IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 3, pp. 294–320. DOI:10.46586/tches.v2023.i3.294-320

JitSCA: Jitter-based Side-Channel Analysis in Picoscale Resolution

Kai Schoos, Sergej Meschkov, Mehdi B. Tahoori and Dennis R. E. Gnad

Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany.
kai.schoos@student.kit.edu,{sergej.meschkov,mehdi.tahoori,dennis.gnad}@kit.edu

Abstract. In safety and security conscious environments, isolated communication channels are often deemed necessary. Galvanically isolated communication channels are typically expected not to allow physical side-channel attacks through that channel. However, in this paper, we show that they can inadvertently leak side channel information in the form of minuscule jitter on the communication signal. We observe worst-case signal jitter within 54 ± 45 ps using an FPGA-based receiver employing a time-to-digital converter (TDC), which is a higher time resolution than a typical oscilloscope can measure, while in many other systems such measurements are also possible. A transmitter device runs a cryptographic accelerator, while we connect an FPGA on the receiver side and measure the signal jitter using a TDC. We can indeed show sufficient side-channel leakage in the jitter of the signal by performing a key recovery of an AES accelerator running on the transmitter. Furthermore, we compare this leakage to a power side channel also measured with a TDC and prove that the timing jitter alone contains sufficient side-channel information. While for an on-chip power analysis attack about 27k traces are needed for key recovery, our cross-device jitter-based attack only needs as few as 47k traces, depending on the setup. Galvanic isolation does not change that significantly. That is an increase by only 1.7×, showing that fine-grained jitter timing information can be a very potent attack vector even under galvanic isolation. In summary, we introduce a new side-channel attack vector that can leak information in many presumably secure systems. Communication channels can inadvertently leak information through tiny timing variations, known as signal jitter. This could affect millions of devices and needs to be considered.

Keywords: side-channel · jitter · power · timing · galvanically isolated

1 Introduction

Timing side channels [Koc96] typically affect software implemented cryptography or timing variations in the microarchitecture and are still a practical threat to various systems today [BT11, MSEH20, KGG+18, YSG+19, GYCH18, Sze19]. Power analysis [KJJ99] and electromagnetic attacks often no longer need dedicated measurement equipment, and have been shown feasible from inside the same chip [SGMT18a, ZS18, GKT19, OD19, WPH+22], the same power domain [SGMT18b], or close proximity [KP13, CPM+18, ZZL+22]. When taking a closer look, these attacks cannot be categorized separately, as already Simple Power Analysis (SPA) is basically a timing attack through power measurements [KJJ99]. Vice versa, a new range of side-channel attacks observe very fine timing differences caused by physical variations as an estimate of power consumption in the victim [SGMT18a, ZS18, SGMT18b, GDTM21, MGKT22].

These *timing-based power analysis attacks* typically re-use existing hardware components and reconfigure or use them in a way to be sensitive to power or voltage variations on the device itself [SGMT18a, GDTM21, WPH+22]. When an attacker has access to these

Licensed under Creative Commons License CC-BY 4.0.
Received: 2023-01-15 Accepted: 2023-03-15 Published: 2023-06-09

components, they can thus perform power analysis attacks on other components in the system with the same power domain. Among these, delay line-based sensors have been the most researched, and have been shown sensitive enough to voltage fluctuations not only from the same System on Chip (SoC) [ZS18, GDTM21], but also from other components connected to the same power supply [SGMT18b, GRS20]. Nevertheless, all of these attacks still work in the same power supply domain, where in general galvanic isolation can improve security somewhat [SPK+10, WXL+21]. Timing differences that can be measured from another system were so far leveraged for attacks in a more classical way, when the respective cryptographic implementation was not constant time [BT11]. What has been shown is that timing jitter in a Controller Area Network (CAN) bus signal can be used to identify a hardware device, due to respective *manufacturing process variations* of the device [ODAF21].

