A Lightweight Non-Oscillatory Delay-Sensor for Remote Power Analysis

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

A Lightweight Non-Oscillatory Delay-Sensor for Remote Power Analysis

Anis FELLAH-TOUTA, Lilian BOSSUET, and Carlos Andres LARA-NINO
Université Jean Monnet Saint-Étienne, CNRS, Institut d'Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023,
SAINT-ETIENNE, France.
anis.fellah.touta@univ-st-etienne.fr

Abstract

Traditionally, there have been two main obstacles for practical power analysis attacks: the adversary needed physical access to the device, and they had to use sophisticated sensing equipment to obtain the samples. However, it is now known that an attacker may leverage remote access to the platform and internal sensors to perform power analysis attacks. Internal sensors are circuits created from components native to the device, for example the reconfigurable fabric in some heterogeneous SoCs. Now, the main drawbacks of these sensors are their large sizes and that they require precise placement to improve their fidelity. This facilitates their detection. In this paper, we describe a novel internal sensor created from an 8-bit multiplier. This circuit can be implemented with just two LUT6 and four CARRY4 in AMD-Xilinx FPGAs. It can produce up to 200 MSpS. Furthermore, no precise placement nor special hardware description are required in its implementation. To validate our claims, we have recovered the full key of an unprotected implementation of AES-128 clocked at 100 MHz with under 3e4 encryption traces.

Keywords: Internal Delay Sensor, Multiplier, Remote Power Analysis.

Please cite as:

```
@InProceedings{FBL24,
title = {{A Lightweight Non-Oscillatory Delay-Sensor for Remote Power Analysis}},
author = {Fellah-Touta, Anis and Bossuet, Lilian and Lara-Nino, Carlos Andres},
booktitle = {Proceedings of the 2024 IEEE International Symposium on Hardware Oriented
Security and Trust (HOST)},
pages = {343--348},
year = {2024},
publisher = {IEEE},
location = {Tysons Corner VA, USA},
doi = {10.1109/HOST55342.2024.10545353},
isbn = {979-8-3503-7394-3}}
```

1 Introduction

The integration of FPGAs into modern computing platforms represents a significant advance. Designers can now leveraging FPGA's reconfigurability to enhance computational efficiency in high performance applications. These devices, recognized for their parallelism and adaptability, introduce a dynamic element to computing infrastructures. FPGAs can provide rapid and dynamic customization. This integration promotes scalability, adaptability, and enhanced performance for critical applications. Along with these factors, the monetary cost per logic-cell of reconfigurable fabric has decreased over time, which makes them attractive for bulk acceleration [Kha+18].

FPGAs can be found in multiple edge applications. The greater progress in their adoption of FPGAs is found on SoC-FPGAs, datacenter acceleration cards, and cloud services providers. Under all these systems the designers have sought to separate the FPGA from the rest of the system through different isolation strategies. This is to handle technology differences, but also to provide some logical protection against misuse. Despite these measures, however, FPGAs are frequently targeted in works which seek to compromise the security of the system remotely [Sch+21; ZS18].

A critical weakness of systems with FPGAs arises from the shared power distribution networks (PDN). The PDN is responsible for supplying power to different components of the platform. Since a PDN stretches over the whole system and bypasses logical isolation policies, it is a latent channel for information leakage. Power fluctuations caused by data processing can be sensed across the platform thanks to the shared PDN. Through careful analysis of these variations, an adversary may obtain sensitive information from the device. This situation poses a significant risk to data privacy, even when physical and logical isolation measures are in place.

The vulnerability of FPGA systems against remote power attacks arises from the instantiation of internal sensors in the reconfigurable fabric. The most prominent of these circuits include time-to-digital converters (TDC) [Sch+21] and ring oscillators [ZS18]. The former, measure the propagation time of a clock signal traversing through hardware elements, which provides precise insights into voltage changes. The latter, generate an oscillatory signal which is sensitive to fluctuations in the power supply. Both of these internal sensors enable potential attackers to monitor and analyze fluctuations in the FPGA's power consumption. Nonetheless, the usefulness of internal sensors in power analysis attacks is limited by challenges regarding their physical size and stealthiness. Recent studies have demonstrated that it is trivial to detect combinatorial loops like ring oscillators [La+20]. Deploying TDCs within the FPGA may result in significant hardware resource consumption, thereby increasing their susceptibility to detection.

In this paper, we present a novel internal sensor based on an 8-bit multiplier. This design incurs on minimal hardware overheads by using an innovative sampling method. The use of a common circuit like a multiplier increases the stealthiness of the proposed design. Furthermore, only half of the multiplier output (8-bit) provides enough precision for power monitoring of small circuits. We demonstrate the effectiveness of the proposed approach by conducting a power analysis attack on an unprotected implementation of AES-128.

