's workload. Their receiving counterpart could then decode the data simply by collecting core temperature readings from the same processor. Tian and Szefer [TS19] adopted a comparable approach, illustrating the development of temporal-thermal covert channels in cloud-based FPGAs. Their work introduced the concept that the heat produced by one user's FPGA activity could be discerned by another user in a subsequent session, thereby suggesting the potential for data transmission even during relatively short periods of inactivity.

The power supply can also serve as a hidden communication channel. In the work conducted by Ziener et al. detailed in [ZBT10], the authors demon-

strate the feasibility of using the PDN of an FPGA to transmit data outside the circuit board. To send data bits, they used a large shift register, preloaded with a sequence of alternating ones and zeros that serve as the transmitter. When transmitting a logical 1, they set the shift register in operation, resulting in a noticeable increase in power consumption over a fixed number of cycles. Conversely, to transmit a logical 0, no shifting operation is initiated. The transmitted data can be decoded using any device capable of monitoring the power traces on the board. In their experiment, they used an oscilloscope for this purpose. The reported data transmission rate achieved was approximately 500 Kbps, regardless of the type of board or FPGA configuration. In [Gna+21], the authors introduced the concept of using the PDN to establish covert channels within the FPGA. They used non-combinatorial ring oscillators as transmitters that caused voltage fluctuations across the PDN, and TDC-based sensors as receivers. This approach achieved transmission rates of 8 Mbps with a minimal error rate of only 0.003%. However, only depending on logic isolation as a preventive measure against this class of internal attacks proves inadequate, primarily because the PDN serves as a common resource in the majority of SoC designs.

Alagappan et al. [Ala+17] pioneered the use of frequency modulation as a covert channel attack in sophisticated communication systems. Their work introduced a novel approach that demonstrated the efficacy of dynamic frequency adjustments for secure data transmission between covert processes and their designated recipients. The technique involves strategic overloading of the CPU to induce adaptive frequency changes, depending on the system's active governor mode. This manipulation enables the transmission of both logical ones and zeros. Decoding, which is accomplished through basic frequency readings, is then a straightforward process for the recipient. Additionally, [TSS17] exposed the CLKSCREW attack, which exploits vulnerabilities in dynamic voltage and frequency scaling (DVFS) mechanisms. This attack successfully extracted cryptographic keys and escalated privileges by injecting code into TrustZone, as shown also in [BB18].

The ease of using frequency-based covert channels lies in their effective integration in standard system operations. These channels rely on the fundamental principle of frequency modulation, making it challenging for external entities to detect or intercept covert communications. Moreover, the technique leverages existing hardware functionalities, thereby eliminating the need for specialized equipment or resources. These factors increase the attractiveness of frequency-based covert channels for both legitimate secure communications in sensitive environments and as potential vectors for malicious activities. This is why we chose frequency-based covert channels in our work to address the alignment challenge of power traces, providing a stable reference point for accurate alignment of the traces.

3 Combined Remote attack

This section provides details regarding the experimental setup. The different components of the system are described, along with guidelines for the proposed attack. All the experiments were performed on the Zynq-7000 SoC-FPGA.

3.1 Threat Model

The threat model we use follows the same threat model as the RPA attack [Gla+20; Ram+18], assuming a shared FPGA environment in a cloud or datacenter. In this scenario, each FPGA tenant is allocated a specific region of the FPGA. These regions are both logically and physically isolated from one another, although they all share the same PDN. Every tenant has the capability to implement any circuit in their designated FPGA section. Additionally, tenants can use customized FPGA logic, known as a shell, to access external interfaces such as PCIe and external DDR.

This threat model consists of two main actors: the victim shell and the attacker shell, as illustrated in Fig. 3. It is assumed that the victim responds to legitimate encryption requests from the attacker by encrypting plaintext using a hardware encryption accelerator implemented in the FPGA fabric. The resulting ciphertexts are then transmitted over a public channel to the attacker. Simultaneously, the attacker uses on-chip sensors to detect the leakage that propagates through the PDN and has the capability to extract power traces from outside the circuit.

3.2 Proposed combined attack

Fig. 4 illustrates our proposed approach, combining side channel and covert channel techniques. This combination addresses a critical challenge in RPA attacks, the remote alignment of traces, which has proven to be a significant obstacle in this type of attack.

The first attack vector uses a side channel technique. In this scenario, the attacker uses an on-chip sensor to detect the power leakage, in the form of voltage fluctuations that propagate through the PDN, generated by the targeted system, which in this case is AES. This leakage enables the extraction of the AES key, breaching the logical isolation between the attacker and the victim through an indirect connection via the PDN.

Simultaneously, the second vector uses a frequency-based covert channel. In this context, the covert channel serves to transmit synchronization data. These data are indispensable for accurate cutting and aligning of the traces received from the sensor, thereby guaranteeing the leakage patterns corresponding to the victim's activity are preserved. Just before each encryption call, the adversary adjusts the frequency of a reconfigurable phase locked loop (PLL) to covertly transmit information concerning synchronization to the acquisition system, while avoiding arousing suspicion or alerting the victim.

By simultaneously combining these attack vectors, the attacker's capabilities are enhanced, making the attack more complex and challenging to prevent.

3.3 Remote sensors

In this work, the RO-based sensors are used as described in [ZS18; Gra+19], together with TDC-based sensors described in [Gra+20], to acquire the data.

The main advantage of the RO-based sensor described in [Gra+19] is its acquisition rate. When implemented in the reconfigurable fabric, these sensors make it possible to use sampling frequencies of up to 250 MHz in the Zynq-7000 family of SoC-FPGAs, and even higher in newer technologies. This is

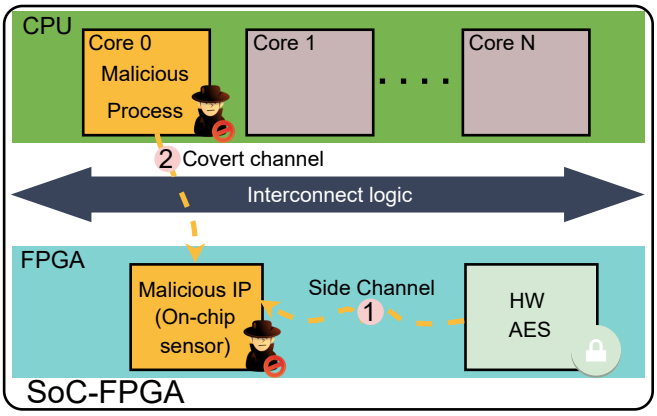


Figure 4: The proposed attack: Combining power analysis and covert channels.

achieved thanks to the use of a Johnson counter that mitigates the need for the carry propagation found in generic binary counters. The output register is 8-bits wide and captures the difference between counter states through a combinatorial function.

RO-based sensors have a smaller implementation size than TDC-based sensors, so they are easy to use, which also reduces the need for precise placement for viable implementation. These advantages are attractive, particularly considering that in RPA, sensor logic can be placed automatically. It also makes the attack less dependent on the technology. Avoiding a large hardware footprint is another potential advantage, however the trade-off is lower quality data acquisition. When a Johnson counter is used, its size determines the size of the quantization step. The dynamic range of the output will be associated with the sampling frequency. If the size of the counter is reduced (to 8-bits in our case) quantization errors will affect the measurement. To mitigate these errors, we use an array of RO-based sensors; the final output is a sum of their contributions. Using 64 RO-based sensors cadenced at 250 MHz has been found to reduce the quantization errors so that high resolution can be achieved [Gra+19]. For this reason, 64 RO-based sensors are used to perform the attack.

On the other hand, TDC-based sensors have higher sensitivity than RO-based sensors and can detect nanoscale-scale transients [Zic+13], but TDC-based sensors also have drawbacks. They require more logical resources, calibration of the initial delay, and precise placement. A recent study [KGT20] proposed using a TDC-based sensor that allows runtime calibration, thereby avoiding the need for calibration before the sensor is used.

