Combined Internal Attacks on SoC-FPGAs: Breaking AES with Remote Power Analysis and Frequency-based Covert Channels

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

▶ **To cite this version:**

Anis Fellah-Touta, Lilian Bossuet, Carlos Andres Lara-Nino. Combined Internal Attacks on SoC-FPGAs: Breaking AES with Remote Power Analysis and Frequency-based Covert Channels. 8th IEEE European Symposium on Security and Privacy (EuroSP2023), IEEE, Jul 2023, Delft, Netherlands. 10.1109/EuroSPW59978.2023.00035 . hal-04165230

HAL Id: hal-04165230
https://hal.science/hal-04165230v1

Submitted on 18 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Combined Internal Attacks on SoC-FPGAs: Breaking AES with Remote Power Analysis and Frequency-based Covert Channels

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

In recent years, the field of side-channel analysis has observed a revolution in the design of the attack methodology. Conventional approaches which require the use of highly specialized equipment like oscilloscopes and spectrum analyzers, despite highly precise, might be regarded as impractical in some scenarios. On the other hand, the use of less-accurate internal sensors which can monitor the power footprint of a circuit has risen in popularity. Delay sensors have shown promising results. These structures are interesting since they can be implemented from regular hardware resources available in most circuits. This means that components already available in the target platform might be leveraged to implement a side-channel attack. Moreover, it has been shown that is not necessary to have direct access to the platform to carry out such an attack; which implies that if there is a remote link such as Ethernet, an adversary might be able to perform Remote Power Analysis (RPA) of the system. So far, the main challenge for the success of this kind of attack is the problem of cutting and aligning the power traces. This is usually achieved through secondary digital channels which carry some trigger information. In this paper, we simplify the conditions for an RPA attack to take place. Namely, our method mitigates the need for connecting digital triggers to the remote sensor. We demonstrate this approach by performing a successful key recovery on a hardware implementation of AES.

Keywords: AES, Covert Channels, Remote Power Analysis, TDC.

Please cite as:

```
@InProceedings{FBL23,
title = {{Combined Internal Attacks on SoC-FPGAs:  Breaking AES with Remote Power Analysis
and Frequency-based Covert Channels}},
author = {Fellah-Touta, Anis and Bossuet, Lilian and Lara-Nino, Carlos Andres},
booktitle = {Proceedings of the 2023 IEEE European Symposium on Security and Privacy
Workshops (EuroS&PW)},
pages = {281--286},
year = {2023},
publisher = {IEEE},
location = {Delft, Netherlands},
doi = {10.1109/EuroSPW59978.2023.00035},
isbn = {979-8-3503-2720-5}}
```

1 Introduction

Most of the modern applications of cryptography are considered secure from an algorithmic point of view. It is understood that, under reasonable assumptions, formal or statistical proof of security warranties that an adversary cannot compromise the security of the system. However, if the threat model supposes that the attacker has physical access to the platform, these notions of security wane [Byr13]. Under such scenarios, it is necessary to design and implement protections which can mitigate the information leakage of the hardware platform [ISW03; PR13].

A side channel is, in a simplified way, a physical magnitude which can be measured and correlated with the operations being performed in the platform. Electromagnetic emanation, power dissipation, thermal irradiation, energy consumption, and noise are prime suspects for information leakage [MOP08]. Then, we can envision side channel attacks [KJJ99; BCO04] as the application of a sensing strategy with a subsequent information-processing step. For the first part, it is necessary to possess a sensor which can transform the physical magnitude into a digital signal that can be understood by a computer. Such a system will then process the samples in a given time. With the adequate quality and volume of data it might be possible to break just any cryptographic implementation. Our work focuses on the first of the two aforementioned tasks, we investigate the process for obtaining these data.

According to the sampling theorem [PM06], the first requirement for the appropriate acquisition of samples is to use a sampling frequency at least two times greater than that of the target magnitude. Secondly, the sensor has to perform a quantization step which will encode the sample into a finite array of possible values, generally through binary representation. The more bits we use, the greater the sampling resolution will be, and subsequently we will obtain a greater fidelity to represent the signal of interest. However, more bits and more samples also imply that we must store and transmit more data. In the end, the goal of a sampling scheme is to adequately represent the physical magnitude with the minimal storage and bandwidth costs.

