SoK: Remote Power Analysis

Macarena Martínez-Rodríguez, Ignacio M. Delgado-Lozano, and Billy Bob Brumley

Tampere University, Tampere, Finland
macarena@imse-cnm.csic.es,ignacio.delgadolozano@tuni.fi

Abstract. In recent years, numerous attacks have appeared that aim to steal secret information from their victim, using the power side channel vector, without direct physical access and using instead, resources that are present inside the victim environment. These attacks are called *Remote Power Attacks* or *Remote Power Analysis*. However, there is no unified definition about the limitations that a power attack requires to be defined as remote. This paper aims to propose a unified definition and threat model to clearly differentiate remote power attacks from non-remote ones. Additionally, we collect the main remote power attacks performed so far from the literature, and the principal proposed countermeasures to avoid them. The search of such countermeasures denoted a clear gap in order to find technical details on how to prevent remote power attacks. Thus, the academic community must face an important challenge to avoid this emerging threat, given the clear room for improvement that should be addressed in terms of defense and security of devices that work with private information.

Keywords: hardware security; applied cryptography; side channel analysis; power analysis; remote power analysis; countermeasures

1 Introduction

Side Channel Analysis (SCA). Although there are plenty of cryptographic algorithms that are mathematically safe, because of their implementation in applications and devices, they can leak side-channel information by applying *Side Channel Analysis* (SCA) attacks. The main sources of leaked information are delay during key operations within the process of information encryption or decryption, leading to timing attacks [38], as well as power consumption [39] or electromagnetic radiation [56].

Remote hardware attacks. Gravellier et al. [28, Sect. 1] use the term *remote hardware attacks* or *software induced hardware attacks* to describe remote and semi-remote SCA attack vectors with the following characteristics. (i) Different from traditional SCA, they require no additional equipment for signal procurement outside system resources that are already available, and no proximity requirement since attackers communicate with the target over e.g. Ethernet

(feasibility). (ii) The root cause of the vectors lies in the hardware design; complete mitigations require redesign and (unlike software) patching fielded devices is rigid and costly at best (robustness). (iii) Due largely to the feasibility characteristic, attacks exploiting these vectors automate and deploy efficiently; again in contrast to traditional SCA, requiring a specialized procurement research environment per target device (scalability). Given these characteristics, localized [19, 20, 21] and far-field EM attacks [25, 13, 12] as well as acoustic attacks [22] do not qualify as remote hardware attacks since e.g. they utilize specialized procurement equipment to capture emanations at a reasonable distance (feasibility), and furthermore do not scale (scalability).

Remote power analysis. Building on these characteristics, we use the term *remote power analysis* to refer to a subset of remote hardware attacks where the additional following characteristic holds. (iv) The attack vector is passive and its underlying phenomena is a byproduct of transistor-level physics (physicality).

Passive vs. active is a gray area when it comes to software-assisted SCA, but our intention is to exclude software-induced fault attacks from this category. Hence, attack techniques such as RowHammer [36], CLKSCREW [61], FPGAhammer [41], VoltJockey [55], Plundervolt [50], V0LTpwn [35], etc. are not remote power analysis techniques since the underlying attack vectors are voltage-related software-induced faults. Emphasizing the physicality characteristic, traditional software-based microarchitecture attacks exploiting e.g. data cache contention [53, 52], instruction and last-level cache contention [1, 31, 64, 30], branch prediction [4, 3, 2], port contention [6, 9], etc., as well as transient execution such as Meltdown [46] and Spectre [37], are also not remote power analysis techniques since the underlying attack vectors are due to microarchitecture optimizations and not tied to e.g. power consumption. While the attack vectors mentioned in this paragraph are extremely interesting and impactful due to their semi-remote application, they remain out of scope for our study.

Contributions. The following bullet points summarize our main contributions.

1. We present a concrete definition and a unified threat model for remote power analysis research.
2. We collect and study the main remote power analysis techniques, paying special attention to the source of leakage that enables the attacks.
3. Likewise, we study countermeasures associated to the previous attacks.
4. We detect a considerable gap in the proposal of countermeasures, which are usually not technically detailed, and only proposed as future work that is still pending.
5. We identify gaps in defense-in-depth applications of remote power analysis, i.e. uses for good rather than evil in the security domain.

Outline. The organization of the paper is the following. Section 2 provides background on side-channels, power side-channels and remote power side-channels. Section 3 reviews existing work proposing remote side-channel attacks and Section 4 reviews the proposed countermeasures against them. Finally, we conclude in Section 5.

2 Background

As examples of timing attacks, Kocher [38] demonstrated that by measuring the amount of time required to perform different kinds of private key operations, it was possible to retrieve secret parameters from public key algorithms as, for example, fixed Diffie-Hellman exponent, factor RSA keys or other cryptosystems as DSA. Nine years later, Bernstein [8] performed a cache-timing attack that was able to achieve complete key recovery from AES, one of the most important symmetric-key crypto algorithms. He carefully details attacks that demonstrate the vulnerabilities of AES design using known-plaintext timings, due to the difficulties associated to write constant-time crypto algorithms with high requirements on speed. It is important to notice that this attack showed vulnerabilities on the design of the algorithm itself, and not only on the library used by a certain server.

Besides that, electromagnetic (EM) emanation is a leakage source that was discovered in a very primitive way by van Eck [16], when he was able to capture emanations from computer monitors that allowed to infer the information showed in the display. Several years later, Quisquater and Samyde [56] and Gandolfi et al. [17] produced the first works considering EM emanations for SCA while computing cryptographic operations. These attacks were precarious, as they required small antennas to be as close as possible to the circuit being attacked, which usually was a chip card. As a matter of fact, to succeed most of these attacks were slightly invasive, given the fact that they require a partial target decapsulation.

In 2002, these limitations were removed when Agrawal et al. [5] demonstrated that electromagnetic analysis (EMA) on cryptographic devices presented a real threat and they can be a source of leaked information to perform distance attacks and find an alternative way to attack devices when it is not possible to physically place a probe to measure the power consumption in a System-on-Chip (SoC) [32].

In this paper, we focus on remote power attacks. The rest of this section gives an introduction to classical power attacks, the threats that they represent, and how to face them with appropriate countermeasures.

2.1 Power Attacks

In 1999, Kocher et al. [39] showed for the first time that it is possible to find secret keys by tampering cryptographic devices that, from an algorithmic point of view are totally secure. They present the first *Simple Power Analysis* (SPA) methods to obtain secret parameters from the DES block cipher using uniquely one power trace. However, they focus their article on the usage of a high number of traces to retrieve the secret parameters leading to what is known as *Differential Power Analysis* (DPA). This method allow to substantially increase the probability of a successful attack due to the simplicity that it provides to reduce noise and measurement errors, at the same time highlighting dependencies between power and data or operations. Since the first appearance of these attacks, some of the most important block ciphers have been successfully attacked, such as AES and

others [54, 47, 60]. These attacks are often based on the correlation presented by the data involved in an operation and the dynamic power consumption using statistic models based on the Hamming Weight [47, 49] or Hamming Distance [10] in some input/output key points.

Furthermore, not only symmetric algorithms are targets for attacks based on power analysis as SPA or DPA. Kunihiro and Honda [42] demonstrated recovering secret parameters from the RSA algorithm using DPA analysis, and retrieving information from noisy analog data. Additionally, some other attacks as horizontal attacks [7] have appeared more recently to demonstrate that there is still space for new statistical techniques to break systems as RSA. DPA and horizontal attacks have also been applied to other asymmetric algorithms, for example elliptic curve cryptosystems [26, 34].