We use both a TDC-based sensor and an RO-based sensor as on-chip voltage sensors. Our approach involves gathering power traces from both sensors, during which we will identify a common issue referred to as the misalignment problem. Subsequently, our overall aim is to develop a universal alignment technique tailored to each type of sensor. Our specific objective is then to demonstrate the effectiveness of this alignment method by conducting a CPA attack on AES.

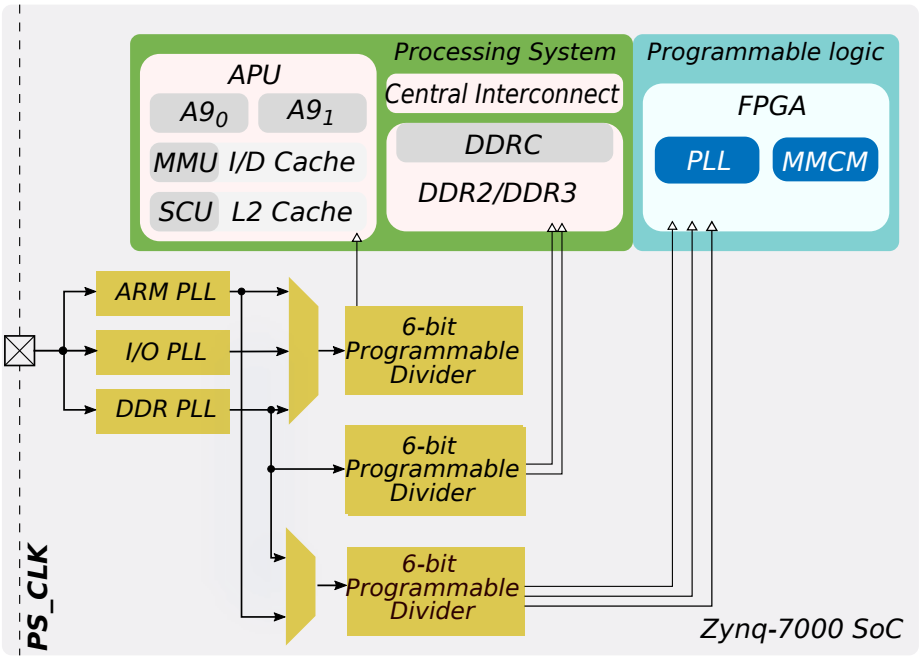


Figure 5: The clocking resources of Zynq-7000 SoC-FPGAs

3.4 Frequency-based covert channels

The approach we use to implement the frequency-covert channel in this work is inspired by the approach detailed in [BB18], in which the authors successfully demonstrated the feasibility of manipulating the output of PLL within Zynq7000 SoC-FPGA using a variety of modulation strategies. Their primary objective was to encode messages, with the aim of bypassing Trust-Zone protections. This enabled the establishment of a covert communication channel between distinct components that were not originally intended to communicate. To give an example, a trusted application could exchange information with an untrusted hardware accelerator, thereby demonstrating the potential security issues of such covert channels. In the present study, covert channels are not used to encode a message. Instead, the aim is to use a frequency modulation to align the power traces obtained by the delay sensor.

The Zynq-7000 SoC-FPGA includes a sophisticated array of clocking resources that play a critical role in managing timing and synchronization within the chip. The features include different clocks sourced from silicon PLLs, that flow from the processing system into the programmable logic, along with other components including processor cores, GPIO controllers, and communication interfaces. These clocks are derived from a group of main PLLs and, through a set of multipliers and dividers, produce the desired frequency output. These multipliers and dividers are simply digital values stored in registers, which can be modified from the processors of the SoC. On the other hand, reconfigurable PLLs offer a higher degree of flexibility. They can be programmed by the user and dynamically configured to generate custom clock frequencies. This adapt-

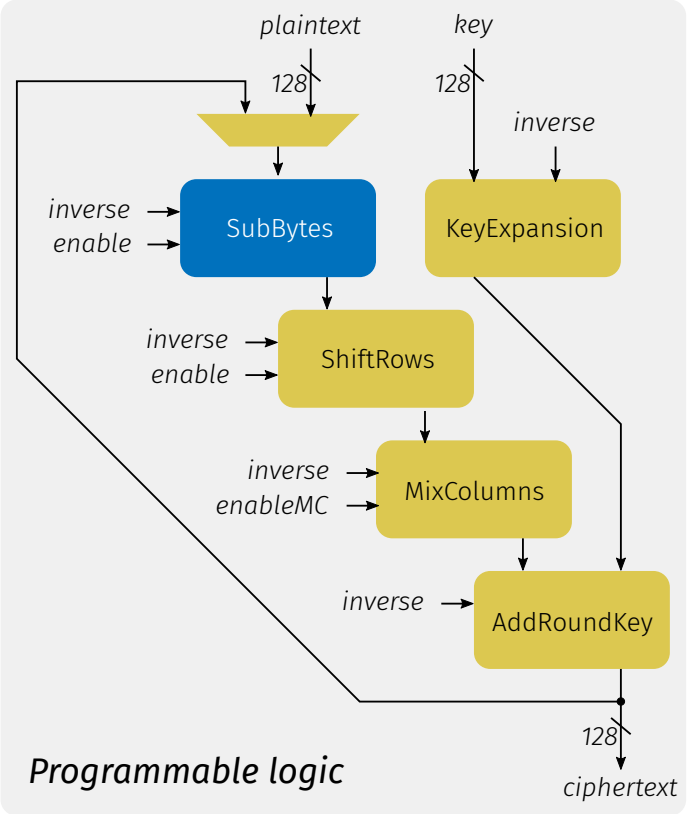


Figure 6: The AES architecture used as case study

ability is particularly valuable in scenarios where dynamically changing clock frequencies are needed. By combining these types of PLLs, the Zynq-7000 SoC guarantees precise control over the clocking architecture, and provides a wide range of applications with diverse clocking needs. Fig. 5 illustrates the clocking resources in the Zynq-7000. The yellow blocks represent the main PLLs used to generate the primary clocks of the SoC, while the blue blocks represent the reconfigurable PLLs, which primarily consist of PLLs and mixed-mode clock managers (MMCMs).

With the Zynq-7000 SoC-FPGA, the construction of the covert channel does not involve the use of the silicon PLLs (yellow blocks in Fig. 5), because their modification requires root privileges when the chip is protected by TrustZone. Instead, the reconfigurable clocking resources, mainly the MMCM and PLL of the programmable logic (see Fig. 5), are used. These two features enable the generation of a desired frequency without root privileges and also allow dynamic changes to the clock frequency. This means that users can modify the frequency at runtime. To implement the covert channel, the attacker will modify the clock output frequency of MMCM during each call to the targeted encryption algorithm. To perform the acquisition, the frequency change is detected, and based on the detected change, a trigger is generated to control the operation of the sensor.