These are not trivial challenges when the targets of the sensing are digital circuits. Most of the time, the operational frequency of these designs is in the order of gigahertz (GHz). All the while the signals to be sampled generate transceiver outputs in the order of micro-volts (μV). Consequently, a sampling scheme for a digital system must produce a few billion samples per second, with the adequate resolution to represent fluctuations of the millionth part of a volt! Thankfully, the quantization step depends not so much on the scale of the physical magnitude to be measured, but rather on the dynamic range of the signal.

It should be evident that performing side channel analysis of a chip requires specialized equipment. Multi-channel digital oscilloscopes and high-frequency spectrum analyzers are some of the most popular tools for this task [MOP08]. Yet, their monetary costs are so high that only private companies and large laboratories can acquire these systems. Then, even if the adversary has the resources to acquire the sensing hardware (which would be the case in state-sponsored attacks), they must also gain physical access to the target platform. These two limitations could be sufficient for a designer to declare that their implementation is totally safe from side channel attacks. Nonetheless, recent

works [ZS18; Sch+21] have demonstrated that neither expensive equipment nor physical access to the chip are necessary to perform power analysis on a digital circuit.

Using internal sensors created from digital components is not a particularly novel idea [Fra+10; Hen10]. These constructions have been employed to monitor the operation of the circuit and assess whether everything is working as intended. However, until recently it was not considered viable that such sensors could be used to retrieve sensitive information from the platform. It is now known that sensors based on Time-to-Digital converters (TDC) [Gra+21] and Ring-Oscillator (RO) [Gra+19] can be exploited to perform power analysis with moderate accuracy. Moreover, since these circuits can be implemented and operated remotely, performing RPA on a chip is just a matter of exploiting some software vulnerability [SB15]. It is no longer required to assume that the attacker has physical access to the target.

Despite the evident advantages, RPA attacks must still cope with the original sensing problems: obtain more samples with more resolution, while reducing storage and transmission costs. The latter is a particularly interesting problem, since a remote origin of the attack also means that the sampled data must travel back to this origin. Ubiquitous connectivity might provide viability for this approach. The Internet-of-Things (IoT) supposes that everyday objects will be connected to the internet. If the attacker finds a way to access the circuit remotely, then they can leverage hardware components already in the platform to mount an attack, and possibly compromise the security of the system.

A problem so far not sufficiently addressed in the literature is that RPA, just as classical power analysis, requires some additional information for cutting and aligning the traces. When the attacker has access to the platform, we assume that they can poke around until they find some *trigger* signal which can be used to determine the start of a *trace*. However, in a remote-attack model this is not trivial. We cannot assume that the *start* and *done* signals of the target will be available to the sensor. Therefore, the problem of remotely determining the point for cutting and aligning the traces is of significant relevance.

In this work, we use the same sensors that are employed to perform RPA in [Gra+19; Sch+21]. Our attack model considers that the adversary must only retrieve a train of samples from the internal sensor to conduct RPA. Unlike previous works [Sch+21], we do not depend on the apparent voltage drop of the traces to perform the attack. We achieve this feat by leveraging strategies previously used for covert-channel communications [BB18]. Thus, our approach can be considered more generalized, and it will be effective even when the behavior of the target is not apparent. For example, if there is anything else operating on the same platform.

To demonstrate the viability of this approach we perform the RPA of an unprotected implementation of AES. Our findings suggest that the proposed approach is viable as we managed to align the power traces with high accuracy and recover the secret key. Nonetheless, there are multiple limitations that must be addressed to reach a point where such an attack becomes a practical concern, for example for an IoT platform. In particular, the problem of obtaining and transmitting large volumes of data is critical for carrying out a successful attack on the target circuit. This is a characteristic of remote attacks which falls out of the scope of this work.