To avoid power attacks, there are three main groups of countermeasures, independent from the level of abstraction. These are: (i) Masking countermeasures, that try to execute additional random operations to mask and decorrelate operations and data from power consumption [14, 18, 33, 48]. These countermeasures were proved from the very beginning to not avoid the feasibility of a power attack, but only to delay the success [39]. Nonetheless, they are still used in real life since they can be practical in some scenarios. (ii) Blinding countermeasures in asymmetric cryptography that aim to prevent attackers to know or induce, analyzing the power consumption, cryptographic algorithm state that should remain secret [15, 11]. (iii) Hiding countermeasures, that could range from transistor-level to software implementations that seek to have a dynamic power consumption that is independent in every moment from the operations being carried out [44, 62, 63, 43].

2.2 Remote Power Attacks

Traditionally, power consumption has been captured physically on the devices using a probe. Recently, new ways to get this information leakage remotely have been explored. Nowadays, most of the systems have analogical and digital components. Mixed-signal components could leak information about the activity of the digital part. One of those components could be an analog-to-digital converter (ADC), which is an instrument that converts an analog signal, as a sound picked up by a microphone, into a digital signal. These ADCs can also measure and convert an input analog voltage or current to a digital number that indirectly represents the quantity of these magnitudes. This is normally a binary number directly proportional to the input. Gnad et al. [23] and O'Flynn and Dewar [51] use the ADC available in many systems as remote probe and, therefore, it is not necessary to have physical access to the platform to obtain power traces.

Voltage drop caused by the executed operation can be also captured by sensors implemented on the programmable logic as shown in [28, 24, 58, 59, 29, 27, 65, 57].

In [28, 24, 58, 29, 27, 65, 57], authors implement a sensor that measures the power side-channel leakage on the FPGA inside the chip. However, this leakage can also be captured from another chip included in the same board as exposed

in Schellenberg et al. [59]. In this case, the TDC of a malicious chip senses the voltage fluctuations caused by other chips on the same board.

Lipp et al. [45] monitored the values correlated with the power consumption using Intel Running Average Power Limit (RAPL) interface. Finally, Krautter et al. [40] present a countermeasure against on-chip voltage side-channel leakage in multi-tenant FPGAs based on mapping an active fence of ring-oscillators (ROs).

While the above provides a brief overview of this budding research field, we explore these works and more in a deeper fashion next in Section 3.

3 Attacks

3.1 ADC-based remote attacks

Gnad et al. [23] demonstrated that digital logic within mixed signal devices causes noise in the analog components, as ADCs or any kind of sensor. Both Gnad et al. [23] and O'Flynn and Dewar [51] capture the noise of ADC data while performing cryptographic operations, in the digital logic of different boards. Specifically, in Gnad et al. [23], they focus on leakage from AES and RSA, as shown in the second row of Table 1.

The adversarial model in [23], shown in Figure 1, considered that an attacker should have full or partial access to the ADC present in the board while the victim is running cryptographic code. This ADC must be read during the execution of cryptographic operations, and the access to the ADC data must be provided through another task in the system or through a webserver that hosts the sensor data. It is important to remark that, normally, an ADC can be read simultaneously to the execution of other operations by using a second core, Direct Memory Access (DMA), which is available in most microcontroller architectures.

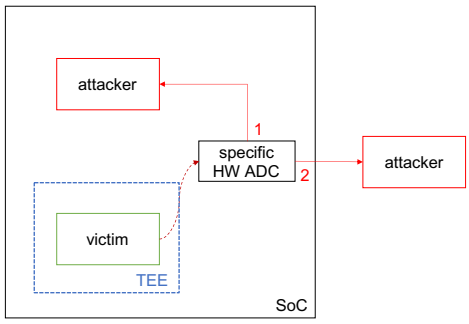

Fig. 1. Adversarial model where the victim leaks information to an ADC within the SoC, and the attacker task has access to the ADC data

Given these circumstances, the authors perform leakage assessment tests over the data captured by the ADC with several different configurations depending

on frequency, supply voltage of ADC, algorithms and boards used (see [23, Table 2]). In most of the configurations, leakage assessment experiments indicate leakage for both algorithms. The operation selected in the case of RSA is modular exponentiation, while the AES analysis showed that the last AES round is the source of the main leakage. It is remarkable the authors are able to retrieve critical leakage, for both algorithms, even if the ADC is not connected to any supply voltage. Further, *Correlation Power Analysis* (CPA) attacks are able to relate key bytes of AES to variations in the voltage measurement of ADC, proving that the leakage can be exploited. However, the authors do not perform an attack on RSA, even though they demonstrate that the execution of the modular exponentiation operation present in this algorithm leaks private information that can be used by a potential attacker.

With a similar adversarial model, O'Flynn and Dewar [51] obtain sufficient leakage from an on-board ADC being utilized to capture power traces while hardware encryption operations are taking place. The main difference in the adversary model lies in that the crypto algorithm implementation is inside a trusted execution environment (TEE) depicted in blue dashed line in Figure 1. Specifically, they use a SAML11 hardware AES accelerator which contains an M23 core with Trustzone-M that provides hardware-level isolation. Assuming an attacker that has first gained ability to execute code on the unsecure side of the device, they would potentially be able to trigger encryption operations and use the on-board ADC to capture power traces during such encryptions. This could ultimately lead to the revelation of secret parameters within a cryptographic algorithm.

As main results, with different external-aided circuit configurations, the authors mount a CPA attack over the S-Box input from the last round of AES-128, using a Hamming Weight model (third row in Table 1). With this attack, they retrieved all AES key bytes even with an important sample reduction on the ADC with respect to the main clock of the system. Particularly, with only one sample per 26 clock cycles, attacks are still successful and can be eased using an external amplifier to improve the quality of power traces.

3.2 TDC-based remote attacks

Another possibility to sense supply voltage fluctuations that could lead to leakage of useful information is applying time-digital converters (TDCs). TDC-based sensors convert propagation delay variations caused by power supply fluctuations into digital information that can be related to the secret state of cryptographic operations.

The functioning system of TDCs, shown in Figure 2, is the following: We have a clock signal, clk connected to the init delay block input, which is delayed to create a δclk. The difference between clk and δclk at the end of the delay line fluctuates with voltage variations. The init delay is set in such a way that the δclk is inside the delay line when the state is captured by the TDC register. What is saved by the register is the Hamming Weight of the stored value in each round, so this provides data about the supply voltage level and its fluctuations.

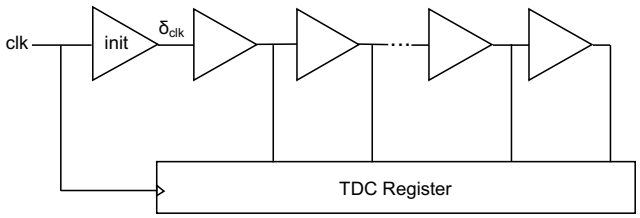

Fig. 2. TDC-based power sensor

If a voltage rise occurs, the propagation delay of the init block is reduced, so the δclk rising edge is faster and arrives further in the delay line, capturing more "1" values in the register and increasing the Hamming Weight. On the contrary, a voltage drop causes a higher number of "0", given the increase in the propagation delay, so the Hamming Weight will diminish.

Gravellier et al. [28, 29], Gnad et al. [24], and Schellenberg et al. [58, 59] use this mechanism to attack different implementations.

Specifically, Gravellier et al. [28] target a hardware AES implementation on FPGA and two software implementations, an 8-Bit Tiny AES and another within the OpenSSL library. The target board is a Xilinx Zynq 7000 which implements a Xilinx Artix-7 FPGA and an ARM Cortex-A9 CPU as shown in the third row of Table 1.