What has not been shown so far, is that minuscule timing differences such as signal jitter can show data-dependent *runtime variations* sufficiently for side-channel attacks. Furthermore, all the mentioned works stay in the same power domain and thus cannot differentiate whether the measured side-channel leakage is from the signal that is measured (i.e. the clock) or the actual sensor being sensitive to the respective physical variations such as voltage [SGMT18a, ZS18, SGMT18b, GDTM21, MDL+22].

In this work, we will address those points. We will show for the first time that signal jitter contains enough side-channel information for a key recovery attack, which is dependent on the physical variations in the transmitter of that signal. We can reject that those variations come from direct electrical coupling, since we also show our attack to work with galvanic isolation. To demonstrate that, our experiments are first performed on a single Field Programmable Gate Array (FPGA) platform for reference, where our design consists of a victim that is transmitting a clock as its signal, and an attacker that receives this signal, which is essentially reproducing on-chip power analysis. We gradually spread out this design to two FPGAs, then a communication through HDMI, and finally a galvanically isolated HDMI signal. In summary, we make the following contributions:

- We show that signal jitter is a new side-channel attack vector, which we can clearly separate from direct power/voltage side channels.
- Escalating timing-based power analysis attacks from being performed within systems supplied by the same power supply to galvanically isolated communication between two systems.
- By carefully designed experiments, we clearly differentiate between the power leakage observed *inside* an FPGA-based delay sensor, and the *outside* timing leakage from the signal jitter that is measured using the sensor of the adversary.
- We show that our experiments are generic enough to be performed with two FPGA systems from different vendors and a galvanically isolated HDMI link between the boards.

The impact of this work is beyond our results and implies that many systems that were assumed to be connected securely are suspect to this new type of side-channel leakage. Many mission critical systems in medical and military applications use galvanically isolated device-to-device communication or enforce unidirectional communication [Fib23, Ind16], but also consumer-oriented or generic networking devices are at risk.

In the remaining paper, we will first summarize background and related work in Section 2 and give generic adversarial models for the experiments performed later. In Section 3 we will explain how the individual components of information leakage are measured. Our experimental setups are described in Section 4. In Section 5 we show our results with an interpretation and discussion in Section 6. Finally, Section 7 concludes the paper.

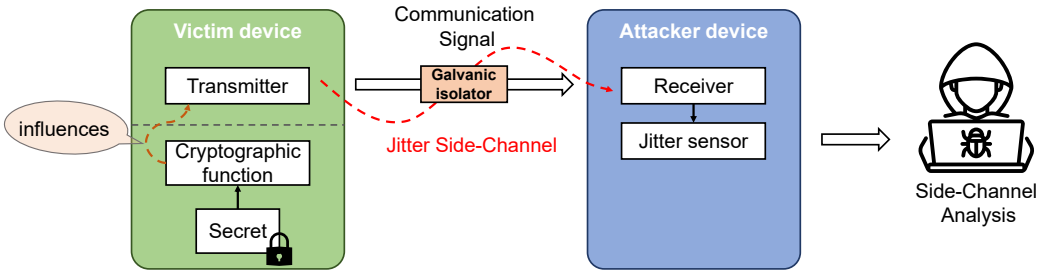


Figure 1: Generic Adversary Model used in this work.

2 Preliminaries

2.1 Adversarial Models

The results from this paper can apply to various scenarios. However, in all cases the attacker has access to a communication signal coming from the victim, while the victim is performing sensitive computations such as cryptography, as shown in Figure 1. These computations affect the transmitted signal and even if it is galvanically isolated, an adversary can measure jitter, tiny timing deviations from true periodicity, of the received signal. This information can then be used for side-channel analysis to recover the secret information.

Additionaly, we assume three scenarios of the generic adversarial models shown in Figure 2, which are later used in our experimental setups. The on-chip case Figure 2a assumes that the attacker and victim are on the same chip, and the attacker has access to a communication signal from the victim. Both, the attacker and the victim share the same power supply, which is also addressed in previous publications [SGMT18a, SGMT18b, GRS20, ZS18, GDTM21]. This scenario can occur inside SoCs or cloud environments, where the tenants have access to different parts of the hardware. This model is mainly used as a baseline comparison in this paper.