The rest of the paper is structured as follows. In Section 2 we describe our

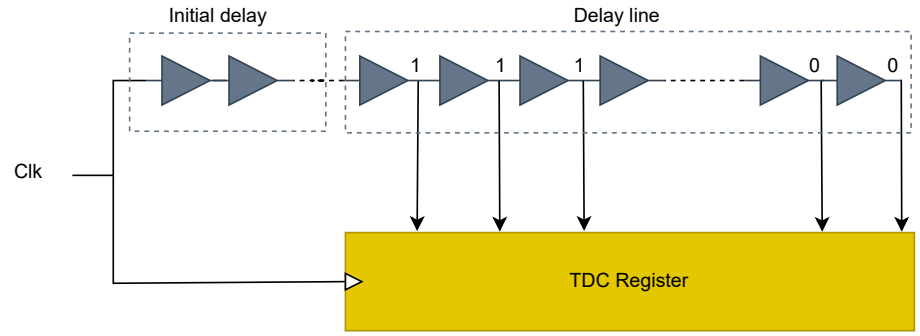


Figure 1: TDC-based sensor design.

methodology and the materials used in our experimentation. In this section we also provide a formal description of the proposed attack scenario. The derived results are subsequently reported in Section 3. Finally, our findings and conclusions are summarized in Section 4.

2 Materials and Methods

In this Section we provide details regarding our experimental setup. We describe the different components of our system and outline the guidelines for the proposed attack.

2.1 Threat Model

Our threat model follows the same threat model of RPA attack [Gla+20; Ram+18] which assumes a shared FPGA in cloud or datacenter. Each FPGA tenant can access a specific FPGA region which is logically and physically isolated from the other region, but all FPGA regions use the same power distribution network (PDN). Tenants can create practically any hardware design except for designs that could be blocked by Design Rule Checks (DRCs) like combinational loops. They can also employ customized FPGA logic called shell to access external interfaces such as PCIe and external DDR. The attacker deploys on-chip sensors to detect information leakage from the unwary victim that replies to the legitimate encryption requests by encrypting plaintext and sending the ciphertext over a public channel to the attacker. The adversary has the possibility to retrieve the power traces outside of the circuit for the analysis by reading the memory that saves the measurement. Just before each encryption call the adversary modifies the frequency of a reconfigurable PLL to send the synchronization information covertly to the system acquisition.

2.2 Remote sensors

A delay sensor is a circuit capable of measuring the variation 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. Therefore, as the chip performs different processing tasks, these will

influence the delay propagation of the target oscillator. This delay will be measured and then sampled into a digital signal. Both Time-to-digital converters (TDCs) and ring oscillators (ROs) can be used to implement delay sensors.

Ring-oscillator-based sensors have a smaller implementation size, so they are easy to deploy. However, power analysis attacks based on ring oscillators are easier to identify, especially in cloud environments where RPA is very practical [La+20]. In the other hand, TDCs have high sensitivity compared to ROs and can detect nanoscale-scale voltage fluctuations. But TDCs have also their drawbacks. They demand more logical resources and need calibration of the initial delay and precise placement requirements. Recent works [KGT20] propose TDC sensor which allows runtime calibration. This approach mitigates the need for calibration before the deployment of the sensor. In this work we use the TDCs sensors from [Gra+19] to perform the data acquisition. Figure 1 illustrates the general architecture of this circuit.

The Initial delay block is a chain of buffers that a clock signal passes through. It must be calibrated so that the edge of the input signal can be captured by the TDC register. The delay line consists also of a uniform chain of buffers. The propagation of the clock signal will depend on the voltage of the chip. To capture the output of each buffer, each output is connected to a latch that is enabled by the clock signal. If the hamming weight of the TDC register increases, it means that the pulse propagation has increased. Therefore, the propagation delay of the delay chain is reduced, and vice versa. Hence, the TDC sensor delivers a clear image of the voltage change inside the circuit by converting the propagation delay into digital values that can be processed by a computer.

2.3 Covert channels

Frequency-based covert channels have been explored in [BB18] with the goal of bypassing Trust-Zone protections in a Zynq-7000 SoC-FPGA. The authors demonstrated that it was possible to modify the output of Phase Locked Loops (PLL) in the circuit with different modulation strategies in order to encode a message. This would create a covert channel between different components that were not supposed to communicate with each other. For example, a trusted application would exchange information with an untrusted hardware accelerator.