The threat model in [28] assumes a cloud scenario in which FPGA-based voltage sensors can be maliciously implemented through cloud FPGA rental, untrusted IP insertion or bitstream reconfiguration. Additionally, given the current SoC context, these sensors could be part of FPGA modules that are inserted in the same die as a CPU. That is the reason why the authors consider both hardware and software implementations of AES (see Figure 3).

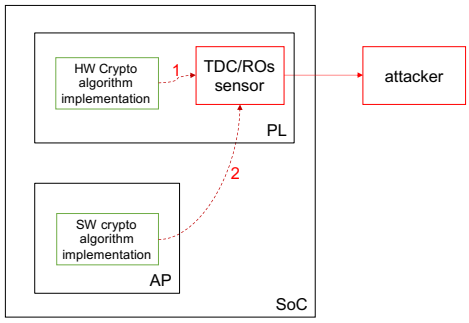

Fig. 3. Adversarial model where the attacker can implement a TDC/ROs sensor on the PL part of a SoC and the crypto algorithm leaks information if implemented on the PL part or run on the AP within a SoC

Gravellier et al. [28] perform a CPA attack on different operations of the algorithm depending on the implementation. They are able to retrieve the AES key from the hardware implementation with just over a thousand traces. Moreover, they obtain similar results for a wide variety of boards in [24], where the same sensor is implemented over the same adversarial scenario; see Table 1, rows 4–5. For the software cases, they require around 100,000 traces to break this implementation, the first ever based on FPGA sensors and targeting software cryptographic implementations. To end, the authors compare the obtained results with traditional EMA. In the OpenSSL case, the amount of traces needed is roughly the same for both conventional EMA and emerging ones based on FPGA sensors, concluding that they represent a real threat to security, considering malicious co-location in cloud scenarios which can lead to remote attacks. In the case of Tiny AES, the authors require around 50,000 traces using the electromagnetic vector to retrieve the secret key, so the emerging attacks have in this case room for improvement.

Schellenberg et al. [58] attack a hardware AES module using TDC-based sensors considering two adversarial scenarios similar to the one given in [28]. This is, a first scenario in which a malicious user has partial access to an FPGA shared by several users (label 1 in Figure 3 denotes this scenario) and another one where the attacker has full access to an FPGA which is part of a SoC where a CPU resides on the same die (label 2 in Figure 3). The authors build what they called a "Hardware Power Distribution Network (PDN) Trojan" formed by TDC-based sensors placed on a main FPGA which runs an AES hardware implementation mounted over a SAKURA-G board that contains another FPGA (control FPGA) that generates and sends random plaintexts to the main FPGA. They perform experiments varying sampling frequency, and initial delay of TDCs configurations that lead to a successful CPA attack to retrieve the AES key with less than 5,000 traces (row 6, Table 1). As an important contribution, they consider the placement of the sensors in two different positions inside the FPGA where the AES algorithm is implemented. One of them is placed as near as possible to the AES module while the other one is placed as far as possible. The results show that the attack is still possible with only a slight decrease in the correlation, which implies that placing sensors inside an FPGA running cryptographic operations can always represent a threat to security, even if some measures are taken to isolate in a logic level the cryptographic module from other modules present in a certain FPGA.

With similar implementation compared to [58], Schellenberg et al. [59] use two different SoC for the victim and for the attacker that share the PDN as shown in Figure 4. That is, they use the smaller FPGA that lies in a SAKURA-G board to implement an AES and an RSA hardware module, while the larger one uses TDCs sensor to perform attacks over both algorithms. To attack the hardware AES module, they perform a CPA attack varying sampling rates from 24 MHz to 96 MHz and with different capacitor configurations inside the board. Concrete configurations could benefit the attack, being successful with 20,000 traces, but some others lead to harder efforts in order to have successful key re-

covery. In the worst case scenario, Schellenberg et al. [59] need 2.5 million traces to achieve a successful attack. Although this could seem like a huge amount of power traces, it represents only 38.5 MB of encrypted data, which is quite manageable regarding memory and other computing resources. Moreover, the authors conduct an SPA attack by capturing power traces from the binary exponentiation during RSA decryption. After applying a 900 kHz low-pass filter, it is easy to differentiate if the state of the multiplication module has changed in each individual step of the binary exponentiation. This allows to differentiate the steps in which a square and multiply operation is performed from those where only the squaring operation takes place. This, ultimately, leads to the recovery of the secret exponent (row 7, Table 1).

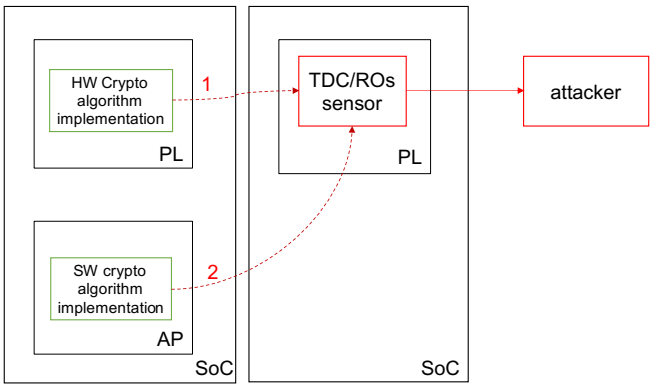

Fig. 4. Adversarial model where the attacker can implement a TDC/ROs sensor on the PL part of a SoC and the crypto algorithm leaks information if implemented on the PL part or run on the AP in another SoC that shares the PDN

To end, Gravellier et al. [29] present several CPA attacks performed over two different ARM CPUs (ARM Cortex-A9 and ARM Cortex-A7). In this case, the authors take into consideration three attacks conducted using different methodologies and varying from bare metal to algorithms running over an OS. The authors consider that an attacker should have access to the delay line based sensors present in each core. For the first attack, authors consider that both attacker and victim are running their respective binaries on bare metal, each on a distinct core of the Cortex-A9 application processor (AP). The attacker code collects the AES leakage data by configuring the access to the delay-locked loop (DLL) main register that enables the possibility to sample its values while the core that is running AES is performing encryptions (row 8, Table 1). The attacker core also provides the plaintexts ciphered by the victim, triggers the encryption and readback of DLL states. This DLL can be considered as a variation of a delay line, similar to that included in TDCs, and is used in this paper to track variations in temperature. Specifically, the authors use a cooling spray to

cool down the SoC package to demonstrate that each time a spray shot induces a temperature drop, it is possible to observe a DLL command drop, which means that a DLL is suitable to dynamically track the SoC temperature variations. Authors use a DMA to improve the sampling rate and synchronization at the moment of capturing the traces. Using this method, authors are able to retrieve the AES key after 20 million traces.

For the two remaining attacks, they use the ARM-Cortex A7 SoC with an OpenSSL AES implementation setup changing the DLL by delay-lines that act as TDCs. In one case, the attacker is running a bare metal binary (microcontroller unit, MCU) while the victim is using a Linux OS (AP), while the opposite configuration (attacker running in Linux OS and victim running in bare metal) applies for the second case. 40 million traces are needed when the victim runs over the OS, while only 10 million are sufficient when the encryption is produced over bare metal. Normally, the attacks performed while the victim process is taking place on bare metal retrieve the key using a lower number of traces, since there are no interruptions related to the OS that may disturb the attack and victim processes causing synchronization issues. As a summary, they require between 9 and 24 hours to achieve a successful attack for the three cases, using different DMA and target frequencies. The main reason for having such a large difference in the amount of traces needed to recover the correct key is that both users are running on different cores, and not using FPGAs in any case. Additionally, the DMA frequency in the best of the cases has a sampling rate 10 times lower compared to that of the target. This limited sampling contributes to the high number of acquisitions required to recover key values.