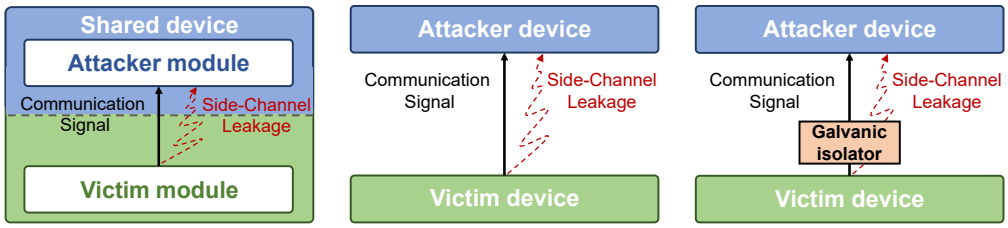
For the inter-device case Figure 2b, attacker and victim are on different devices. They do not share a power supply, but are still connected by the same ground. The attacker receives a communication signal from the victim, which carries the side-channel information. This is the most common case and includes scenarios like an attacker having access to interface of a digital clock or only having access to the communication I/O-signals of an otherwise tamper-resistant device operating on sensitive data.

Finally, for the isolated inter-device case Figure 2c, the two devices are only connected by a galvanically isolated digital communication signal. Both devices do not share any common electrical reference. This setup can be found in devices that transmit data via optical links or galvanically isolated connections, as they are found for example in high-speed networking devices or in various mission-critical appliances with respect to safety and security [Fib23].

2.2 Related Side-channel Attacks

Timing-based side-channel attacks typically use variations in the runtime of a software, which can leak information secrets about the data that is being processed [Koc96]. Even though such attacks have been known for over two decades, timing attacks can still be real-world problems, as recently shown by breaking firmware-based Trusted Platform Modules (TPMs) [MSEH20]. Furthermore, modern CPU architectures use various best-effort optimizations that lead to inconsistent timing that gets increasingly exploited [GYCH18, YSG+19, Sze19, WPH+22], especially since the introduction of Spectre attacks [KGG+18].

Power analysis attacks are different in a way that they are usually harder to prevent from



(a) On-Chip **(b)** Inter-Device **(c)** Isolated Inter-Device

Figure 2: Adversary Models. (a) On-chip attack, where victim and attacker reside on the same chip. (b) Inter-device attack, where victim and attacker share ground and communication signals. (c) Galvanically isolated inter-device attack, where victim and attacker only share a galvanically isolated communication signal.

the software implementation level compared to timing. Thus, cryptographic accelerators often leak information through data-dependent power consumption, if no specific attempts are made to prevent it [MKP12]. Since the introduction of Differential Power Analysis (DPA) by Kocher et al. [KJJ99], more and more attacks have shown the security risk from allowing unprivileged power measurements [LKO+21], for which countermeasures of various types are actively developed [MKP12, WXL+21, KMMS21]. However, as power analysis attacks analyze the power consumption over time, they implicitly and sometimes explicitly consider timing as well [KJJ99].

More recently, various types of very fast timing measurements have also been shown to allow indirect measurements of power, which we would call *timing-based power side channel attacks*. Inside FPGAs, Time-to-Digital Converters (TDCs) have been used to measure transistor delay, which is among other effects affected by voltage of the entire FPGA chip, and can be used for power analysis attacks [SGMT18a, ZS18, MDL+22]. Similarly, generic delay lines in SoCs were also shown to allow side-channel attacks [GDTM21], which extends such attacks to much more potential systems. In [MGKT22] chip testing methods are used to directly see the transient circuit behavior that would later cause differences in power consumption that can be used for attacks. Extending the FPGA-based attacks beyond a single chip, it was shown that even other devices connected to the same power supply could be measured [SGMT18b, GRS20]. Timing measurements inside a processor can also be used as a proxy for the power consumption of the respective processor, when the scheduling of Dynamic Voltage and Frequency Scaling (DVFS) is being monitored, leading to a successful key recovery attack [WPH+22]. These works looked into self-measuring the power consumption of the respective chip, while in this paper, we will look into timing of the signals received from *another* device, where the jitter in the signal will be a proxy for the power consumption of the transmitting device.