In our work we do not use covert channels to encode a message. Rather, we intend to use a frequency modulation to align the power traces obtained by the delay sensor. One of the major challenges of remote power analysis is the alignment of power traces. In a real scenario it is impractical, if not impossible, to capture a trigger signal which 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 suggested using the same sensors to create a trigger mechanism by enabling the start of sample storage when a large voltage drop was 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 the circuit noise that can prevent this mechanism from working properly.

FPGA-enabled SoCs such as the Zynq-7000 boards feature different clocks which flow from the processing system into the programmable logic, see Figure.2. These are sourced from a group of main PLLs, and through a set of multipliers and dividers produce the desired frequency output. These multipliers and di-

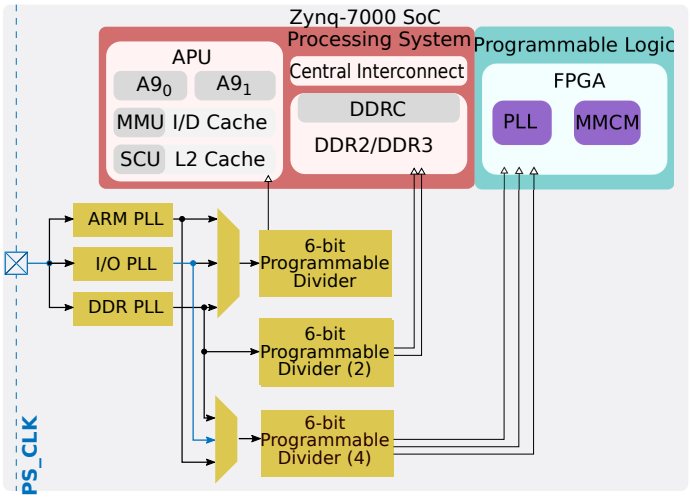


Figure 2: The clocking resources of Zynq-7000 SoC-FPGAs. Silicon PLLs are shown in yellow with the target reconfigurable PLLs presented in blue.

viders are simply digital values stored in registers which can be modified from the processors of the SoC.

In this work, we use a Zynq-7000 SoC-FPGA as implementation platform. To construct the convert channel, we will not use the silicon PLLs of Zynq, because their modification requires root privileges when the chip is protected by TrustZone, for example, even this technology may be compromised by other remote attacks which leverage fault-injection to achieve privilege escalation [Ben+17]. Instead, we will use the clocking resources which are mainly a mixed-mode clock manager (MMCM) and phase-locked loop (PLL). These two features are reconfigurable oscillators which allow to generate a desired frequency without any root privileges and allow dynamic change of the clock frequency, this means that user can modify the frequency at runtime. To implement the covert channel the attacker will modify the clock output frequency of MMCM during each circuit under attack call. To perform the acquisition, we detect this frequency change and create a trigger that controls the operation of the sensor.

2.4 Target architecture

The Advanced Encryption Standard (AES) is perhaps the most popular algorithm used for providing confidentiality and authentication to the data. It follows a substitution-permutation network (SPN) composition with a data input of 128-bits and keys of 128, 192, or 256 bits. The data is processed using its internal permutation or *round function* along 10, 12, or 14 rounds, which correspond to the key length. In our work we focus on the 128-bit variant which is the most commonly used.

The internal permutation of AES is composed of four transformations which confer diffusion and confusion to the data, see Fig. 3. These functions consist of substitutions, permutations, and finite-field operations with a respective invertible equivalent used in the decryption process.