3.3 Ring oscillators-based remote attacks

Finally, the other main way to exploit leakages related to remote attacks is to take advantage of ring oscillators (RO) based voltage sensors within systems that implement FPGAs. Ring oscillators (Figure 5) are components composed of an odd number of inverters, whose output oscillates between two voltage levels attached in a chain and the output of the last inverter is fed back into the first one. Since we have an odd numbers of inverters, the last output of the chain will be the opposite of the first input. The final output is established a certain time after the first input is introduced, and the feedback of the last output to first input generates oscillations. Using FPGA modules to appropriately place ROs in systems where the PDN is shared by the logic modules among them and also with the CPU module in case the device incorporates it, could lead to critical information leakage. Supply voltage fluctuations can be measured using RO-based sensors, which can be used to retrieve private information from cryptographic operations running both in hardware or software inside the system.

Gravellier et al. [27] consider a multi-user FPGA cloud scenario where RO-based sensors are enabled to perform nanosecond scale measurement of the FPGA internal voltages. In a time-shared system, users are logically isolated and the kernel controls all communication among them. However, malicious users can query RO-based sensors to sense supply voltage variations in the PDN of the

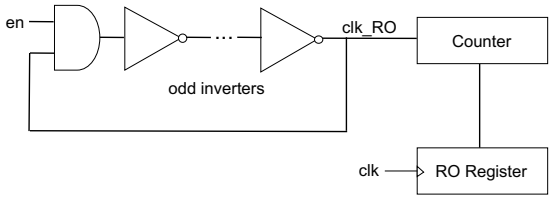

Fig. 5. Ring-oscillator-based power sensor

FPGA that allows to steal information about an honest user (Figure 3). The attack model consists of targeting a state register that stores data coming from each round transformation of AES. The 128-bit register is refreshed at the end of each round and generates an important fluctuation in the supply voltage level. The authors exploit this leakage in order to perform a successful CPA attack using a Hamming Distance model between two consecutive states of the register. In this case they attacked the last round of the algorithm in a known ciphertext attack model.

Gravellier et al. [27] use the RO-based sensors to take measurements from the internal voltage while an AES hardware module is running at 50 MHz on a Xilinx Zynq 7000 board (row 9, Table 1). They also performed attack experiments using different frequencies obtaining similar results. The place and route of the design is made specifically to have the sensor instances as far as possible from the hardware modules. Even when these measures are taken into consideration, the authors are able to retrieve the AES secret key using less than 100,000 traces. Several configurations varying the number or RO-based sensors involved demonstrated that using 64 ROs, only 8,000 traces are needed to retrieve the correct AES key. Moreover, the authors conducted a CPA attack using TDC-based sensors and a traditional EM side-channel in order to conclude that on-chip sensors offer very similar results when they are compared to traditional EM attacks. Although the RO-based sensors do not reach the accuracy of TDC-based sensors, the results are tied thanks to a higher proximity of these sensors. Additionally, they offer a lighter area overhead and an easier implementation since they are composed of basic logic gates.

Zhao and Suh [65] performed two attacks using SPA over the modular exponentiation on a hardware implementation of RSA, monitoring power consumption thanks to RO-based sensors. They consider two attacks. The first uses an FPGA shared among multiple users, where a malicious user implements an attack circuit on one part of the FPGA aiming to steal secret information from the victim's circuit present in the same FPGA. The second one consists of an FPGA-to-CPU attack where an attack circuit on the FPGA targets a CPU present in the same SoC, sharing the same PDN. In particular, the square and multiply algorithm is the target of their attacks, since if the bit exponent is 1, the multipliers will perform sequences of additions leading to a high switching activity in FFs and LUTs, while if the exponent is 0, only the squaring multiplier's logic will generate a switching activity, leading to a lower power consumption. [65,

Figs. 10,11] demonstrate this hypothesis and the SPA attack is successful and able to retrieve the correct keys, using three different configurations depending on the placement and route of attacker's and victim's modules. These range from physical isolation between ROs-based sensors and RSA module which is the more difficult configuration for the attacker, to a specific place and route that is selected by the attacker to benefit him passing through another configuration where the placement and route is not specific. Additionally, the second attack uses power traces to perform an FPGA-to-CPU attack. Nonetheless, these traces are used to enable timing attacks on software programs since the power consumption reveals the start and end of internal program operations, so they act in practice as triggers that delimit the operations, even if some masking countermeasures are taken into consideration, as introducing delay in outputs. This attack is out of the scope for our survey paper since, although it uses power traces to perform a remote attack over a CPU, it reveals the secret RSA key by analyzing timing differences instead of power consumption ones.

To end, Ramesh et al. [57] use ROs in a different way to register leakage coming from long wires maliciously placed in a multi-tenant FPGA environment (Figure 3). Their key insight is that the logic value carried on a long wire influences the delay of another long wire close to the first. This way, when a logic "1" value is carried on one wire, that we can denote as transmitter, the delay in the neighbor, which can be denoted as receiver is lower relative to a case when a logic "0" is transmitted. In a first experiment, they develop a setup that consists of a test pattern generator that assigns either a logic "1" or "0" to the transmitter long wire, while the receiver is implemented as a three-stage RO with one inverter and two buffers, and one of its wire adjacent to the transmitter. Then, with a counter and evaluating differences in RO frequencies, they are able to compute the count difference of the receiver, and classify the logic value that carries the transmitter based on its value. After this preliminary experiment, the same is applied to a hardware 128-bit AES implementation. The attack extracts a single byte of the round key in the final round of AES by using the measured counts of an RO that targets a specific selected wire in the design of AES. This is repeated for every byte and the encryption key can be calculated by inverting the key schedule (row 11, Table 1).

The authors conduct further experiments to check the importance of wire lengths, clock frequencies and constraints of placement and route for their attack. Results show that as the length of the victim wire increases, the attack is easier due to a higher coupling effect that leads to a larger side channel signal. Besides that, a higher operating frequency hinders the attack since the sampling rate that ring oscillators can provide does not vary, and the number of samples that are able to capture per operating period is lower. Additionally, the authors carry out an experiment to check the difference that an automatic place-and-route could present relative to a manual one. The attack is still successful with the automatic configuration and the main differences are given by the operating frequency ranging from 217 measurements to disclose (MTD) the correct

key with an operating frequency of 10 kHz and 1.5 million MTD for a 4 MHz frequency.

3.4 Other attacks

Apart from the main sensors that are placed on-chip and can detect leakage, there are software alternatives to detect and measure power variations that allow to carry out successful attacks. Intel Running Average Power Limit (RAPL) is an interface present in Intel processors that allows to control core frequency and voltage as well as directly monitor the power consumption data of socket and memory domain. Lipp et al. [45] use this tool to distinguish instructions, operands and data from the Linux kernel and SGX enclaves that allow to access this interface with a range of different access privileges. After a preliminary study where they are able to perform different experiments over laptop, desktop and server CPUs, this tool allows key recovery from a software mbed TLS RSA implementation using an SGX enclave. Finally, they perform CPA attacks to extract AES keys both from the Linux kernel and the SGX enclave, even utilizing AES-NI native instructions.