Regarding galvanic isolation, it was shown that output port pins of a victim chip with a constant value can be connected to an optocoupler, and when the receiving side is measured, the analog variations from the victim are still observable in the coupled side [SPK+10], but increasing the required traces for a successful power analysis attack by about 4500×. In [WXL+21] an ASIC was manufactured with a galvanically isolated AES module integrated, they stopped measuring after about 600× as many traces than for their non-isolated comparison and could not launch a successful attack with that.

Considering on-chip power side-channels in related work, they were not just exploited through delay-based sensors in FPGAs [SGMT18a, ZS18]. It was also shown that the influence from digital logic can affect analog components in the same SoC, which can influence a Digital-to-Analog Converter (DAC) and similarly it can even end up in a wireless signal [CPM+18]. Other works have shown that on the system itself, the noise of

an Analog-to-Digital Converter (ADC) can be used for power analysis attacks, sufficient to break through privilege levels [GKT19, OD19, GNST22]. These works acquire side-channel information in the same power domain, by measuring analog properties, such as time and magnitude of a signal. The typical time resolution is in the range of clock cycles of the victim system, i.e. $10 - 100$ ns, with a typical resolution of $6 - 12$ bit. In comparison, we only measure a discrete digital 1-bit signal from which also a disrete (clocked) time behavior is expected. However, because we measure at a much higher time resolution than previous work (12 ps), analog timing properties in the form of jitter become visible, which we exploit as a side channel.

2.3 Power model and Leakage Assessment

In this work, we execute a Correlation Power Analysis (CPA) attack [BCO04], and correlate the actual jitter measurements as a power estimate with a model. The used power model is slightly different from classical CPA attack and targets a single bit instead of a whole byte, as previous works in FPGAs have done successfully [SGMT18a]. For the attack on the last encryption round of AES, we model power with $p_m = sbox^{-1}(k_{guess} \oplus c_i) \wedge 2^b$, where k_{guess} is the key byte guess, c_i is the corresponding ciphertext byte and b is the bit selected for the attack. The attack to recover one key byte is successful if at least one of its bits has a higher absolute correlation value than the other key guesses. In this work, we will report the required amount of traces for the correct key guess to be the point after which no other key will correlate more than the correct one. For our experiments, we use the FIPS AES test cipher key *2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c* [DBN+01].

To generally show the presence of first order statistical leakage in our experiments, we analyze two sets of traces as introduced in Test Vector Leakage Assessment (TVLA) [GGJR+11]. The first set is measured when computing on a fixed input value, and the second set when computing on various random input values. Then, a Welch's t-test is performed to statistically compare the two sets. When the two sets are indistinguishable, no leakage is assumed.

Due to the high number of samples K per trace, the t-test can not be evaluated against a single threshold value TH_t. As the number of samples per trace grows, the chance of having certain samples randomly surpassing a fixed threshold grows with it. In fact, for traces with 1 million sample points, 99.9% of leakage free devices are classified as leaky under a threshold of $TH_t = 4.5$, which is classically used [DZD+18]. The threshold for a given significance level can be found by interpreting all of the t-tests as a mini-p procedure. The threshold is then computed by:

$$\alpha_{TH} = 1 - (1 - \alpha)^{\frac{1}{K}} \tag{1}$$

$$TH_t = CDF^{-1}_{\mathcal{N}(0,1)}(1 - \alpha_{TH}/2) \tag{2}$$

With α being the desired significance level and K the number of samples per trace. Theoretically, we would need to use the Cumulative Distribution Function (CDF) for the t-distribution here. However, as the CDF for the t-distribution converges toward the CDF for the standard normal distribution $\mathcal{N}(0, 1)$ for high degrees of freedom, the latter can be used, as this boundary condition holds in our case. This kind of evaluation is used for what is called the *TVLA total-plots* in this work.