The basic approach for implementing AES is to perform an encryption round

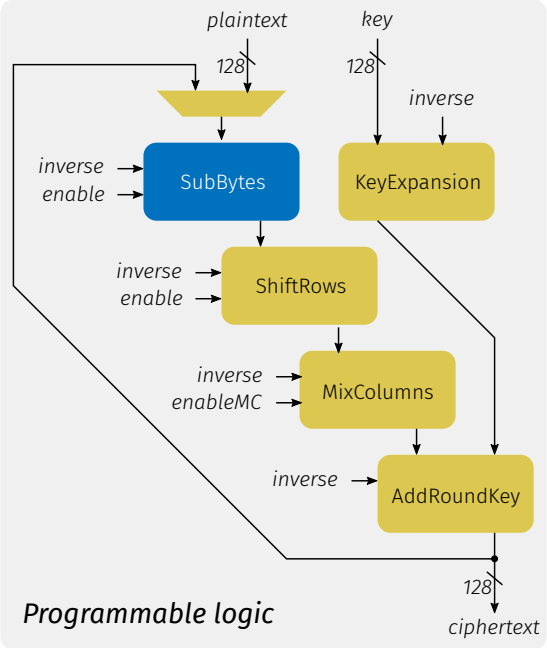


Figure 3: The encryption process using AES

per cycle, that is, to apply the four transformations of the round function consecutively and then to register the intermediate result. We use the same AES implementation from [Gra+19] which includes the forward and inverse transformations of the cipher. Only the encryption is used to perform our attack. The different blocks in the architecture can be disabled in order 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.

The `KeyExpansion` has a latency of two cycles, then we have an initial round, nine regular rounds, and a final round, for a total latency of 13 cycles per block. At 10MHz this implementation requires about 18% of the LUTs and 6% of the FF in the Zynq (xc7z010clg400-1).

We target the output of the SubBytes block in the first round, which combines the plaintext and the round keys. Since AES has a fixed-size input (128-bits), this means that a partitioning strategy must be used to divide longer messages in blocks. For the usual power analysis techniques [KJJ99; BCO04] we require a large number of known inputs and their respective outputs. If the AES accelerator is configured in *electronic code book* mode, which is unusual, then we must perform a large number of queries and provide a large input set. However, by configuring the accelerator in *code-block chain* mode we need only provide the initial plaintext and then observe the output of the cipher. That is a vulnerability found in all non-CCA secure modes of operation.

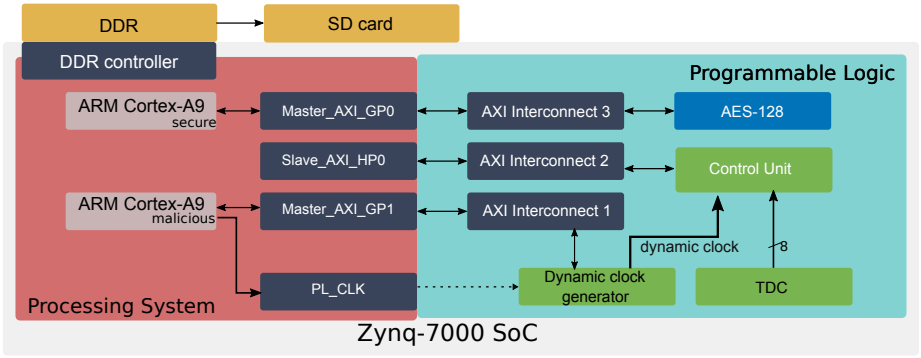


Figure 4: The architecture created to emulate an RPA attack. The hardware modules implemented in the programmable logic of the Zynq-7000 are controlled by the processing system using AXI4-Lite interfaces. The samples are stored in DDR and finally SD.

2.5 Experimental setup

We have implemented on a Xilinx Zynq 7000 heterogeneous SoC-FPGA an architecture which allows to demonstrate the viability of the proposed attack, see Fig. 4. The main change between this approach and a practical case study is that we do not transfer the information 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 can query the AES accelerator controlled by the second core, considered secure. The malicious process control a third party IP which is dynamic clock generator IP from digilent (Available from `https://github.com/Digilent/vivado-library`). This IP has a standalone driver which we use to control the MMCM primitive in the Zynq fabric with an AXI Lite interface from the PS. It allows to generate different frequencies at runtime. The main characteristic of this oscillator is that it will continue operating regardless of what happens to the main PLLs of the SoC. The FPGA fabric integrates TDC-based sensor sampled at 200 MHz and a control unit that manages the data store in the DDR. To read out the acquired data we write the DDR contents to an SD card. The frequency of the AES module is 10 MHz.