In this case, the "sensor" that detects the leakage and enables the possibility to a side channel attack is the own vendor of the processor that provides a high resolution probe in the form of a software interface that can jeopardize every secret information processed by the CPU, which causes evident security flaws. This is another level of security threat, because for previous cases, attackers need to learn to use components present in SoCs as side-channel leakage vectors/sensors, and explicitly be granted access to these components. But in this case, vendors provide a mature application that allow attackers to simply read power values with high resolution, easing enormously the attack.

3.5 Summary

Table 1 collects the main information we surveyed concerning works that performed remote power analysis. Source of leakage, system and algorithm targeted, and type of attack performed are gathered in this table to have a general overview of differences and similarities among the several researches present in the recent literature.

4 Countermeasures

4.1 Countermeasures to ADC-based remote attacks

Gnad et al. [23] expose that noise of analog components as ADCs should not be considered as regular noise margin, but instead, to be treated as possible information leakage in digital components running cryptographic code within the system. They propose several countermeasures ranging from very restrictive ones to more flexible.

Table 1. Overview of works that carry out remote power side-channel attacks

	Source	Target	Algorithm	Attack
Gnad et al. [23]	ADC	Different boards	SW mbedTLS RSA	TVLA
			SW mbedTLS AES	TVLA,CPA
O'Flynn and Dewar [51]	ADC	ARM Cortex-M23	specific HW AES	TVLA,CPA
Gravellier et al. [28]	TDCs	Xilinx Artix-7	HW AES module	CPA
		ARM Cortex-A9	SW Tiny AES	CPA
		ARM Cortex-A9	SW OpenSSL AES	CPA
Gnad et al. [24]	TDCs	Different boards	HW AES module	CPA
Schellenberg et al. [58]	TDCs	Xilinx Spartan 6	HW AES module	CPA
Schellenberg et al. [59]	TDCs	Xilinx Spartan-6	HW RSA module	SPA
			HW AES module	CPA
Gravellier et al. [29]	TDCs	ARM Cortex-A9	SW OpenSSL AES	CPA
		ARM Cortex-A7	SW OpenSSL AES	CPA
Gravellier et al. [27]	ROs	Xilinx Artix-7	HW AES module	CPA
Zhao and Suh [65]	ROs	Xilinx Artix-7	HW RSA module	SPA
Ramesh et al. [57]	ROs	Different boards	HW AES module	CPA
Lipp et al. [45]	Intel RAPL interface	Linux kernel	SW AES-NI	CPA
		SGX enclave	SW AES-NI	CPA
		SGX enclave	SW mbedTLS RSA	SPA

In a first approach, authors aim to guarantee that measurements cannot be taken when security-related computations are running. However, since this is a very restrictive proposal, they consider to perform leakage assessment on every measurement data that could be exploited by attackers. If the analysis reveals the data could be a source of leakage, those measurements should be treated with the same security level as the secret data processed by the cryptographic code.

Finally, if any of those cannot be achieved, the authors propose to filter the ADC data in a way that hinders exploiting leakage to find secrets in the system. In any case, the countermeasures proposed by Gnad et al. [23] focus on avoiding that underprivileged tasks could take ADC measurements with impunity, while cryptographic operations are taking place in other parts of the system, since this could lead to power analysis that potentially reveals secret information.

O'Flynn and Dewar [51] give similar proposals. In its countermeasures section, their first proposal is to move the ADC to the secure world of the M23 core, but it is applicable to any device with a logical separation between secure and non-secure internal components. As well, authors state this is valid for other peripherals that can provide side channel leakage similar to the one provided by the ADC.

As an alternative, O'Flynn and Dewar [51] present the possibility of validating peripherals before starting a critical operation inside the secure world. For instance, if prior to the beginning of an encryption, the TEE detects that the ADC is enabled to take measurements when it should not normally be, the encryption could be suspended and the device could try to disable the ADC before performing another attempt to run the encryption operation.

To end, since normally remote attacks based on ADCs require a lot of traces in order to mount a CPA (or DPA) attack, they propose protocol-level restrictions. For instance, establishing a limit to the number of times an encryption operation is performed with a single key could be a practical solution to avoid successful attacks, even if they are not banned or avoided by design.

4.2 Countermeasures to TDC-based remote attacks

The countermeasures proposed by Gravellier et al. [28] fundamentally consist of having an independent power supply for each FPGA chip in a cloud scenario. However, the authors recognize that this looks difficult in SoCs where a CPU and FPGA lie together in the same die, since the division of power supply would increase design costs. Additionally, they consider that the usage of classical countermeasures based on masking and shuffling should never be dismissed, since they can impede, if not avoid, TDC-based remote attacks.

On the other hand, Gnad et al. [24] considered that any traditional countermeasure against side-channel attacks, or fault injection attacks (since they use ring oscillators to generate faults) should always be considered. They also propose bitstream checking in order to detect malicious designs that can intentionally cause abrupt voltage drops, before they are loaded to the FPGA. For that, they need to formulate the basic circuits properties required to sense voltage fluctuations or cause faults, requiring a certain knowledge about the related influence between the voltage drops and the design over the FPGA, which can be obtained by analyzing the netlist of the design. Additionally, they propose an electrical isolation by active fencing between the potential victim and the attacker. They focused their effort on obtaining an on-chip PDN that mitigates the effect that one module implemented on the FPGA (attacker) can cause to another module implemented also on the FPGA (victim) where a crypto algorithm could be running.

Gravellier et al. [29] propose four countermeasures. (i) Add software randomization that efficiently desynchronizes computations, hindering trace alignment in order to perform a CPA attack. On the monitoring side, adding phase and frequency jitter to the clock signal used for sampling the delay-line registers would mitigate a possible attack. Note these are traditional SCA countermeasures. (ii) Restricting the access to the delay-line registers by unauthorized users. An example to carry out this countermeasure is to place delay-lines in the secure world and prevent the access to any user or application present in the non-secure world. (iii) Reduce the delay-line sampling rate, for instance, by limiting the access rate to the registers that store delay-line information. (iv) Finally, the most drastic solution they propose is to remove delay-lines from SoC altogether. However,

this seems an unfeasible measure in the short term because manufacturers and vendors will not suspend their product development pipeline, and these elements are likely used by other applications with functional (non-malicious) purposes.

4.3 Countermeasures to Ring oscillators-based remote attacks

Gravellier et al. [27] are pessimistic about possible countermeasures against on-chip sensors in the case of RO-based ones. Since they show that isolation among logic blocks is ineffective against power side-channel attacks in a multi-user cloud scenario, they only consider as a solution forbidding the RO implementation by restricting place and route designs. However, this is not possible because many of the main applications of FPGAs requires the usage of different types of sensors. Finally, they consider trojan detection routines as a temporary solution, but a huge development effort will be required according to the authors. Furthermore, attackers can adapt in an attempt to bypass the new security restrictions.

Zhao and Suh [65], once again propose classical countermeasures, as adding dummy operations or masking power consumption by randomizing intermediate values. However, the authors recognize that these countermeasures have an associated overhead in terms of performance and energy, and that other hardware countermeasures as the use of dynamic logic styles is not possible in FPGA environments. Similarly to the proposal by Gnad et al. [24], they consider to prevent malicious designs by checking FPGA designs before placing the logic onto a physical FPGA, and disallow designs with FPGA-based monitors by analyzing the design netlist. However, this is difficult to establish since there are legitimate uses for ROs, e.g. the design of physically unclonable functions (PUFs) that generate secret unique keys. Moreover, detailed analyses of netlists is time-consuming, and even impractical for encrypted bitstreams. The authors acknowledge the need for a non-trivial solution.