The rest of the paper is organized as follows. Section 2 provides a complete

review of the state of the art on internal sensors used for remote power analysis. In Section 3 we describe our design rationale and characterize critical aspects of the proposed sensor. Section 4 describes the experimental methods used to validate the functionality of the proposed sensor. Finally, Section 5 provides our final remarks and concludes the paper.

2 State of the Art

Power analysis attacks exploit variations in power consumption during device operation, allowing attackers to deduce sensitive data. It was conventionally believed that such attacks required direct physical access to the targeted device and specialized measurement equipment. However, adversaries may leverage remote access to the target and internal sensors to retrieve the power footprint of the device. In FPGAs, these sensors detect changes in the propagation delay of logic elements. By exploiting the inverse relationship between power supply voltage and propagation delay, it is possible to mount power analysis attacks remotely. These attacks have been used to extract secret information in scenarios where the attacker and victim share the same FPGA but maintain logical isolation [Sch+21; Gra+19; ZS18]. And also when different FPGAs are used but share the same power supply [Sch+18; Sch+23]. In heterogeneous SoCs, the vulnerability against power analysis attacks persists even when the target resides within the CPU while the attacker uses the FPGA [ZS18; Gra+20]. This is made possible by the shared PDN that delivers power to all components within the SoC. As a result of the victim's operations, voltage fluctuations propagate through the system. This enables on-chip delay sensors to reveal a picture of the power footprint of the system.

Several types of internal sensors have been proposed in the literature. The largest groups rely on the step response of a delay line (TDCs) and the propagation of a pulse within a hardware loop (ring oscillators). TDCs are logical delay lines, comprised of uniform buffers through which a stimulus propagates. The output of each buffer is sampled by a register. The Hamming weight of these registers is directly associated with the propagation delay of the buffers, providing insights into voltage fluctuations. Different works have demonstrated the feasibility of using this approach to perform remote power attacks [Sch+21; Gla+20]. Ring oscillators are circuits that produce oscillations based on a feedback loop with an odd number of inverters. The frequency of oscillation is highly dependent on the power supply voltage. As the voltage fluctuates, it directly impacts the speed of the loop elements. This self-oscillatory signal can be used to clock a counter, which is then sampled to produce the observations. An adversary can study the variations in the counter value to deduce information about the internal operations of the target [ZS18; Gra+19].

Ring oscillators and TDCs have different drawbacks. The former, are small but suffer from quantization errors that arise from misalignment between the sampling clock and the ring output [ZS18; Gra+19]. Consequently, an array of sensors may be required to mitigate the effect of quantization errors [Gra+19]. This can result in an increase in resource usage. They also are easily detectable through bitstream checking techniques [La+20; Gna+18] and electromagnetic inspection [Bay+16]. On the other hand, TDCs suffer their large resource utilization. They also require precise placement and careful calibration of the de-

lay line. Similar to ring oscillators, TDCs may be detected by using bitstream checking techniques as their structure is well understood [La+20].

Prior work has shown the possibility of dynamic calibration [KGT20; SGS23] but this further impacts the hardware costs. Usual self-calibration strategies rely on coarse delay elements (LUT) and fine delay elements (CARRY). Other approaches [Zic+13] suggest to induce a clock phase shift between the clock sourcing delay line and the clock sampling the output register. But this method has the disadvantage that clock jitter introduces significant noise [KGT20]. Another method for fine-tuning the initial delay of TDCs uses the adjustable input delays found in FPGAs (IDELAY) [Udu+21]. In that work, the authors propose a TDC implementation that avoids placement constraints and reduces the resource utilization.

In [SGS23], the authors address the placement placement issue of TDCs by using FPGA routing resources to create the tapped delay line. That work also proposes to use self-calibration. However their sensor requires significant resources as its resolution is proportional to the size of the output register. In [Udu+21], the authors propose another TDC that also avoids placement constraints and uses fewer hardware resources. This is achieved by leveraging the IDELAY elements to precisely calibrate the sensor. Their approach was further developed in [Udu+22] by replacing the delay line with a logic circuit which can measure the pulse width of an oscillator. That approach managed to reduce the resource utilization to just three LUT elements. The resource utilization boundary is pushed in [JUP23] where a single LUT to construct a delay sensor. The idea relies on leveraging the propagation delay across the LUT to retrieve the power dissipation of the circuit. The authors use an IDELAY to precisely sample the LUT output. The drawback of this sensor is that while it can reveal some information, many traces seem to be required to perform a successful attack. Moreover, the required number of traces is inversely related to the sampling frequency (Fig. 23 of [JUP23]). Therefore, the sensor must be clocked by a high frequency circuit like a PLL.