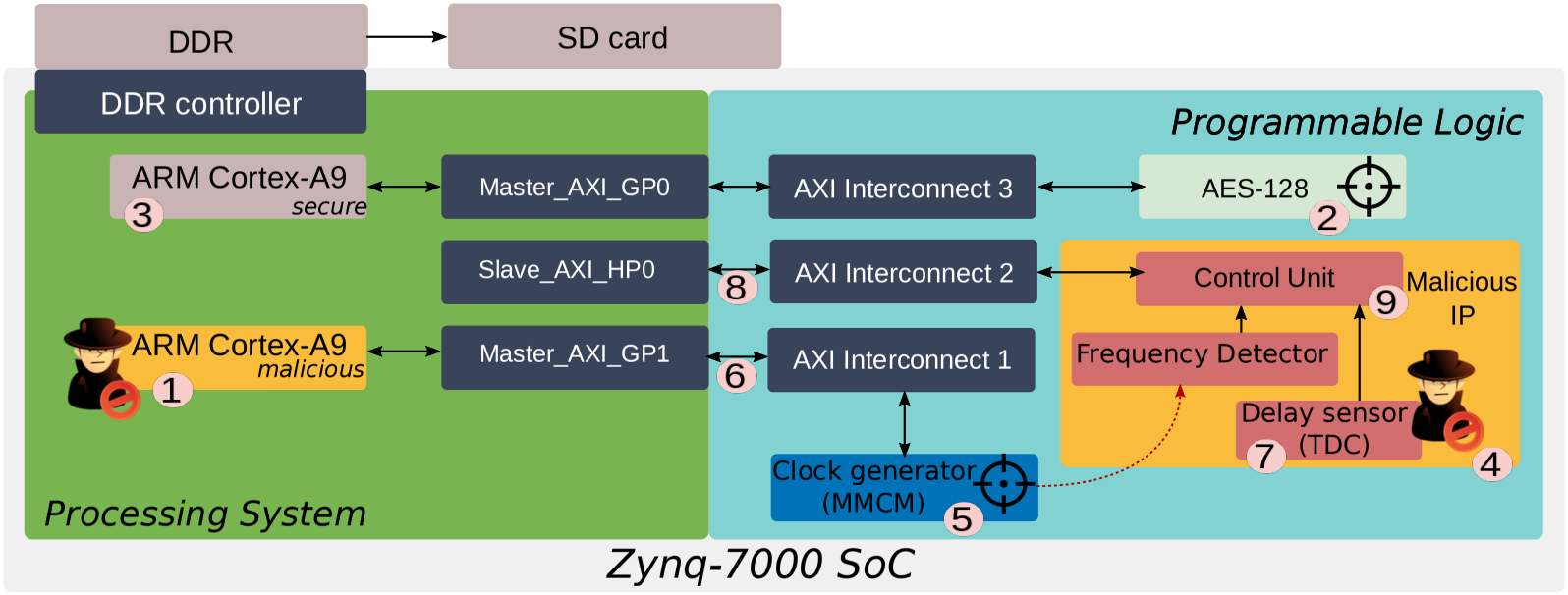


Figure 7: The architecture created to emulate RPA attacks. The hardware modules implemented in the programmable logic of the Zynq-7000 are controlled by the processing system using AXI4-Lite interfaces. The samples are stored in DDR and finally SD.

3.5 Target hardware architecture

The AES is a standard algorithm used to provide confidentiality and authentication to data. It is composed of a substitution-permutation network with a data input of 128-bits and keys of 128, 192, or 256 bits. The data are processed using its internal permutation or *round function* in 10, 12, or 14 rounds, which correspond to the key length. The focus of this work is on the 128-bit variant, as it is the most commonly used.

The internal permutation of AES is composed of four transformations that confer diffusion and confusion to the data, see Fig. 6. These functions consist of substitutions, permutations, and finite-field operations with a respective invertible equivalent used in the decryption process. The `KeyExpansion` process guarantees that each round uses a different subkey derived from the initial encryption key. This process involves taking the original key and expanding it into a set of round keys.

The basic approach used to implement AES is performing an encryption round for each cycle, that is, applying the four transformations of the round function consecutively and then recording the intermediate result. The same AES implementation as in [Gra+20] is used as a case study. It includes the forward and inverse transformations of the cipher. Only the encryption functionality is used to perform the attack. The different blocks in the architecture can be disabled to streamline the datapath, for example, in the initial round only the `AddRoundKey` operation is active and in the last round the `MixColumns` block is disabled. The `KeyExpansion` is resolved when the system is initialized and maintains the sub-keys until the process is completed.

In our hardware implementation, the `KeyExpansion` has a latency of two cycles, then an initial round, nine regular rounds, and a final round, for a total latency of 13 cycles per block. When operating at 10MHz, it consumes approximately 18% of the LUTs and 6% of the FF in the Zynq (xc7z010clg400-1).

Since AES has a fixed-size input (128-bits), a partitioning strategy must be used to divide longer messages into blocks. For the usual power analysis techniques [KJJ99; BCO04] a large number of known inputs and their respective outputs are required. If the AES accelerator is configured in *electronic code book* mode, which is unusual, then a large number of queries must be performed, requiring a significant input set. However, by configuring the accelerator in *code-block chain* mode only the initial plaintext needs to be provided, after which the output of the cipher can be observed.

3.6 Experimental setup

An architecture to demonstrate the viability of the proposed attack was implemented on an AMD-Xilinx Zynq 7000 heterogeneous SoC-FPGA, see Fig. 7. The main difference between this approach and a practical case study is that the data are not transferred remotely. Rather, an acquisition system is implemented to reduce the evaluation time. The malicious application implemented in one of the ARM cores of the SoC (labeled “1” in Fig. 7) can query the AES hardware accelerator (labeled “2”) controlled by the second core (labeled “3”), considered secure. The malicious process controls a third-party IP (labeled “4”)

which is dynamic clock generator IP from Digilent.[1] The MMCM primitive in the Zynq fabric (labeled "5") is controlled using a standalone driver included in this IP. Control is achieved through an AXI Lite interface from the PS (labeled "6"), which allows different frequencies to be generated at runtime. The main characteristic of this MMCM-based clock generator is that it will continue operating regardless of what happens to the main PLLs of the SoC. To incorporate the delay sensors (labeled "7") into the system, the TDC-based sensor and RO-based sensor IP cores obtained from a publicly available git repository were used.[2] These cores were configured through the processing system using the AXI4-Lite interface (labeled "8"). The sampling rate of these sensors was set at 200 MHz. The fabric also includes a control unit (labeled "9") that is responsible for managing the storage of data in the DDR. To access and extract the collected data, the contents of the DDR memory are transferred to an SD card. Transferring the data to an external storage medium enables flexible access for offline processing. The AES accelerator operates at a frequency of 10 MHz.

4 Experimental results

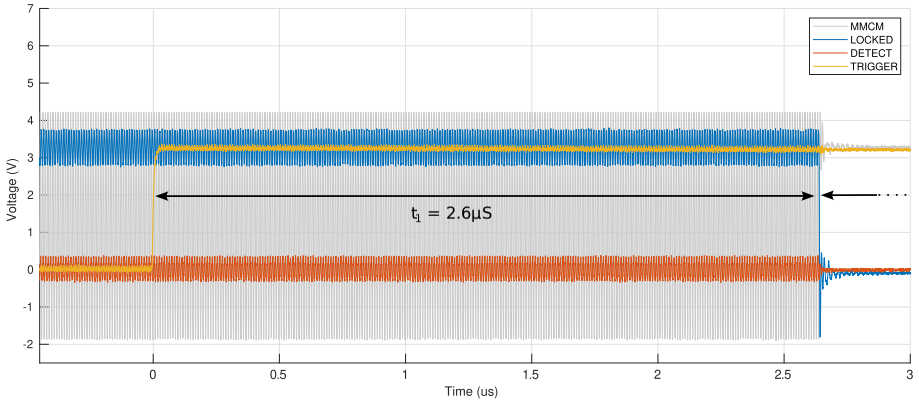
We used a Digilent Zybo Z-7010 development board in the experiment. This platform features a Zynq-7000 SoC-FPGA (xc7z010clg400-1). We used a AMD-Xilinx 2022.1 toolchain, created the hardware specification in Vivado and programmed and launched the applications through Vitis.