Paradoxically, Krautter et al. [40] use ROs based sensors to fence a cryptographic module with different operation modes. For the first operation mode, these ROs based sensors are activated randomly using pseudo-random number generators (PRNGs). Their mere usage increases the noise inside the system, and therefore, it reduces the signal-to-noise ratio (SNR) and masks the voltage fluctuations that the cryptographic modules could generate. This hinders the voltage measurement not only when the attacker uses ROs based sensor, but also if the attacker uses other sensors like ADCs. The second operation mode aims to activate an exact amount of ROs depending on the voltage fluctuation that the cryptographic module is generating, to compensate and flatten in the victim side those fluctuations, in turn reducing leakage that the attackers can use to perform their attack.

4.4 Countermeasures to other remote attacks

To solve the problems related to the Intel RAPL interface that allow to use this software tool as a power monitor to perform side-channel attacks, Lipp et al. [45] propose to restrict the access to unprivileged users from getting data from

this application. Additionally, another possible solution could be to limit the resolution of data, in such a way they still give valuable information about the inner power, frequency and voltage state of the system but making impossible an statistical analysis that could lead to a successful power attack. Intel plans to release updates that prevent the vulnerabilities present in Lipp et al. [45]. They aim to avoid to distinguish the same instructions with different data or operands when SGX enclaves are used. Additionally, an update made over the Linux kernel restricts the access of unprivileged users to model-specific registers.

As a general rule, we noticed that authors tend to remotely attack different implementations that vary in many distinct levels (algorithm, hardware or software, type of attack) providing a vast quantity of technical details on how to mount and make functional attacks. However, with the exception of Krautter et al. [40], there is no deep study and evaluation of which countermeasures could be applied with the same level of details, and all the articles we surveyed only offer discursive potential solutions that are not elaborated and tested in a practical way. Thus, it is difficult to obtain a clear understanding about how feasible they are in the real world. Even more, Ramesh et al. [57] and Schellenberg et al. [58, 59] propose no countermeasures at all.

Table 2 summarizes different countermeasures among different proposed works in state-of-the-art.

Table 2. Overview of proposed countermeasures

	Leakage Analysis	Limiting Resolution / Filter data	Access restriction to unprivileged users	Prior checking of sensors implementation/design	Limit the usage of the same key to avoid CPA	Power Isolation	Classical Countermeasures	Removal of sensors	On-board design restriction	Active fences to compensate voltage fluctuations
Gnad et al. [23]	✓	✓	✓							
O'Flynn and Dewar [51]			✓	✓	✓					
Gravellier et al. [28]						✓	✓			
Gnad et al. [24]				✓			✓			✓
Gravellier et al. [29]		✓	✓				✓	✓		
Gravellier et al. [27]									✓	
Zhao and Suh [65]				✓			✓			
Krautter et al. [40]										✓
Lipp et al. [45]		✓	✓							

5 Conclusion

To conclude, we detail the main contributions and important aspects of this paper. First, we present a unified definition of the concept of *Remote Power Analysis*, since many articles recently used this term to denote attacks where the adversary does not need physical access or contact with the victim, but without clearly stating which are the limits to consider what is a remote attack. Here, it is defined as a passive attack that seizes leakage, using a hardware component inside the design attacked due to transistor-level physics in a direct or indirect way. With this definition, we distinguish and collected the main articles that performed Remote Power Analysis over their victims, classifying them by their source of leakage and the distinct sensors that are able to capture it. Likewise, we provided a classification of the countermeasures proposed by the authors of these attacks. We close with some insightful observations revealed by our taxonomies.

Rigorous countermeasures. In most of the cases, the surveyed countermeasures are not technically detailed by the authors, but rather deferred as retrospective future work that has yet to materialize. Thus, there is clearly room for improvement to determine which is the root cause for why remote attacks work, and to establish the best way to holistically prevent them or at least hedge against them and provide reasonable trade-offs.

Constructive applications. Finally, we have identified that there is a lack of research proposals that exploit the remote power side-channel for defense-in-depth applications. That is, to use the remote power side-channel for good side instead of evil side. While it is clear we have technologies that utilize these sensors as fundamental building blocks in ICs, such as PUFs and RNGs, these are dedicated, single-purpose designs. The previously discussed countermeasure work by Krautter et al. [40] is in this vein. Hence, we view this as an open research challenge—finding clever ways to utilize these technologies to improve platform security in a flexible and broadly-applicable way.

Acknowledgments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 804476).

Supported in part by the Cybersecurity Research Award granted by the Technology Innovation Institute (TII).

Supported in part by CSIC's i-LINK+ 2019 "Advancing in cybersecurity technologies" (Ref. LINKA20216).

References

1. Acıiçmez, O., Brumley, B.B., Grabher, P.: New results on instruction cache attacks. In: Mangard, S., Standaert, F. (eds.) Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6225, pp. 110–124. Springer (2010), https://doi.org/10.1007/978-3-642-15031-9_8

2. Acıiçmez, O., Gueron, S., Seifert, J.: New branch prediction vulnerabilities in OpenSSL and necessary software countermeasures. In: Galbraith, S.D. (ed.) Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4887, pp. 185–203. Springer (2007), https://doi.org/10.1007/978-3-540-77272-9_12
3. Acıiçmez, O., Koç, Ç.K., Seifert, J.: On the power of simple branch prediction analysis. In: Bao, F., Miller, S. (eds.) Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, AsiaCCS 2007, Singapore, March 20-22, 2007. pp. 312–320. ACM (2007), http://doi.acm.org/10.1145/1229285.1266999
4. Acıiçmez, O., Koç, Ç.K., Seifert, J.: Predicting secret keys via branch prediction. In: Abe, M. (ed.) Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4377, pp. 225–242. Springer (2007), https://doi.org/10.1007/11967668_15
5. Agrawal, D., Archambeault, B., Rao, J.R., Rohatgi, P.: The EM side-channel(s). In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 29–45. Springer (2002), https://doi.org/10.1007/3-540-36400-5_4
6. Aldaya, A.C., Brumley, B.B., ul Hassan, S., Pereida García, C., Tuveri, N.: Port contention for fun and profit. In: 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019. pp. 870–887. IEEE (2019), https://doi.org/10.1109/SP.2019.00066
7. Bauer, A., Jaulmes, É., Prouff, E., Wild, J.: Horizontal and vertical side-channel attacks against secure RSA implementations. In: Dawson, E. (ed.) Topics in Cryptology - CT-RSA 2013 - The Cryptographers' Track at the RSA Conference 2013, San Francisco,CA, USA, February 25-March 1, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7779, pp. 1–17. Springer (2013), https://doi.org/10.1007/978-3-642-36095-4_1
8. Bernstein, D.J.: Cache-timing attacks on AES (2005), http://cr.yp.to/papers.html#cachetiming
9. Bhattacharyya, A., Sandulescu, A., Neugschwandtner, M., Sorniotti, A., Falsafi, B., Payer, M., Kurmus, A.: SMoTherSpectre: Exploiting speculative execution through port contention. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019. pp. 785–800. ACM (2019), https://doi.org/10.1145/3319535.3363194
10. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings. Lecture Notes in Computer Science, vol. 3156, pp. 16–29. Springer (2004), https://doi.org/10.1007/978-3-540-28632-5_2
11. Brumley, D., Boneh, D.: Remote timing attacks are practical. In: Proceedings of the 12th USENIX Security Symposium, Washington, D.C., USA, August 4-8, 2003. USENIX Association (2003), https://www.usenix.org/conference/12th-usenix-security-symposium/remote-timing-attacks-are-practical
12. Camurati, G., Francillon, A., Standaert, F.: Understanding screaming channels: From a detailed analysis to improved attacks. IACR Trans. Cryptogr. Hardw.