Arithmetic circuits play a fundamental role in digital systems. They serve as fundamental components for various computational tasks. Their widespread availability emphasizes their importance in modern computing architectures. However, recent works have revealed a novel concern regarding the potential use of arithmetic circuits as voltage sensors [Gna+21]. This capability introduces a new aspect to the assessment of security threats. It emphasizes the importance of understanding and addressing potential vulnerabilities arising from the misuse of these circuits in digital systems. In [Gna+21], the authors illustrate that on-chip voltage sensors can take the form of regular circuits. This challenges the efficacy of bitstream checking techniques that rely on detecting particular structures. The authors of that work explore the possibility of leveraging the arithmetic functions of an arithmetic and logic unit by re-purposing them as on-chip voltage sensors. This method eliminates the need for placement and route constraints. By overclocking the ALU, its output value can be captured before reaching its final state. Due to delayed carry propagation during voltage fluctuations, observing the output value provides a picture of the power footprint of the device. While the method of monitoring the output of arithmetic circuits running at higher frequencies may initially seem stealthy, the solution reached in [Gna+21] is characterized by significant resource consumption. In their work, the authors use a 192-bit adder to mount a successful attack with 15e4 traces.

In Table 1 we summarize the main characteristics of the different on-chip voltage sensors reviewed. Importantly, it is observed that there is a lack of small sensors whose placement is unconstrained and that possess high sensitivity. Our paper aims at closing this gap. We propose the use of a very small sensor with high sensitivity. It uses a common multiplier as its base, making it difficult to detect. We adopt a similar approach to [Gna+21] demonstrating the utility of arithmetic circuits as voltage sensors. However, we introduce a novel method for output capture and propose a new sampling technique that significantly reduces the required resources. Moreover, the proposed method doesn't require running the arithmetic circuit at high frequencies to use it as a sensor, making it independent of the operating frequency.

3 Sensor design

Given our primary emphasis on minimizing resource consumption, we conducted a comparative analysis of the response behavior of adders and multipliers. For this purpose, we implemented ripple carry adder and parallel multiplier circuits of different sizes on an AMD-Xilinx Artix7 FPGA (XC7A100-2TFTG256) and performed a post-implementation simulation using Vivado 2020.2. In both cases, we selected inputs resulting in periodic output variations from zero to the maximum value (all bits set). Our experiments showed that the carry propagation delay in multipliers is significantly greater than that of ripple adders, which rapidly converges to its final value. This observation aligns with the known complexity introduced by multipliers due to partial product generation and accumulation, making them inherently slower than adders. When the goal is to minimize resource usage for sensing the carry propagation then choosing a multiplier becomes preferable. Adders, being fast, present challenges for capturing carry propagation unless a large adder is used or a precise delay element is introduced.

In Fig. 1, we present the design of the proposed sensor. This circuit is based on an 8-bit multiplier with an 8-bit output. The first input (A) is a constant with a value of 0x01. The second input (B) is a periodic 8-bit signal at a frequency f_s which commutes between 0x00 and 0xFF. Depending on these inputs, the output (D) fluctuates between 0x00 and 0xFF. The goal for choosing these inputs was to set all the output bits. A possible alternative was to use a 0xFF constant and a single oscillatory input, but this leads the toolchain to optimize the multiplier circuit rendering it too fast for use. We only use half the multiplier output as this allows to reduce hardware resources while maintaining enough sensitivity to perform power analysis on the device.

Given that the multiplier introduces a propagation delay before reaching its final output, we propose using a sampling clock (C) with frequency f_s, but phase-shifted (f_s'). In contrast to [Gna+21], this technique eliminates the need to operate the multiplier at higher frequencies to sense its propagation delay. While it remains essential to operate the sensor at higher frequencies for attacking fast architectures, using the phase-shifted clock allows for effective sensing without requiring increased operating frequencies. By sampling the multiplier output using f_s', we obtain intermediate output values between 0x00 and 0xFF. These intermediate values provide insight into the extent of the carry propagation. The range of phase shift values corresponds to the phases where the

Table 1: Characteristics of the on-chip voltage sensors used in remote power attacks

Year	Ref.	Sensor	Structure	Hardware resources				Best key rank	Easiness of deployment	Stealthiness	Hardware utilization
				LUT	FF	CARRY	Other				
2019	[Gra+19]	RO	Combinational loop	1	8	0	DECODER	-	++	−−−	++
2021	[Sch+21]	TDC	Tapped delay line	32	160	32	-	∼15e3 [SGS23]	−−−	−−	−
2021	[Gna+21]	ALU	Arithmetic unit	-	192	-	ALU	150e3 [Gna+21]	++	++	−−−
2021	[Udu+21]	VITI	Tapped delay line	4	4	0	2×IDELAY	20e3 [Udu+21]	++	+	++
2022	[Udu+22]	PPWM	Delay element	3	1		IDELAY	16e3 [Udu+22]	++	+	++
2023	[SGS23]	RDS	Tapped delay line	32	160	24	-	∼12e3 [SGS23]	++	+	−
2023	[JUP23]	1LUT	Delay element	1	1	0	IDELAY, PLL	100e3 [JUP23]	+	+	+++
2024	Ours	MULT	Arithmetic unit	2	8	4	PLL	25e3	++	++	++