3 Experimental results

We used a Zybo development board in our experimentation. This platform features a Zynq-7000 SoC-FPGA (xc7z010clg400-1). We used the AMD-Xilinx 2022.1 toolchain, creating the hardware specification in Vivado and programming and launching the applications through Vitis.

3.1 MMCM's transition delay

We first studied the MMCM response times in order to assess whether it was viable to modify the frequency of these components from the processing system. Figure 5 shows the results for this experiment. We first enabled a digital trigger

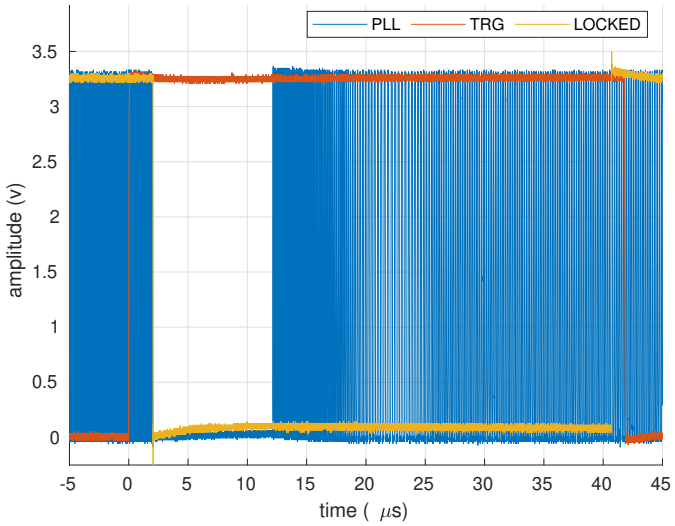


Figure 5: The step response of a MMCM in the Zynq-7000 SoC-FPGAs. Obtained with a digital oscilloscope at 20 GSps

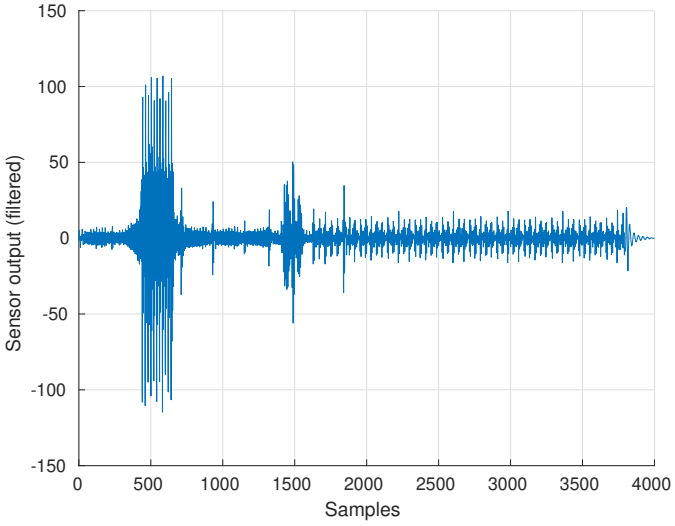
from the processor and then requested a frequency change from 66 MHz to 200 MHz. We estimated that it takes approximately 2 μs for the MMCM to start the process to modify its output. Then another 39 μs are used to perform the requested change; during this time the output of the MMCM is unstable. Finally, it takes approximately 1 μs for the processor to be notified of the change. In total, we need 42 μs to detect a rising frequency change. Since the MMCM calls are blocking and also they are before the encryption calls, it is not necessary to account for the transition delay in the acquisition.