4.1 Step response of the MMCM

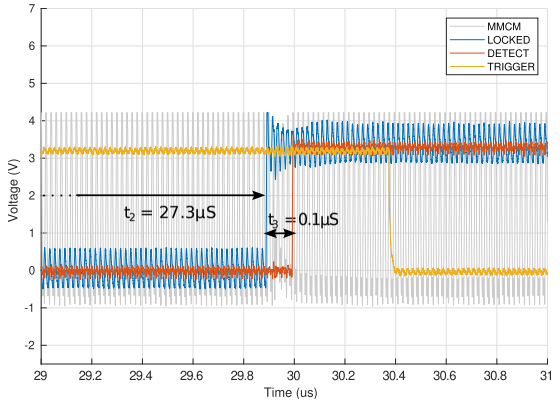
The response times of the MMCM were initially investigated to evaluate the feasibility of modifying the frequency of these components in the processing system. The experimental results are presented in Fig. 8. The objective was to compare the response time when a decrease in the output frequency is requested with the response time when an increase is requested. Initially, a digital trigger was activated from the processor, followed by a request to change the frequency from 100 MHz to 50 MHz. It was estimated that the MMCM requires approximately $t_1 = 2.6$ μs to initiate the process of modifying its output, as depicted in Fig. 8a. Subsequently, an additional duration of $t_2 = 27.3$ μs was necessary to execute the requested frequency change. During this interval, the output of the MMCM is unstable, meaning it is not producing a consistent and reliable output. This instability is a natural consequence of the reconfiguration process occurring within the MMCM. Once the frequency modification was completed, it took approximately $t_3 = 0.1$ μs for the fabric-based frequency detector to detect this change, as illustrated in Fig. 8b. The entire process of detecting a decreasing frequency change took approximately 30 μs.

Conversely, when the processor requested a frequency change from 50 MHz to 100 MHz, it was estimated that approximately $t_4 = 2.3$ μs were needed to initiate the output modification process, see Fig. 8c, followed by an additional $t_5 = 26.6$ μs to perform the requested change. In this case, the frequency detector required 0.01 μs to detect the increasing output frequency of the MMCM, causing the

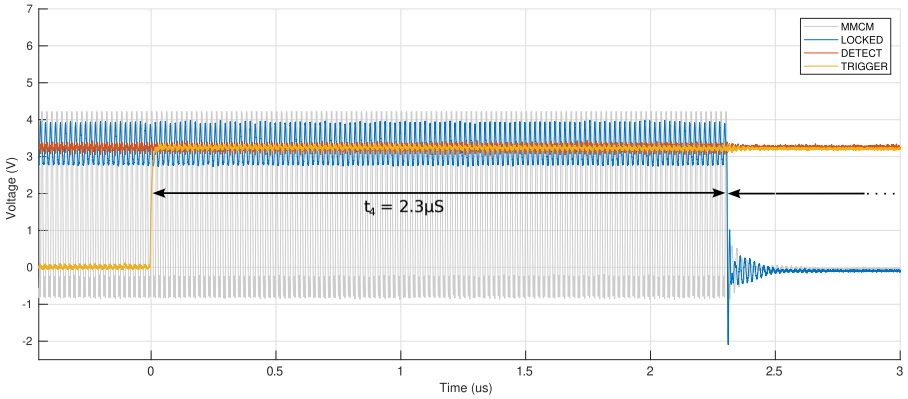
[1] `https://github.com/Digilent/vivado-library`
[2] `https://github.com/emse-sas-lab/SCAbox-ip/`



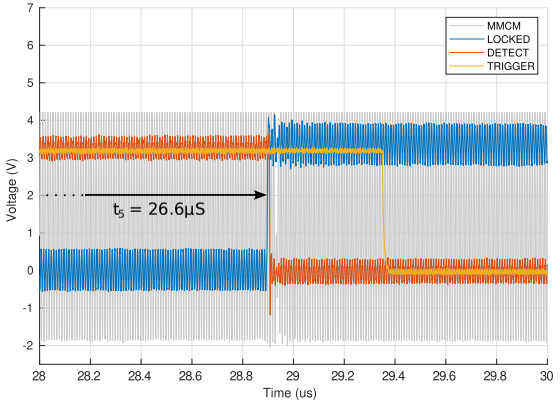
(a) Pull trigger. The PS requests the MMCM to decrease its output frequency (100MHz to 50MHz).



(b) Start of capture. The MMCM frequency stabilizes.



(c) Release trigger. The PS requests the MMCM to restore the previous frequency.



(d) End of capture. The MMCM frequency stabilizes.

Figure 8: The step response of a MMCM in the Zynq-7000 SoC-FPGAs.

Algorithm 1 The channel modulation strategy

Require: f_1, f_2 a pair of frequencies
  $f_{MMCM} = f_1$
  while TRUE **do**
    $f_{MMCM} \leftarrow f_{MMCM} = f_1? \quad f_2 : f_1$
    `AES`(ENCRYPT)
  end while

detect signal to fall, as depicted in Fig. 8d. In all, 28.91 μs were necessary to detect a rising frequency change.

Based on this experiment, it can be inferred that the MMCM responds faster when an increase in frequency is requested compared to a decrease. It is worth noting that since the MMCM calls are blocked and positioned before the encryption calls, there is no need to account for the transition delay during the acquisition process.

4.2 The covert channel

In the proposed approach, the algorithm in Fig. 1 was used to encode the synchronization information into the covert channel. Before each call to the cipher, the malicious application modified the output frequency of the MMCM. This value would iterate between two predefined values. This frequency manipulation guarantees the encoded information remains concealed but is still discernible to the data acquisition system. To detect this frequency modulation, the pulse duration of the clock signal was measured. Depending on the pulse length, a write enable signal was generated to control the data transfer from the sensor to memory.

4.3 Validation and analysis

To assess the feasibility of the proposed approach, the acquisition of 200k traces from the AES encryption of 200k different plaintexts was performed using a TDC-based sensor. The sensor was situated 20 horizontal slices from the AES, constituting a far setup (see Fig. 9). Each of the slices contains 4 lookup tables, 3 multiplexers, 8 flip-flops, and 1 carry logic unit. Previous works have highlighted the significant impact of calibration and sensor proximity to the victim block on the number of traces required to infer the AES key. It has been observed that when the sensor is positioned near the victim block and is properly calibrated, the number of traces needed for successful key inference is significantly reduced [Gra+20; Sch+21].

Fig. 10a depicts a power trace that corresponds to the AES encryption operation. At first sight it is possible to identify the sequence of samples corresponding to the execution of AES. These samples demonstrate an observable increase in power consumption, which reflects the increased power requirements associated with the process of encryption. Despite the length of the traces affecting the execution time for performing an attack, this should not have an impact on the correlation analysis. Nonetheless, lengthier traces imply that more data must be transmitted.

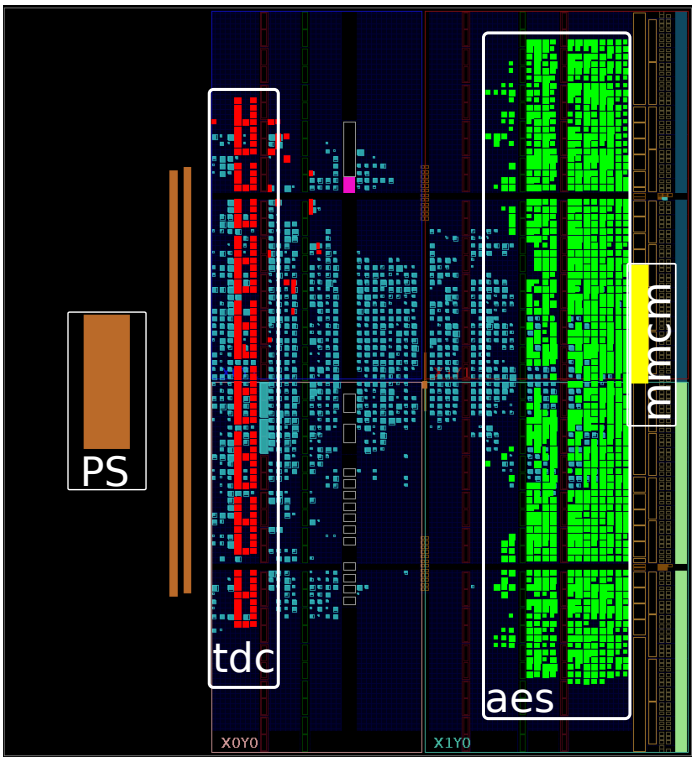
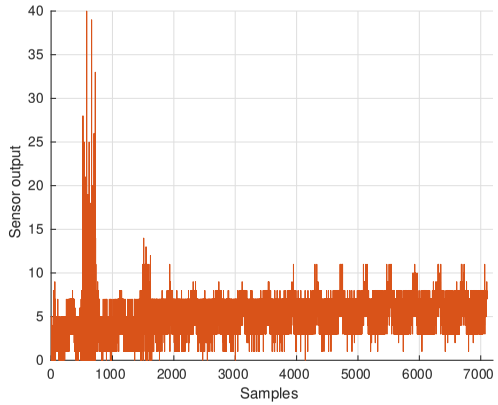