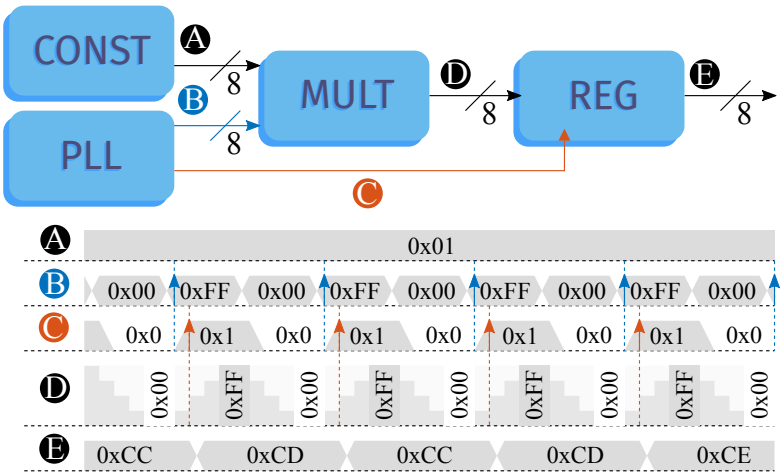


Figure 1: The proposed multiplier-based sensor design

output differs from 0x00 and 0xFF. The dynamic capabilities of FPGA clock generator primitives, such as phase-locked loop (PLL) and mixed-mode clock manager (MMCM), facilitates easy phase shifting of the clock enabling the implementation of our proposed sensing approach.

3.1 Implementation

It is crucial to consider two key factors when selecting the type of multiplier for constructing our sensor. First, the carry propagation delay must be sufficiently large to allow the sampling clock to capture the intermediate output values. Second, we aim to use minimal resources in building the multiplier. To explore these factors, we compared two different parallel multipliers: an AMD-Xilinx IP multiplier and an RTL description from the `numeric_std` package, which is part of the IEEE standard VHDL libraries. We implemented both types of 8-bit multipliers on an AMD-Xilinx Artix7 FPGA. We used the inputs described in Fig. 1 and set f_s=200 MHz. The implemented circuits were then simulated using Vivado 2020.2 to study the timing behavior of the circuit. For the AMD-Xilinx IP multiplier, it uses 4 slices of the FPGA (4 CARRY4 + 2 LUT6), with a potential range of sampling-clock phase-shift varying from +0.046 degrees to +39 degrees. It's important to note that these simulated values cannot be produced by an MMCM, as it provides a limited set of possible phase shifts. At 200 MHz, the corresponding effective range of phase shift generated by the MMCM varies from +9 degrees to +36 degrees. On the other hand, the RTL multiplier consumes 4 CARRY4 and shows a negligible propagation delay. This implies that using a sampling clock produced by a PLL/MMCM would make it challenging to capture the intermediate output values. Based on these results, we opted to use the AMD-Xilinx IP multiplier.

3.2 Characterization

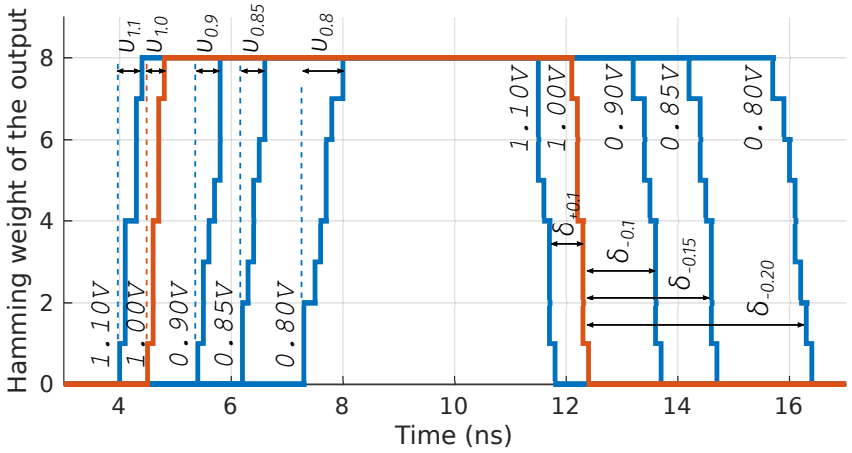
We studied the response of the proposed sensor to physical changes. For this, we implemented the 8-bit multiplier on a CW305 prototyping board (XC7A100-

2TFTG256) and used different core voltages and input frequencies. The synthesis and configuration was performed using the AMD-Xilinx Vivado 2020.2 toolchain. The multiplier outputs were captured using a digital oscilloscope with a sampling rate of 10 GSpS.