3.2 The covert channel

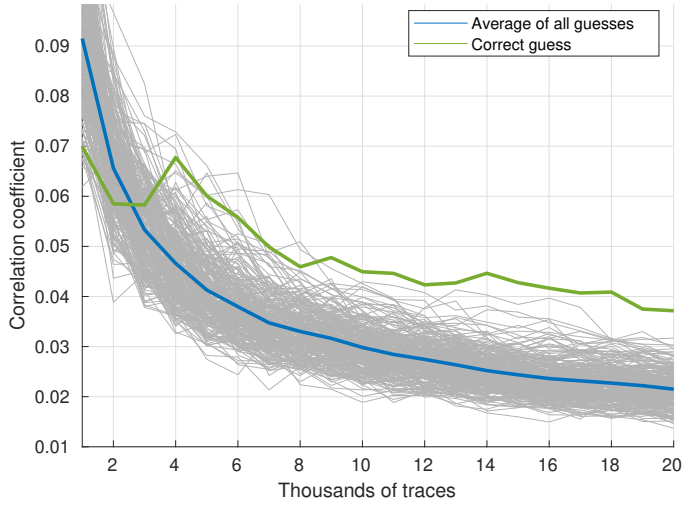
We used the algorithm in Fig. 6 to encode the synchronization information into the covert channel. Before each call to the cipher, the malicious application modified the output frequency of the MMCM. This value would iterate between two predefined values. To detect this frequency modulation, we measure the pulse duration of the clock signal. Depending on the pulse length, we generate a write enable signal to control the data transfer from sensor to memory.

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

Figure 6: The channel modulation strategy



(a) average of 10,000 traces



(b) as the number of samples increases

Figure 7: Correlation power analysis

3.3 Validation and analysis

To analyze whether the proposed approach would be practical we performed the acquisition of 20K traces from the encryption of AES for different plaintexts. In Fig. 7a we show the average of 10K of these traces. On first sight it is possible to identify the sequence of samples corresponding with the execution of AES. Despite the length of the traces affecting the execution time for performing an attack, this should not have an impact in the correlation analysis. Nonetheless, lengthier traces imply that more data must be transmitted. On the other hand, miss-alignment would have a negative effect for statistical attacks like correlation

power analysis (CPA). In our case the MMCM is not ideal which means that the generated clocks have slightly different phase shift. Thus, the power traces are fairly misaligned. Therefore, an alignment of power traces is needed before launching the CPA. By using cross-correlation technique we were able to remove the phase shift of the different power traces. Because we activate our trigger before the start of AES encryption we acquire some circuit noise that does not correspond to the encryption activity. Therefore, we tried to remove this noise before applying the cross-correlation to the power traces. To do this, we applied bandpass filter to allow only the target frequency.

We performed CPA attack over the set of traces with aims to verify whether it was possible to retrieve the secret key of the architecture under attack. Our target was the first key byte. We chose to conduct the CPA on the first-round key [Put+20], because the initial key is used during the first SubBytes (round0). Moreover, we looked for a non-linear function to have the maximum voltage variations. That's the reason why we chose the SubBytes of round1 as it is not linear given the properties of the AES'S-box. As it can be observed in Fig. 7b, the correct guess emerges with $\sim$ 9K traces and it separates from the group by the 20K traces with a clear advantage over the second guess.

4 Conclusions

In this paper we have described a new strategy for the automatic alignment of traces in the scope of RPA attacks. Our findings suggest that the proposed method is viable under certain assumptions. As case study we performed correlation power analysis on an unprotected implementation of AES and managed to successfully recover an octet of its secret key.

The proposed attack scenario considers that the attacker can manipulate the MMCM primitives of the FPGA and deploys TDC sensors in the fabric. However, the use of TDCs can be detected by bitstream-scanning strategies [Gna+18]. In [KBG09] authors proved that the added power noise will not affect the power attack result if this noise does not depend on the input data of the AES module. As a result, combining the sensor with a second module to mask its intent will not affect the attack efficiency provided that the activity of the added module is independent to the encrypted data.

Acknowledgements

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

[BB18] El Mehdi Benhani and Lilian Bossuet. "DVFS as a Security Failure of TrustZone-enabled Heterogeneous SoC". In: *ICECS 2018*. Dec. 2018, pp. 489–492. DOI: 10.1109/ICECS.2018.8618038.

[BCO04] Eric Brier, Christophe Clavier and Francis Olivier. "Correlation Power Analysis with a Leakage Model". In: *CHES 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 2004, pp. 16–29. ISBN: 978-3-540-28632-5. DOI: 10.1007/978-3-540-28632-5_2.