Figure 9: Sensor placement in “far” setup

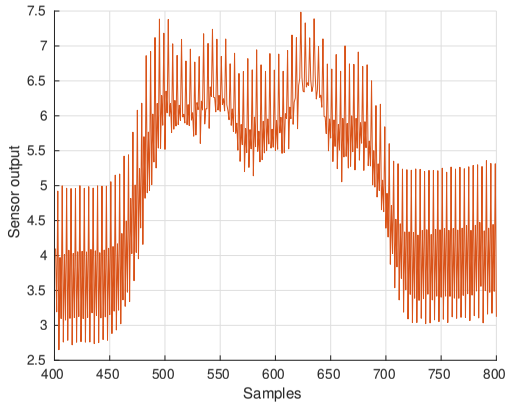
On the other hand, misalignment of power traces can impede statistical attacks such as CPA.

Ideally, the transition delay of the MMCM would be constant, simplifying the alignment process. However, in reality, slight variations in transition delay can occur due to factors such as circuit delay, heating, and noise, which can cause jitter. This results in slightly different phase shifts in the generated clocks. As mentioned above, the generation of the enable signal relies on the pulse duration of the MMCM clock, giving rise to slight shifts in the generated trigger, and thus leading to minor miss-alignments between consecutive measurements, typically on the order of a few clock cycles. Fig. 10b demonstrates the effect of this misalignment, showing the loss of leakage information and the inability to distinguish the 11 rounds of AES when averaging 200k unaligned traces, which means that the traces are not well aligned.

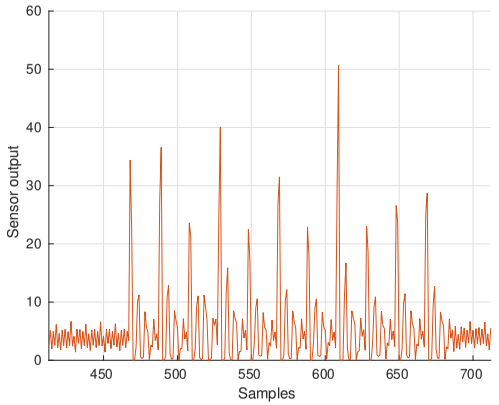
Consequently, alignment of power traces is imperative before launching the CPA. To achieve the alignment of power traces, we used a cross-correlation technique, but prior to this, a noise filtering process was required. Given that the trigger is activated before AES encryption, some circuit noise unrelated to the encryption activity was acquired. To mitigate this noise, we performed a frequency analysis of the sensor output to identify the primary frequencies of the power trace, as illustrated in Fig. 12. The analysis revealed a dominant spike in the frequency spectrum at the target frequency $f_t = 10$ MHz, which corresponds to the operating frequency of the AES. A band-pass filter was in-



(a) A single trace

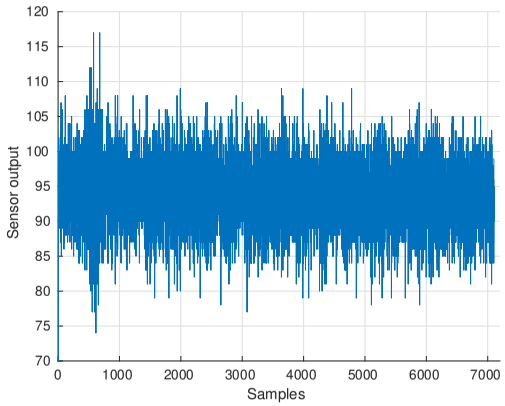


(b) Mean of 200k traces, unaligned

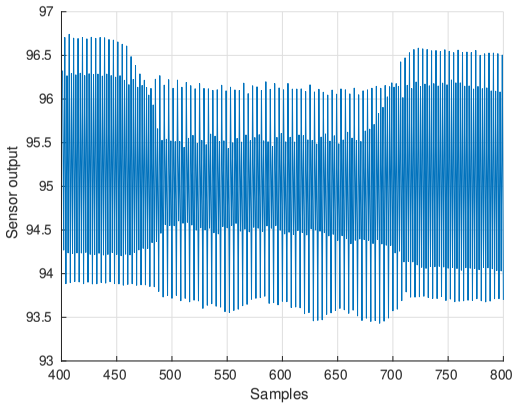


(c) Mean of 200k traces, aligned

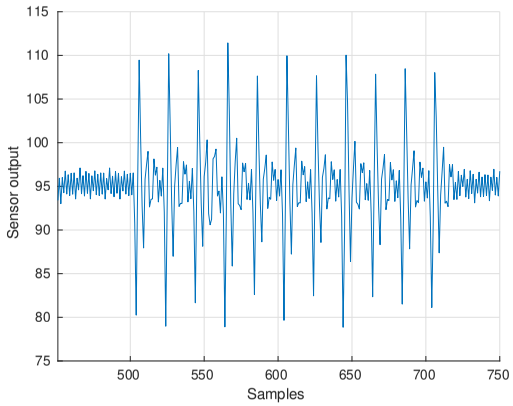
Figure 10: Traces obtained with a TDC-based sensor



(a) A single trace



(b) Mean of 200k traces, unaligned



(c) Mean of 200k traces, aligned

Figure 11: Traces obtained with a RO-based sensor

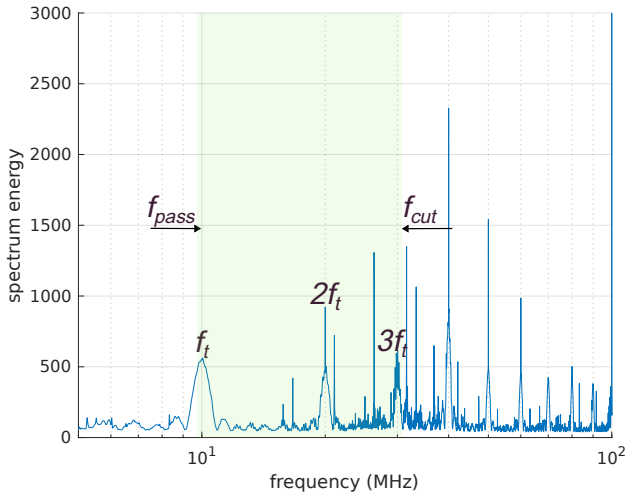


Figure 12: Frequency analysis of the sensor output

tentionally designed in response to this analysis. This filter allows not only the target frequency itself but also its first two harmonics ($2f_t$ and $3f_t$) to pass through. These harmonics correspond to 20 MHz and 30 MHz, respectively. Its operational range spans from a lower pass frequency of $f_{\text{pass}} = 9$ MHz to an upper cut frequency of $f_{\text{cut}} = 31$ MHz. This tailored filtering process is implemented to effectively reduce unwanted noise and interference.

To confirm the effectiveness of the proposed alignment method, the average of 200k aligned traces was calculated, as depicted in Fig. 10c. The 11 rounds of the AES algorithm are clearly visible, indicating successful alignment of the traces. Finally, a CPA attack was performed on the aligned trace set to check the possibility of retrieving the secret key of the targeted AES. We chose to attack the last AES round [BCO04]. As shown in Fig. 13, all 16 key bytes were successfully revealed. In all cases, the accurate guess reliably emerged from the group of hypotheses with a clear advantage over the second guess, requiring fewer than 10k traces in all.