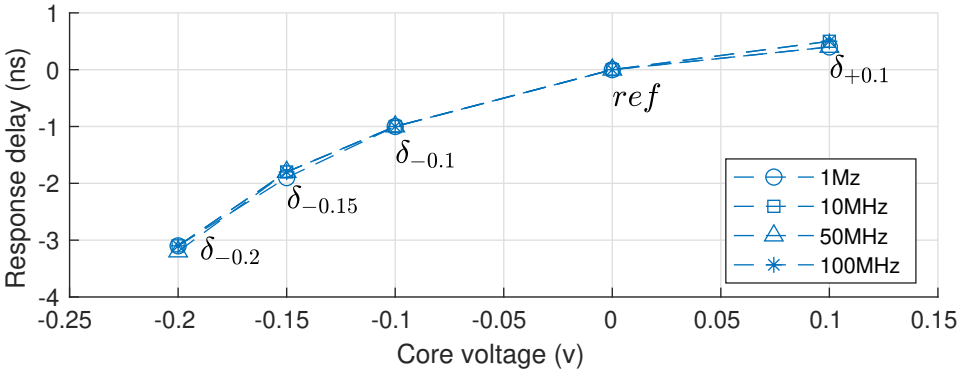
To analyze the voltage dependency of the multiplier, a gradual variation in the FPGA core voltage was performed with a step of 0.1V, considering the nominal voltage for this core set at 1.0V. The eight output signals of the multiplier output were captured using an oscilloscope. For signal extraction, a PMOD connector embedded in the development board was used, powered by the board's supply voltage of 3.3V. As the output is an analog signal, it was converted to a digital signal using a threshold of 2V. Signals greater than 2V were considered a logical one, while signals less than 2V were considered logical zero. The Hamming weight, indicating the count of logical ones in the output, was computed to provide a quantitative measure of the output characteristics. In a second experiment, our goal was to investigate how variations in the frequency of the input signal impact both the propagation delay and response delay of the multiplier. To achieve this, we used multiple frequencies of the input pattern and measured the corresponding response delay and propagation delay at each frequency while adjusting the core voltage. The results of these experiments are presented in Fig. 2.

As illustrated in Fig. 2a, variations in the core voltage have a noticeable impact on both the response delay and propagation delay of the multiplier. To establish a baseline for analysis, values of response delay and propagation delay at the nominal voltage of 1.0V were taken as reference. A significant observation emerges from the analysis. When the core voltage decreases, there is a considerable increase in the response delay variation (δ in Fig. 2a), indicating a longer time for the multiplier to converge. Conversely, an increase in core voltage results in an observable decrease in response delay, showing that the multiplier responds more quickly when the voltage is higher. Considering Fig. 2a, it is also apparent that the observed correlation in response delay extends to propagation delay, indicating a relationship between the time taken for the carry to propagate and fluctuations in core voltage. Specifically, a decrease in core voltage aligns with an increase in propagation delay (υ in Fig. 2a), while an increase in core voltage results in a decrease in propagation delay.

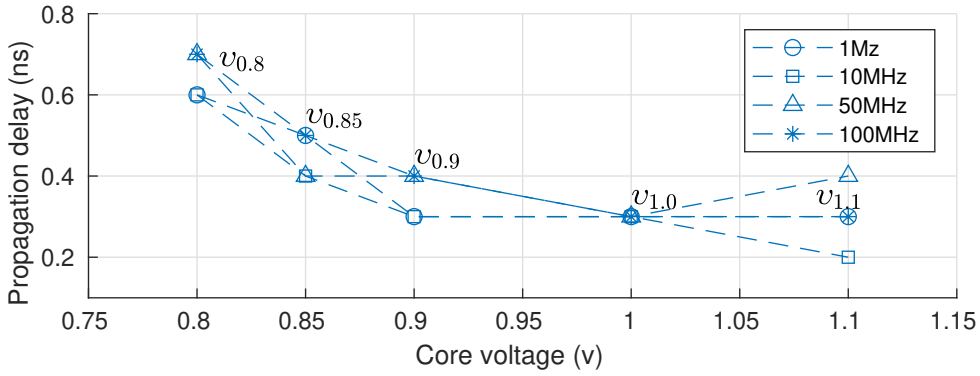
In Fig. 2b, it is apparent that altering the frequency of the input pattern does not affect the response delay at a given core voltage. The presence of negative values for response delay indicates a timing shift of the output signal to the left of the reference, suggesting that the output emerges earlier than the output reference, which is the output at 1.0V. Similar observations are noted for propagation delay, as depicted in Fig. 2c. Despite variations in frequency, the propagation delays remain relatively consistent at a given core voltage. These results imply that using the multiplier at higher frequencies may not be necessary to exploit propagation behavior, as it is primarily influenced by voltage variations. Conversely, these experiments demonstrate that the sensor is expected to behave consistently with any input frequency, so long as the setup times are respected.