Embed. Syst. 2020(3), 358–401 (2020), https://doi.org/10.13154/tches.v2020.i3.358-401

13. Camurati, G., Poeplau, S., Muench, M., Hayes, T., Francillon, A.: Screaming channels: When electromagnetic side channels meet radio transceivers. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018. pp. 163–177. ACM (2018), https://doi.org/10.1145/3243734.3243802
14. Cnudde, T.D., Nikova, S.: Securing the PRESENT block cipher against combined side-channel analysis and fault attacks. IEEE Trans. Very Large Scale Integr. Syst. 25(12), 3291–3301 (2017), https://doi.org/10.1109/TVLSI.2017.2713483
15. Coron, J.: Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1717, pp. 292–302. Springer (1999), https://doi.org/10.1007/3-540-48059-5_25
16. van Eck, W.: Electromagnetic radiation from video display units: An eavesdropping risk? Comput. Secur. 4(4), 269–286 (1985), https://doi.org/10.1016/0167-4048(85)90046-X
17. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2162, pp. 251–261. Springer (2001), https://doi.org/10.1007/3-540-44709-1_21
18. Gebotys, C.H.: A table masking countermeasure for low-energy secure embedded systems. IEEE Trans. Very Large Scale Integr. Syst. 14(7), 740–753 (2006), https://doi.org/10.1109/TVLSI.2006.878344
19. Genkin, D., Pachmanov, L., Pipman, I., Tromer, E.: Stealing keys from PCs using a radio: Cheap electromagnetic attacks on windowed exponentiation. In: Güneysu, T., Handschuh, H. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9293, pp. 207–228. Springer (2015), https://doi.org/10.1007/978-3-662-48324-4_11
20. Genkin, D., Pachmanov, L., Pipman, I., Tromer, E.: ECDH key-extraction via low-bandwidth electromagnetic attacks on PCs. In: Sako, K. (ed.) Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9610, pp. 219–235. Springer (2016), https://doi.org/10.1007/978-3-319-29485-8_13
21. Genkin, D., Pachmanov, L., Pipman, I., Tromer, E., Yarom, Y.: ECDSA key extraction from mobile devices via nonintrusive physical side channels. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016. pp. 1626–1638. ACM (2016), http://doi.acm.org/10.1145/2976749.2978353
22. Genkin, D., Shamir, A., Tromer, E.: RSA key extraction via low-bandwidth acoustic cryptanalysis. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol.

8616, pp. 444–461. Springer (2014), https://doi.org/10.1007/978-3-662-44371-2_25
23. Gnad, D.R.E., Krautter, J., Tahoori, M.B.: Leaky noise: New side-channel attack vectors in mixed-signal IoT devices. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019(3), 305–339 (2019), https://doi.org/10.13154/tches.v2019.i3.305-339
24. Gnad, D.R.E., Krautter, J., Tahoori, M.B., Schellenberg, F., Moradi, A.: Remote electrical-level security threats to multi-tenant FPGAs. IEEE Des. Test 37(2), 111–119 (2020), https://doi.org/10.1109/MDAT.2020.2968248
25. Goller, G., Sigl, G.: Side channel attacks on smartphones and embedded devices using standard radio equipment. In: Mangard, S., Poschmann, A.Y. (eds.) Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers. Lecture Notes in Computer Science, vol. 9064, pp. 255–270. Springer (2015), https://doi.org/10.1007/978-3-319-21476-4_17
26. Goubin, L.: A refined power-analysis attack on elliptic curve cryptosystems. In: Desmedt, Y. (ed.) Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2567, pp. 199–210. Springer (2003), https://doi.org/10.1007/3-540-36288-6_15
27. Gravellier, J., Dutertre, J., Teglia, Y., Loubet-Moundi, P.: High-speed ring oscillator based sensors for remote side-channel attacks on FPGAs. In: Andrews, D., Cumplido, R., Feregrino, C., Platzner, M. (eds.) 2019 International Conference on ReConFigurable Computing and FPGAs, ReConFig 2019, Cancun, Mexico, December 9-11, 2019. pp. 1–8. IEEE (2019), https://doi.org/10.1109/ReConFig48160.2019.8994789
28. Gravellier, J., Dutertre, J., Teglia, Y., Loubet-Moundi, P., Olivier, F.: Remote side-channel attacks on heterogeneous SoC. In: Belaïd, S., Güneysu, T. (eds.) Smart Card Research and Advanced Applications - 18th International Conference, CARDIS 2019, Prague, Czech Republic, November 11-13, 2019, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11833, pp. 109–125. Springer (2019), https://doi.org/10.1007/978-3-030-42068-0_7
29. Gravellier, J., Dutertre, J.M., Teglia, Y., Moundi, P.L.: SideLine: How delay-lines (may) leak secrets from your SoC (2020), https://arxiv.org/abs/2009.07773
30. Gruss, D., Maurice, C., Wagner, K., Mangard, S.: Flush+flush: A fast and stealthy cache attack. In: Caballero, J., Zurutuza, U., Rodríguez, R.J. (eds.) Detection of Intrusions and Malware, and Vulnerability Assessment - 13th International Conference, DIMVA 2016, San Sebastián, Spain, July 7-8, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9721, pp. 279–299. Springer (2016), https://doi.org/10.1007/978-3-319-40667-1_14
31. Gullasch, D., Bangerter, E., Krenn, S.: Cache games - bringing access-based cache attacks on AES to practice. In: 32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA. pp. 490–505. IEEE Computer Society (2011), https://doi.org/10.1109/SP.2011.22
32. Hayashi, Y.I., Homma, N., Mizuki, T., Aoki, T., Sone, H., Sauvage, L., Danger, J.L.: Analysis of electromagnetic information leakage from cryptographic devices with different physical structures. IEEE Transactions on Electromagnetic Compatibility 55(3), 571–580 (Jun 2013), https://doi.org/10.1109/TEMC.2012.2227486
33. Ishai, Y., Sahai, A., Wagner, D.A.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, Au-

gust 17-21, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2729, pp. 463–481. Springer (2003), https://doi.org/10.1007/978-3-540-45146-4_27
34. Kabin, I., Dyka, Z., Klann, D., Mentens, N., Batina, L., Langendörfer, P.: Breaking a fully balanced ASIC coprocessor implementing complete addition formulas on Weierstrass elliptic curves. In: 23rd Euromicro Conference on Digital System Design, DSD 2020, Kranj, Slovenia, August 26-28, 2020. pp. 270–276. IEEE (2020), https://doi.org/10.1109/DSD51259.2020.00051
35. Kenjar, Z., Frassetto, T., Gens, D., Franz, M., Sadeghi, A.: V0LTpwn: Attacking x86 processor integrity from software. In: Capkun, S., Roesner, F. (eds.) 29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020. pp. 1445–1461. USENIX Association (2020), https://www.usenix.org/conference/usenixsecurity20/presentation/kenjar
36. Kim, Y., Daly, R., Kim, J.S., Fallin, C., Lee, J., Lee, D., Wilkerson, C., Lai, K., Mutlu, O.: Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In: ACM/IEEE 41st International Symposium on Computer Architecture, ISCA 2014, Minneapolis, MN, USA, June 14-18, 2014. pp. 361–372. IEEE Computer Society (2014), https://doi.org/10.1109/ISCA.2014.6853210
37. Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., Yarom, Y.: Spectre attacks: Exploiting speculative execution. In: 2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019. pp. 1–19. IEEE (2019), https://doi.org/10.1109/SP.2019.00002
38. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 104–113. Springer (1996), https://doi.org/10.1007/3-540-68697-5_9
39. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 388–397. Springer (1999), https://doi.org/10.1007/3-540-48405-1_25
40. Krautter, J., Gnad, D.R.E., Schellenberg, F., Moradi, A., Tahoori, M.B.: Active fences against voltage-based side channels in multi-tenant FPGAs. In: Pan, D.Z. (ed.) Proceedings of the International Conference on Computer-Aided Design, ICCAD 2019, Westminster, CO, USA, November 4-7, 2019. pp. 1–8. ACM (2019), https://doi.org/10.1109/ICCAD45719.2019.8942094
41. Krautter, J., Gnad, D.R.E., Tahoori, M.B.: FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018(3), 44–68 (2018), https://doi.org/10.13154/tches.v2018.i3.44-68
42. Kunihiro, N., Honda, J.: RSA meets DPA: Recovering RSA secret keys from noisy analog data. In: Batina, L., Robshaw, M. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8731, pp. 261–278. Springer (2014), https://doi.org/10.1007/978-3-662-44709-3_15
43. Lee, J., Han, D.: Security analysis on dummy based side-channel countermeasures—case study: AES with dummy and shuffling. Appl. Soft Comput. 93, 106352 (2020), https://doi.org/10.1016/j.asoc.2020.106352