A similar analysis was performed on the RO-based sensor. The trace obtained using this sensor is illustrated in Fig. 11a and shows that the noise present in the trace obtained with the RO-based sensor is considerably more significant than that of the TDC-based sensor. Therefore, the resolution of the RO-based sensor is inferior to that of the TDC-based sensor. The misalignment of the traces, which is not linked to the type of sensor but with the clock generator (MMCM), required repetition of the alignment procedure. Fig. 11c depicts the average of 200k aligned traces. The 11 rounds of AES are clearly illustrated. This shows that the traces are properly aligned and ready to be used to conduct the CPA attack. Fig. 14 presents the results of the CPA attack on 16 bytes of AES. All 16 bytes of the key were successfully revealed, and with almost 60k traces, the correct guesses significantly outperformed all the other possibilities.

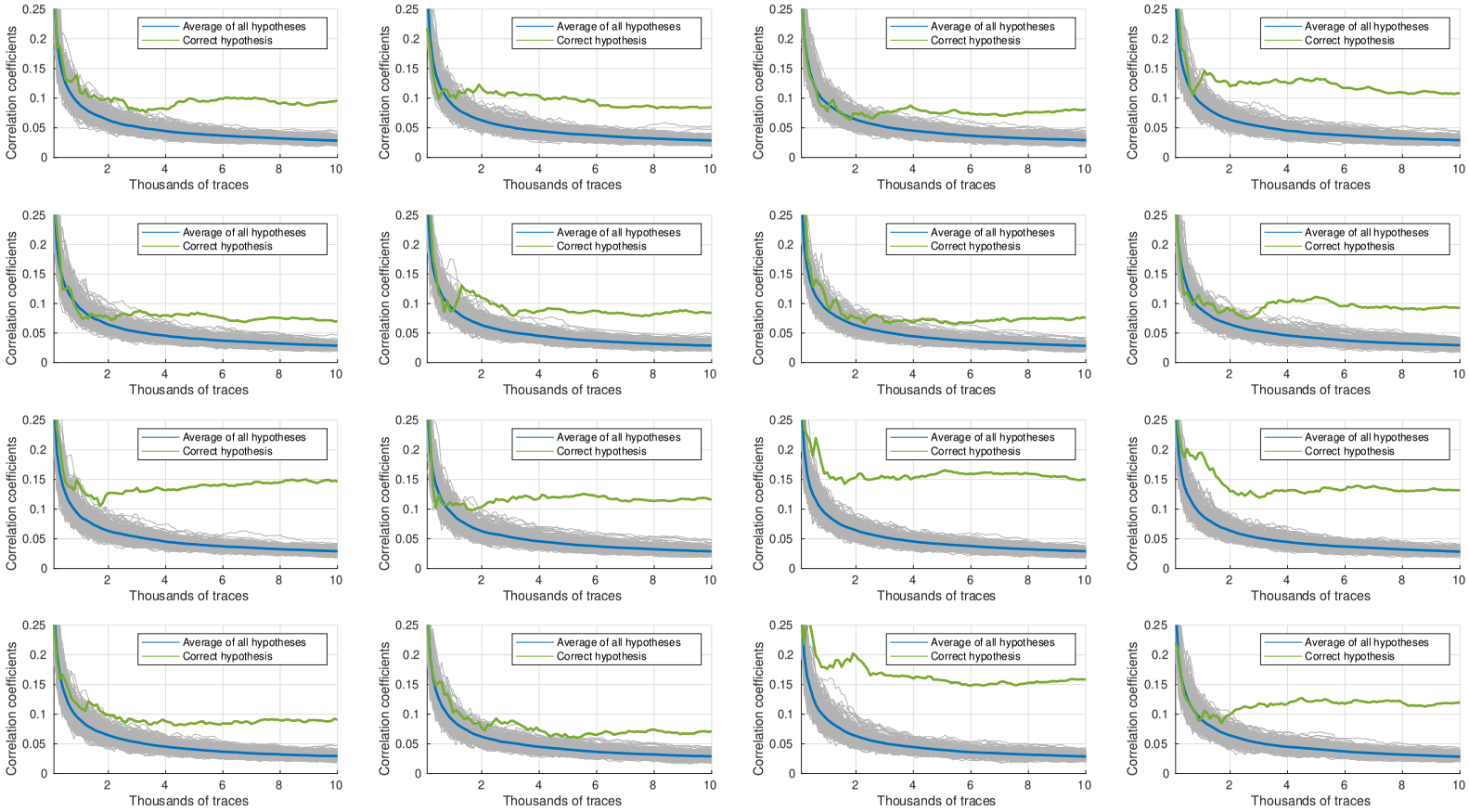


Figure 13: CPA on the 16 bytes of the AES-128 key, using traces from a TDC-based sensor

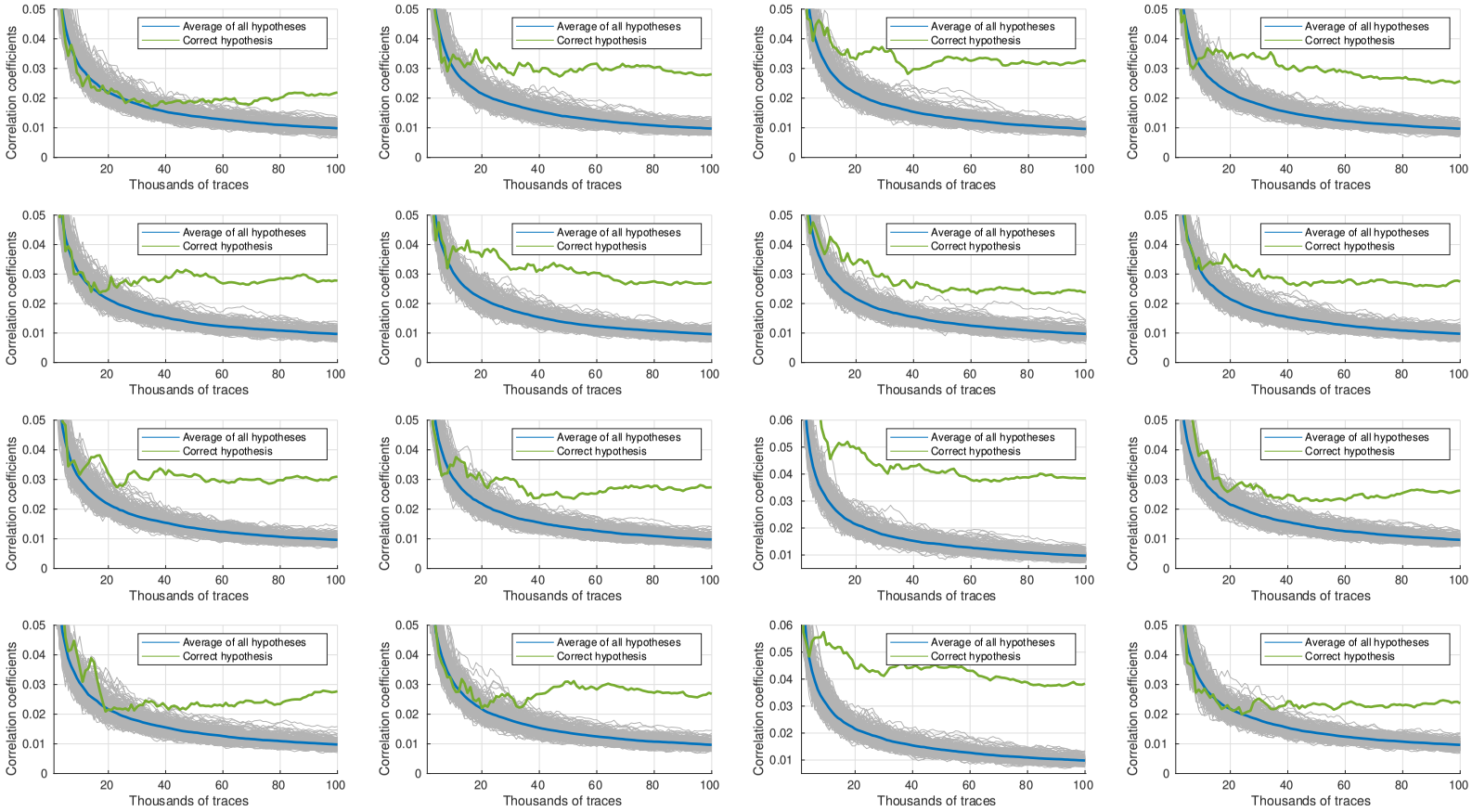


Figure 14: CPA on the 16 bytes of the AES-128 key, using traces from a RO-based sensor