(a) time response as a function of the voltage (50MHz)



(b) response delay as function of the frequency



(c) propagation delay as function of the frequency

Figure 2: Time response of the 8-bit multiplier

3.3 A stealthier sensor

Despite the increased sensitivity of the proposed sensor, we can identify two main limitations for its utilization. The first, is that the optimal inputs are

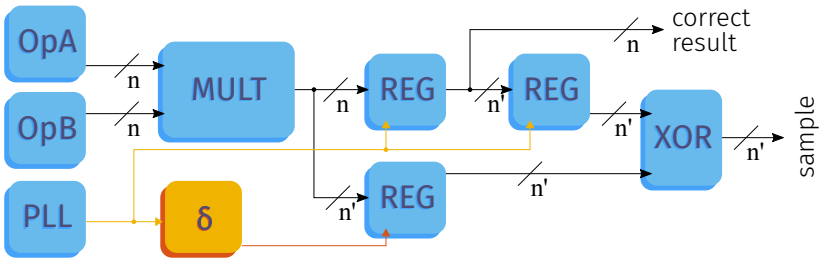


Figure 3: A strategy for integrating a delay sensor into an operational multiplier

those which produce a periodic commutation of the output state. Therefore, it is necessary to connect the multiplier in a way that such values are provided. This would be trivial in heterogeneous SoCs as it could be sourced directly from a malicious application in the processor. Another alternative, however, would be to sample different end points of the multiplier circuit using two registers and then comparing their values. This would allow to use random inputs and therefore to sequester an operational multiplier. The second challenge comes from the use of a PLL to generate the phase shift in the sampling clock. These elements are not always available in FPGAs and the number of possible phase shifts is limited. Instead, logic delay elements like IDELAY and clock buffers (BUFG) could be used to produce the sampling clock. In Fig. 3 we illustrate a potential architecture for a stealthier multiplier sensor.

4 Experimental analysis

To evaluate the effectiveness of the proposed multiplier-based sensor we conducted a power analysis attack using a Zybo development board (XC7Z010-1CLG400C). This platform is equipped with an AMD-Xilinx Zynq 7000 SoC-FPGA featuring two ARM Cortex-A9 CPUs and an Artix-7 FPGA reconfigurable nucleus. The AMD-Xilinx Vivado 2020.2 toolchain was used to generate the FPGA bitstreams and launch applications through Vitis. Default Vivado strategies were used for synthesis and implementation.

We propose an attack scenario where both the adversary and the victim share the processing system and programmable logic, but they are physically isolated in the PL part. The victim is a 128-bit AES hardware accelerator, controlled by one of the processing system's cores. It responds to encryption acceleration requests initiated by the malicious application on the second ARM core, receiving plaintexts and returning ciphertexts. Within the same fabric, the adversary deploys a multiplier-based sensor, and its 8-bit output is connected to a dual-port BRAM. The first port allows reading the BRAM content from the processor part using the malicious application, while the second port allows writing sensor data directly from the FPGA. For offline analysis, the BRAM content is stored in DDR memory and then on an SD card. The malicious application controls an MMCM to generates the clock dedicated to produce the multiplier input signal and a phase-shifted clock for sampling the sensor output. Before initiating the attack, the malicious application adjusts the sampling clock phase to capture intermediate values of the multiplier output within the range of