[Ben+17] El Mehdi Benhani, Cedric Marchand, Alain Aubert and Lilian Bossuet. "On the security evaluation of the ARM TrustZone extension in a heterogeneous SoC". In: *SOCC 2017*. Sept. 2017, pp. 108–113. DOI: 10.1109/SOCC.2017.8226018.

[Byr13] Eric Byres. "The Air Gap: SCADA's Enduring Security Myth". In: *Communications of the ACM* 56.8 (Aug. 2013), pp. 29–31. DOI: 10.1145/2492007.2492018.

[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: *SPL 2010*. Mar. 2010, pp. 133–137. DOI: 10.1109/SPL.2010.5483027.

[Gla+20] Ognjen Glamočanin, Louis Coulon, Francesco Regazzoni and Mirjana Stojilović. "Are Cloud FPGAs Really Vulnerable to Power Analysis Attacks?" In: *DATE 2020*. 2020, pp. 1007–1010.

[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: *FPT 2018*. 2018, pp. 286–289.

[Gra+19] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia, Philippe Loubet-Moundi and Olivier Francis. "Remote Side-Channel Attacks on Heterogeneous SoC". In: *CARDIS 2019*. Pragues, Czech Republic, Nov. 2019. URL: https://hal.science/hal-02380092.

[Gra+21] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia and Philippe Loubet Moundi. "SideLine: How Delay-Lines (May) Leak Secrets from Your SoC". In: *COSADE 2021*. Oct. 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.

[ISW03] Yuval Ishai, Amit Sahai and David Wagner. "Private Circuits: Securing Hardware against Probing Attacks". In: *CRYPTO 2003*. 2003, pp. 463–481. DOI: 10.1007/978-3-540-45146-4_27.

[KBG09] Najeh Kamoun, Lilian Bossuet and Adel Ghazel. "Correlated power noise generator as a low cost DPA countermeasures to secure hardware AES cipher". In: *SCS 2009*. 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 Trans. Cryptogr. Hardw. Embed. Syst.* 2020.3 (June 2020), pp. 121–146.

[KJJ99] Paul Kocher, Joshua Jaffe and Benjamin Jun. "Differential Power Analysis". In: *CRYPTO 1999*. Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 1999, pp. 388–397. ISBN: 978-3-540-48405-9. DOI: 10.1007/3-540-48405-1_25.

[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 Trans. Reconfigurable Technol. Syst.* 13.3 (Sept. 2020). ISSN: 1936-7406.

[MOP08] Stefan Mangard, Elisabeth Oswald and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Advances in information security. Springer, 2008. ISBN: 978-0-387-38162-6.

[PM06] John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing (4th Edition)*. USA: Prentice-Hall, Inc., 2006. ISBN: 0131873741.

[PR13] Emmanuel Prouff and Matthieu Rivain. "Masking against Side-Channel Attacks: A Formal Security Proof". In: *EUROCRYPT 2013*. 2013, pp. 142–159. DOI: 10.1007/978-3-642-38348-9_9.

[Put+20] Septafiansyah Dwi Putra, Arwin Datumaya Wahyudi Sumari, Imam Asrowardi, Eko Subyantoro and Luqman Muhammad Zagi. "First-Round and Last-Round Power Analysis Attack Against AES Devices". In: *ICITSI 2020*. 2020, pp. 410–415.

[Ram+18] Chethan Ramesh et al. "FPGA Side Channel Attacks without Physical Access". In: *FCCM 2018*. 2018, pp. 45–52.

[SB15] Mark Stanislav and Tod Beardsley. *HACKING IoT: A Case Study on Baby Monitor Exposures and Vulnerabilities*. [Online] https://media.kasperskycontenthub.com/. Sept. 2015. (Visited on 03/01/2023).

[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.

[ZS18] Mark Zhao and G. Edward Suh. "FPGA-Based Remote Power Side-Channel Attacks". In: *S&P 2018*. May 2018, pp. 229–244. DOI: 10.1109/SP.2018.00049.