5 Discussion

In the previous sections, we demonstrated that by allowing untrusted users to remotely deploy a power monitor circuit on a heterogeneous SoC device, it is possible to discern the level of dynamic switching activities occurring within the circuit. By leveraging this power monitoring capability, a range of security attacks can potentially be exploited. In [KBG09], the authors proved that the added power noise does not affect the result of the power attack if the noise does not depend on the input data of the AES module. As a result, combining the sensor with a second module to obfuscate its intent will not affect the feasibility of the attack provided the activity of the added module is independent of the encrypted data. This characteristic renders the sensor considerably more challenging to detect or identify. Moreover, when combined with a covert channel for the purpose of synchronization, their impact becomes even more significant and challenging to mitigate. As a consequence, conducting a comprehensive analysis of the system is imperative to address these emerging threats effectively.

Countermeasures to mitigate the risk of remote hardware attacks can be categorized in two main approaches. The first category focuses on enhancing the resilience of cryptographic algorithms against side-channel attacks. This involves the use of masking or hiding techniques to protect sensitive information. Masking techniques involve transforming the implementation of a cryptographic algorithm into an alternative form that remains functionally equivalent and secure [PR13; Sim+23a; Sim+23b]. By doing so, the power consumption patterns, which can be exploited by attackers to infer sensitive information, are concealed. This approach adds a layer of protection to the algorithm's execution, making it more resistant to side-channel attacks. Hiding techniques on the other hand aim to reduce the signal-to-noise ratio in the system, making it more challenging for attackers to extract meaningful information from side-channel leakages. One example of a hiding technique is the insertion of a grid of ring oscillators between the victim and the attacker [Kra+19]. These ring oscillators are randomly activated, introducing additional sources of noise that obscure the power consumption patterns. By increasing the number of required measurements, the attacker's task becomes significantly more difficult.

The second category of countermeasures focuses on preventing the implementation of attacks within the circuit itself. In the context of multi-tenant FPGAs, the objective here is to guarantee that the bitstreams used to configure the FPGA do not contain malicious blocks. One way to achieve this is detecting LUT-based random oscillator designs within the configuration bitstream [Gna+18]. FPGA vendor tools, such as AMD-Xilinx Vivado, already produce warning signals when combinatorial loops are detected during the bitstream generation process. However, [Gna+18] goes further and successfully identifies suspicious timing violations that may indicate the presence of malicious TDC-based sensor circuitry, even when combinatorial loops are not detected. The identification of such violations can help prevent the deployment of configurations containing potential hardware-based attacks. Another tool, FPGADefender, has been developed specifically to address this issue [La+20]. It functions in a similar way to a software virus scanner but specifically targets FPGA configurations for the detection of malicious constructs. The main contribution of this tool is its ability to identify the different types of self-oscillators

and short circuits documented in the literature. However, one limitation of FPGADefender is that it cannot differentiate between the use of ring oscillators for legitimate security purposes and their exploitation for malicious intent. As a result, it may produce false alarms when it detects constructs like true random number generators, which may be used for secure applications.

6 Conclusions

This paper presents an innovative and robust strategy for the automatic alignment of traces in the domain of RPA attacks. The challenge of aligning and synchronizing traces in an automated manner is addressed. The effectiveness of the approach is evaluated through a case study involving correlation power analysis on an unprotected implementation of AES, a widely used encryption algorithm. The entire secret key was successfully recovered by applying the proposed methodology, demonstrating the feasibility and effectiveness of the strategy. The attack scenario considered in this work assumes the capability to manipulate the clock management system of the FPGA and to strategically deploy TDC-based and RO-based sensors within the fabric. By making these assumptions, we create a realistic and sophisticated attack environment where vulnerabilities in the hardware components can be exploited to gain unauthorized access to sensitive information. The process of conducting this study and achieving successful results, provides valuable insights into the potential risks associated with RPA attacks.

References

[Agr+03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao and Pankaj Rohatgi. "The EM Side—Channel(s)". In: *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Berlin, Heidelberg: Springer, 2003, pp. 29–45. ISBN: 978-3-540-36400-9. DOI: 10.1007/3-540-36400-5_4.

[Ala+17] Murugappan Alagappan, Jeyavijayan Rajendran, Miloš Doroslovački and Guru Venkataramani. "DFS covert channels on multi-core platforms". In: *Proceedings of the 2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. IEEE, 2017, pp. 1–6. DOI: 10.1109/VLSI-SoC.2017.8203469.

[Asi+17] Mikhail Asiatici, Nithin George, Kizheppatt Vipin, Suhaib A. Fahmy and Paolo Ienne. "Virtualized Execution Runtime for FPGA Accelerators in the Cloud". In: *IEEE Access* 5 (2017), pp. 1900–1910. DOI: 10.1109/ACCESS.2017.2661582.

[BB18] El Mehdi Benhani and Lilian Bossuet. "DVFS as a Security Failure of TrustZone-enabled Heterogeneous SoC". In: *Proceedings of the 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*. IEEE, 2018, pp. 489–492. DOI: 10.1109/ICECS.2018.8618038.

[BB21] Lilian Bossuet and El Mehdi Benhani. “Performing Cache Timing Attacks from the Reconfigurable Part of a Heterogeneous SoC—An Experimental Study”. In: *Applied Sciences* 11.14 (2021). ISSN: 2076-3417. DOI: 10.3390/app11146662.

[BCO04] Eric Brier, Christophe Clavier and Francis Olivier. “Correlation Power Analysis with a Leakage Model”. In: *Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Berlin, Heidelberg: Springer, 2004, pp. 16–29. ISBN: 978-3-540-28632-5. DOI: 10.1007/978-3-540-28632-5_2.

[Fra+10] John J. León Franco, Eduardo Boemo, Encarnación Castillo and Luis Parrilla. “Ring oscillators as thermal sensors in FPGAs: Experiments in low voltage”. In: *Proceddings of the 6th Southern Programmable Logic Conference (SPL)*. IEEE, 2010, pp. 133–137. DOI: 10.1109/SPL.2010.5483027.

[FVS15] Suhaib A Fahmy, Kizheppatt Vipin and Shanker Shreejith. “Virtualized FPGA Accelerators for Efficient Cloud Computing”. In: *Proceedings of the 7th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. 2015, pp. 430–435. DOI: 10.1109/CloudCom.2015.60.

[GER19] Ilias Giechaskiel, Ken Eguro and Kasper B. Rasmussen. “Leakier Wires: Exploiting FPGA Long Wires for Covert- and Side-Channel Attacks”. In: *ACM Transactions on Reconfigurable Technolies and Systems* 12.3 (2019). ISSN: 1936-7406. DOI: 10.1145/3322483.

[Gla+20] Ognjen Glamočanin, Louis Coulon, Francesco Regazzoni and Mirjana Stojilović. “Are Cloud FPGAs Really Vulnerable to Power Analysis Attacks?” In: *Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 1007–1010. DOI: 10.23919/DATE48585.2020.9116481.

[Gna+18] 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: *Proceedings of the 2018 International Conference on Field-Programmable Technology (FPT)*. IEEE, 2018, pp. 286–289. DOI: 10.1109/FPT.2018.00055.