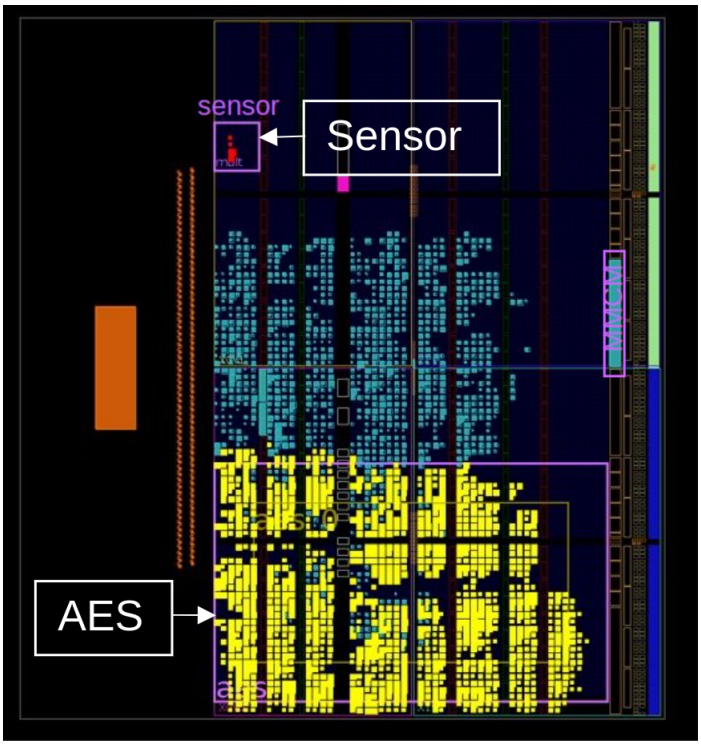


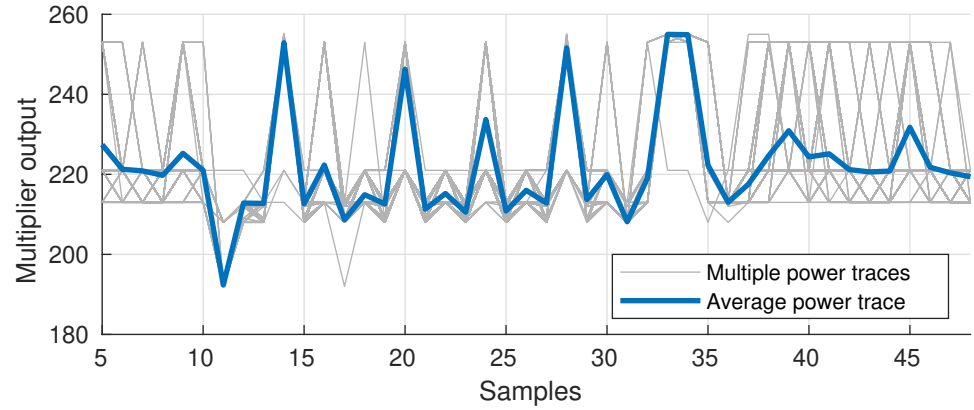
Figure 4: Floor-plan used in the experimental setup

0x00 to 0xFF. We selected a phase shift generating an output with a Hamming weight around 4, representing half the maximum Hamming weight when the circuit is at rest (i.e., PDN is not stressed, and the AES accelerator is not active). To facilitate the trace acquisition, we used the start and end signals of the AES accelerator as triggers to store the sensor data.

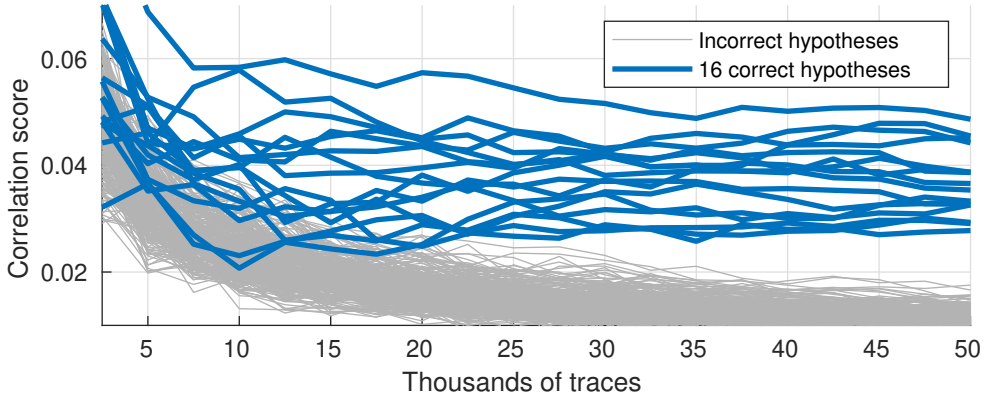
Fig. 4 depicts the layout of the experimental architecture, providing a view of the positioning of system components. The sensor is situated 40 slices away from the AES circuit, constituting a far setup. We set the AES accelerator frequency to 100 MHz and the sensor frequency to 200 MHz. Attacking the target at a higher frequency helps to explore the sensor's capabilities. We acquired 1e6 traces from the encryption operation using different plaintexts. Fig. 5a shows some sample traces, where distinct voltage drops during the AES processing are apparent. We performed correlation power analysis targeting the last AES round. As depicted in Fig. 5b, all 16 key bytes were successfully revealed with a random subset of 3e4 traces. Nonetheless, for the majority of the bytes, an accurate key guess could be made with around 2e4 traces.