44. Levi, I., Fish, A., Keren, O.: CPA secured data-dependent delay-assignment methodology. IEEE Trans. Very Large Scale Integr. Syst. 25(2), 608–620 (2017), https://doi.org/10.1109/TVLSI.2016.2592967
45. Lipp, M., Kogler, A., Oswald, D., Schwarz, M., Easdon, C., Canella, C., Gruss, D.: PLATYPUS: Software-based power side-channel attacks on x86 (2020), https://platypusattack.com
46. Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Fogh, A., Horn, J., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., Hamburg, M.: Meltdown: Reading kernel memory from user space. In: Enck, W., Felt, A.P. (eds.) 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018. pp. 973–990. USENIX Association (2018), https://www.usenix.org/conference/usenixsecurity18/presentation/lipp
47. Lo, O., Buchanan, W.J., Carson, D.: Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). Journal of Cyber Security Technology 1(2), 88–107 (apr 2017), https://doi.org/10.1080/23742917.2016.1231523
48. Mangard, S., Oswald, E., Popp, T.: Power analysis attacks - revealing the secrets of smart cards. Springer (2007), https://doi.org/10.1007/978-0-387-38162-6
49. Mayer-Sommer, R.: Smartly analyzing the simplicity and the power of simple power analysis on smartcards. In: Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1965, pp. 78–92. Springer (2000), https://doi.org/10.1007/3-540-44499-8_6
50. Murdock, K., Oswald, D., Garcia, F.D., Bulck, J.V., Gruss, D., Piessens, F.: Plundervolt: Software-based fault injection attacks against Intel SGX. In: 2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020. pp. 1466–1482. IEEE (2020), https://doi.org/10.1109/SP40000.2020.00057
51. O'Flynn, C., Dewar, A.: On-device power analysis across hardware security domains. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2019(4), 126–153 (2019), https://doi.org/10.13154/tches.v2019.i4.126-153
52. Osvik, D.A., Shamir, A., Tromer, E.: Cache attacks and countermeasures: The case of AES. In: Pointcheval, D. (ed.) Topics in Cryptology - CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2006, Proceedings. Lecture Notes in Computer Science, vol. 3860, pp. 1–20. Springer (2006), https://doi.org/10.1007/11605805_1
53. Percival, C.: Cache missing for fun and profit. In: BSDCan 2005, Ottawa, Canada, May 13-14, 2005, Proceedings (2005), http://www.daemonology.net/papers/cachemissing.pdf
54. Prouff, E.: DPA attacks and S-boxes. In: Gilbert, H., Handschuh, H. (eds.) Fast Software Encryption: 12th International Workshop, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3557, pp. 424–441. Springer (2005), https://doi.org/10.1007/11502760_29
55. Qiu, P., Wang, D., Lyu, Y., Qu, G.: VoltJockey: Breaching TrustZone by software-controlled voltage manipulation over multi-core frequencies. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019. pp. 195–209. ACM (2019), https://doi.org/10.1145/3319535.3354201
56. Quisquater, J., Samyde, D.: Electromagnetic analysis (EMA): measures and counter-measures for smart cards. In: Attali, I., Jensen, T.P. (eds.) Smart

Card Programming and Security, International Conference on Research in Smart Cards, E-smart 2001, Cannes, France, September 19-21, 2001, Proceedings. Lecture Notes in Computer Science, vol. 2140, pp. 200–210. Springer (2001), https://doi.org/10.1007/3-540-45418-7_17
57. Ramesh, C., Patil, S.B., Dhanuskodi, S.N., Provelengios, G., Pillement, S., Holcomb, D.E., Tessier, R.: FPGA side channel attacks without physical access. In: 26th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines, FCCM 2018, Boulder, CO, USA, April 29 - May 1, 2018. pp. 45–52. IEEE Computer Society (2018), https://doi.org/10.1109/FCCM.2018.00016
58. Schellenberg, F., Gnad, D.R.E., Moradi, A., Tahoori, M.B.: An inside job: Remote power analysis attacks on FPGAs. In: Madsen, J., Coskun, A.K. (eds.) 2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018. pp. 1111–1116. IEEE (2018), https://doi.org/10.23919/DATE.2018.8342177
59. Schellenberg, F., Gnad, D.R.E., Moradi, A., Tahoori, M.B.: Remote inter-chip power analysis side-channel attacks at board-level. In: Bahar, I. (ed.) Proceedings of the International Conference on Computer-Aided Design, ICCAD 2018, San Diego, CA, USA, November 05-08, 2018. p. 114. ACM (2018), https://doi.org/10.1145/3240765.3240841
60. Shanmugam, D., Selvam, R., Annadurai, S.: Differential power analysis attack on SIMON and LED block ciphers. In: Chakraborty, R.S., Matyas, V., Schaumont, P. (eds.) Security, Privacy, and Applied Cryptography Engineering - 4th International Conference, SPACE 2014, Pune, India, October 18-22, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8804, pp. 110–125. Springer (2014), https://doi.org/10.1007/978-3-319-12060-7_8
61. Tang, A., Sethumadhavan, S., Stolfo, S.J.: CLKSCREW: Exposing the perils of security-oblivious energy management. In: Kirda, E., Ristenpart, T. (eds.) 26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017. pp. 1057–1074. USENIX Association (2017), https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/tang
62. Tiri, K., Verbauwhede, I.: A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: 2004 Design, Automation and Test in Europe Conference and Exposition (DATE 2004), 16-20 February 2004, Paris, France. pp. 246–251. IEEE Computer Society (2004), https://doi.org/10.1109/DATE.2004.1268856
63. Tiri, K., Verbauwhede, I.: Design method for constant power consumption of differential logic circuits. In: 2005 Design, Automation and Test in Europe Conference and Exposition (DATE 2005), 7-11 March 2005, Munich, Germany. pp. 628–633. IEEE Computer Society (2005), https://doi.org/10.1109/DATE.2005.113
64. Yarom, Y., Falkner, K.: FLUSH+RELOAD: A high resolution, low noise, L3 cache side-channel attack. In: Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014. pp. 719–732. USENIX Association (2014), https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/yarom
65. Zhao, M., Suh, G.E.: FPGA-based remote power side-channel attacks. In: 2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA. pp. 229–244. IEEE Computer Society (2018), https://doi.org/10.1109/SP.2018.00049