[Gna+21] Dennis R. E. Gnad, Cong Dang Khoa Nguyen, Syed Hashim Gillani and Mehdi B. Tahoori. “Voltage-Based Covert Channels Using FPGAs”. In: *ACM Transactions on Design Automation of Electronic Systems* 26.6 (2021). ISSN: 1084-4309. DOI: 10.1145/3460229.

[Gra+19] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia and Philippe Loubet-Moundi. “High-Speed Ring Oscillator based Sensors for Remote Side-Channel Attacks on FPGAs”. In: *Proceedings of the 2019 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. IEEE, 2019, pp. 1–8. DOI: 10.1109/ReConFig48160.2019.8994789.

[Gra+20] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia, Philippe Loubet Moundi and Francis Olivier. "Remote Side-Channel Attacks on Heterogeneous SoC". In: *Proceedings of the 18th International Conference on Smart Card Research and Advanced Applications (CARDIS)*. Springer, 2020, pp. 109–125. ISBN: 978-3-030-42068-0. DOI: 10.1007/978-3-030-42068-0_7.

[Gra+21] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia and Philippe Loubet Moundi. "SideLine: How Delay-Lines (May) Leak Secrets from Your SoC". In: *Proceedings of the 12th International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*. Springer, 2021, pp. 3–30. DOI: 10.1007/978-3-030-89915-8_1.

[Hen10] Stephan Henzler. "Time-to-Digital Converter Basics". In: *Time-to-Digital Converters*. Dordrecht: Springer, 2010, pp. 5–18. ISBN: 978-90-481-8628-0. DOI: 10.1007/978-90-481-8628-0_2.

[HS15] Tzipora Halevi and Nitesh Saxena. "Keyboard acoustic side channel attacks: exploring realistic and security-sensitive scenarios". In: *International Journal of Information Security* 14 (2015), pp. 443–456. DOI: 10.1007/s10207-014-0264-7.

[KBG09] Najeh Kamoun, Lilian Bossuet and Adel Ghazel. "Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher". In: *Proceedings of the 3rd International Conference on Signals, Circuits and Systems (SCS)*. IEEE, 2009, pp. 1–6. DOI: 10.1109/ICSCS.2009.5412604.

[KGT20] Jonas Krautter, Dennis Gnad and Mehdi Tahoori. "CPAmap: On the Complexity of Secure FPGA Virtualization, Multi-Tenancy, and Physical Design". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020.3 (2020), pp. 121–146. DOI: 10.13154/tches.v2020.i3.121-146.

[KJJ99] Paul Kocher, Joshua Jaffe and Benjamin Jun. "Differential Power Analysis". In: *Proceedings of the 19th Annual International Cryptology Conference (CRYPTO)*. Berlin, Heidelberg: Springer, 1999, pp. 388–397. ISBN: 978-3-540-48405-9. DOI: 10.1007/3-540-48405-1_25.

[Koc96] Paul Kocher. "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems". In: *Proceedings of the 16th Annual International Cryptology Conference (CRYPTO)*. Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9.

[Kra+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". In: *Proceedings of the 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–8. DOI: 10.1109/ICCAD45719.2019.8942094.

[La+20] Tuan Minh La, Kaspar Matas, Nikola Grunchevski, Khoa Dang Pham and Dirk Koch. "FPGADefender: Malicious Self-Oscillator Scanning for Xilinx UltraScale + FPGAs". In: *ACM Transactions on Reconfigurable Technolgies and Systems* 13.3 (2020). ISSN: 1936-7406. DOI: 10.1145/3402937.

[LHZ21] Miriam Leeser, Suranga Handagala and Michael Zink. "FPGAs in the Cloud". In: *Computing in Science & Engineering* 23.6 (2021), pp. 72–76. DOI: 10.1109/MCSE.2021.3127288.

[Mas+15] Ramya Jayaram Masti et al. "Thermal Covert Channels on Multi-core Platforms". In: *Proceedings of the 24th USENIX Security Symposium*. Washington D.C., USA: USENIX Association, 2015, pp. 865–880. ISBN: 978-1-939-13311-3.

[PR13] Emmanuel Prouff and Matthieu Rivain. "Masking against Side-Channel Attacks: A Formal Security Proof". In: *Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Springer, 2013, pp. 142–159. DOI: 10.1007/978-3-642-38348-9_9.

[Qas+19] Murad Qasaimeh et al. "Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels". In: *Proceedings of the 2019 IEEE International Conference on Embedded Software and Systems (ICESS)*. 2019, pp. 1–8. DOI: 10.1109/ICESS.2019.8782524.

[Ram+18] Chethan Ramesh et al. "FPGA Side Channel Attacks without Physical Access". In: *Proceedings of the 26th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2018, pp. 45–52. DOI: 10.1109/FCCM.2018.00016.

[Sch+18] 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 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2018, pp. 1–7. DOI: 10.1145/3240765.3240841.

[Sch+21] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi and Mehdi B. Tahoori. "An Inside Job: Remote Power Analysis Attacks on FPGAs". In: *IEEE Design & Test* 38.3 (2021), pp. 58–66. DOI: 10.1109/MDAT.2021.3063306.

[Sim+23a] Mateus Simões et al. "Low-Latency Masking with Arbitrary Protection Order Based on Click Elements". In: *Proceedings of the 2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2023, pp. 36–47. DOI: 10.1109/HOST55118.2023.10133813.

[Sim+23b] Mateus Simões et al. "Self-timed Masking: Implementing Masked S-Boxes Without Registers". In: *Proceedings of the 21st International Conference on Smart Card Research and Advanced Applications*. Cham: Springer International Publishing, 2023, pp. 146–164. ISBN: 978-3-031-25319-5. DOI: 10.1007/978-3-031-25319-5_8.

[SMS20] Zeinab Seifoori, Seyedeh Sharareh Mirzargar and Mirjana Stojilović. "Closing Leaks: Routing Against Crosstalk Side-Channel Attacks". In: *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*. New York NY, USA: Association for Computing Machinery, 2020, pp. 197–203. ISBN: 978-1-450-37099-8. DOI: 10.1145/3373087.3375319.

[TS19] Shanquan Tian and Jakub Szefer. "Temporal Thermal Covert Channels in Cloud FPGAs". In: *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 298–303. ISBN: 978-1-450-36137-8. DOI: 10.1145/3289602.3293920.

[TSS17] Adrian Tang, Simha Sethumadhavan and Salvatore Stolfo. "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management". In: *Proceedings of the 26th USENIX Security Symposium*. USENIX Association, 2017, pp. 1057–1074. ISBN: 978-1-931971-40-9.

[YB17] Sadegh Yazdanshenas and Vaughn Betz. "Quantifying and mitigating the costs of FPGA virtualization". In: *Proceedings of the 27th International Conference on Field Programmable Logic and Applications (FPL)*. 2017, pp. 1–7. DOI: 10.23919/FPL.2017.8056807.

[ZBT10] Daniel Ziener, Florian Baueregger and Jürgen Teich. "Using the Power Side Channel of FPGAs for Communication". In: *Proceedings of the 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2010, pp. 237–244. DOI: 10.1109/FCCM.2010.43.

[Zic+13] 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)*. FPGA '13. Monterey, California, USA: Association for Computing Machinery, 2013, pp. 101–104. ISBN: 978-1-450-31887-7. DOI: 10.1145/2435264.2435283.

[ZMZ17] Elnaz Zafarkhah, Mohammad Maymandi-Nejad and Maryam Zare. "Improved accuracy equation for propagation delay of a CMOS inverter in a single ended ring oscillator". In: *AEU - International Journal of Electronics and Communications* 71 (2017), pp. 110–117. ISSN: 1434-8411. DOI: 10.1016/j.aeue.2016.10.009.

[ZS18] Mark Zhao and G. Edward Suh. "FPGA-Based Remote Power Side-Channel Attacks". In: *Proceedings of the 2018 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2018, pp. 229–244. DOI: 10.1109/SP.2018.00049.