5 Conclusions

In this paper, we have presented a new internal sensor which reduces the resource utilization and detectability. With a net hardware utilization of 4 slices, we managed to mount a correlation power attack on an unprotected imple-



(a) sample traces



(b) correlation power analysis

Figure 5: Power analysis of an unprotected implementation of AES running at 100 MHZ, sampled at 200 MHz

mentation of AES-128. Under 25e3 power traces were required to retrieve the full encryption key. These results demonstrate the potential of exploiting small computing elements for use in power monitoring. Furthermore, the proposed approach is free from any placement or routing constraints.

While sensors in the literature often have larger sizes, the proposed sensor features a notably small and unconstrained design. This characteristic not only ensures flexibility but also enhances stealthiness, making detection more challenging. Furthermore, the introduced sensor excels in sensitivity. Our design proves to be robust in capturing intricate details of power variations. Especially when considering its small size and the number of traces needed to compromise a cryptographic architecture like AES.

The key feature of the proposed sensor is the use a fixed delay in the sampling clock. The clock generates the input pattern, while a derived clock—phase-shifted relative to the first—is used to sample the multiplier output. In our study, we used the dynamic phase capability of the AMD-Xilinx MMCM

to achieve this phase shifting. However, it's worth noting that these clock elements have restrictions in providing all possible phases, impacting the sensor's resolution. This limitation implies that not all carry propagation delay levels within the multiplier can be captured. This makes the sensor resolution dependent on the phase values available. Increasing the number of generated phases can enhance the sensor's resolution and decrease the number of traces needed for a successful power analysis attack. Alternatively, techniques involving logic delays could be used to generate the sampling clock instead of performing a phase modulation.

Acknowledgment

This work has been supported by the French government through the *Agence Nationale de la Recherche* in the framework of the *France 2030* initiative under project ARSENE (ANR-22-PECY-0004).

References

[Bay+16] Pierre Bayon, Lilian Bossuet, Alain Aubert and Viktor Fischer. "Fault model of electromagnetic attacks targeting ring oscillator-based true random number generators". In: *Journal of Cryptographic Engineering* 6 (2016), pp. 61–74. DOI: 10.1007/s13389-015-0113-2.

[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 et al. "Stealthy Logic Misuse for Power Analysis Attacks in Multi-Tenant FPGAs". In: *Proceedings of the 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 1012–1015. DOI: 10.23919/DATE51398.2021.9473938.

[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 2020 International Conference on Smart Card Research and Advanced Applications (CARDIS)*. Springer, 2020, pp. 109–125. DOI: 10.1007/978-3-030-42068-0_7.

[JUP23] Darshana Jayasinghe, Brian Udugama and Sri Parameswaran. "1LUT-Sensor: Detecting FPGA Voltage Fluctuations using LookUp Tables". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 1 (2023), pp. 51–86. DOI: 10.46586/tches.v2024.i1.51-86.

[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), pp. 121–146. DOI: 10.13154/tches.v2020.i3.121-146.

[Kha+18] Ahmed Khawaja et al. "Sharing, Protection, and Compatibility for Reconfigurable Fabric with AmorphOS". In: *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. USENIX Association, 2018, pp. 107–127.

[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 Technology and Systems* 13.3 (2020), pp. 1–31. DOI: 10.1145/3402937.

[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.

[Sch+23] Kai Schoos, Sergej Meschkov, Mehdi B. Tahoori and Dennis R. E. Gnad. "JitSCA: Jitter-based Side-Channel Analysis in Picoscale Resolution". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 3 (2023), pp. 294–320. DOI: 10.46586/tches.v2023.i3.294-320.

[SGS23] David Spielmann, Ognjen Glamočanin and Mirjana Stojilović. "RDS: FPGA Routing Delay Sensors for Effective Remote Power Analysis Attacks". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2023), pp. 543–567. DOI: 10.46586/tches.v2023.i2.543-567.

[Udu+21] Brian Udugama, Darshana Jayasinghe, Hassaan Saadat, Aleksandar Ignjatović and Sri Parameswaran. "VITI: A Tiny Self-Calibrating Sensor for Power-Variation Measurement in FPGAs". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2021), pp. 657–678. DOI: 10.46586/tches.v2022.i1.657-678.

[Udu+22] Brian Udugama, Darshana Jayasinghe, Hassaan Saadat, Aleksandar Ignjatovic and Sri Parameswaran. "A Power to Pulse Width Modulation Sensor for Remote Power Analysis Attacks". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 4 (2022), pp. 589–613. DOI: 10.46586/tches.v2022.i4.589-613.

[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)*. ACM, 2013, pp. 101–104. DOI: 10.1145/2435264.2435283.

[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.