A different evaluation method based on Higher Criticism (HC) is used for what is called the *TVLA progress-plots* in this work [DZD+18]. This method takes into account the distribution of the t-values and thus gains more detection power for devices with some countermeasures in place [DZD+18]. To carry out this procedure, a HC statistic is computed for each of the K t-values where K corresponds to the number of time steps per trace. First, the t-values are transformed into their respective p-values by computing the survival function $1 - CDF_{\mathcal{N}(0,1)}(t)$. Again, the CDF for the standard normal distribution

can be used here because of how the t-distribution converges towards $\mathcal{N}(0,1)$ for high degrees of freedom. The p-values are then sorted in ascending order, before the HC estimator is computed as follows:

$$\widehat{HC}_{K,i} = \frac{\sqrt{K}(\frac{i}{K} - p_i)}{\sqrt{p_i(1-p_i)}} \qquad \text{for i=1,...,K} \tag{3}$$

$$\widehat{HC}_{K,max} = \max_{1 \le i \le \frac{K}{2}} \widehat{HC}_{K,i} \tag{4}$$

Here, K HC estimator values $\widehat{HC}_{K,i}$ are computed. These are the individual HC-values for a given trace of length K, and p are the p-values received from the t-values, sorted in ascending order. $\widehat{HC}_{K,max}$ is the final statistic that will be thresholded to find out, whether or not the signal is leaking. The number of traces N does not have any effect on the computation. This is due to the fact that for large number of degrees of freedom (df), the t-distribution simply becomes a standard normal distribution. Even for the smallest amount of traces analyzed (1000), df is large enough for this boundary condition to bet met.

In order to find the threshold $b^{HC}_{K,\alpha}$, a monte-carlo simulation is run. For the null-hypothesis, the p-values follow a uniform distribution $p \sim U(0,1)$. Thus, for the simulation, $\widehat{HC}_{K,max,null}$ is computed 1,000,000 times for K p-values sampled from $U(0,1)$ in order to estimate the distribution for $\widehat{HC}_{K,max,null}$. The threshold $b^{HC}_{K,\alpha}$ is then defined as the $(1-\alpha)$ quantile of the $\widehat{HC}_{K,max}$ distribution, with α being the significance level. The signal is said to be leaking if $\widehat{HC}_{K,i} \geq b^{HC}_{K,\alpha}$.

2.4 Signal to Noise Ratio (SNR)

The Signal to Noise Ratio (SNR) is used in order to compare the amount of leakage between different conditions. First, the power model is computed for one correct key byte for all the traces. After that, the traces are grouped according to the value of the power model. In our case of the bitwise power model, the traces fall into one of two groups, where the power model either evaluates to a '0' or a '1'. For each group $g \in [0,1]$ and each time step $k \in [1,...,K]$, the signal $Q_{g,k}$ is computed as the arithmetic mean over the traces that are part of that group $\mathscr{T}^g$.

$$Q_{g,k} = mean(\mathscr{T}^g_k) \tag{5}$$

The noise $\nu_{n,k}$ of each of the NxK measurements is determined by subtracting the signal of the group of the n-th trace $Q_{g,k}$ from the original measurement $\mathscr{T}_{n,k}$:

$$\nu_{n,k} = \mathscr{T}_{n,k} - Q_{g,k} \tag{6}$$

Finally, the Signal to Noise Ratio (SNR) for each timestep SNR_k is determined by computing the variance across all the groups for the signal and the noise and taking their quotient:

$$SNR_k = \frac{Var(Q_k)}{Var(\nu_k)} \tag{7}$$

Where Q_k are the signal values for all the groups for time step k and ν_k are all of the noise values for time step k.

2.5 High-Speed Timing Measurements with Delay Lines

One possible way to implement a TDC is a delay line. TDCs were originally used in single timing measurements for physical experiments that required a high resolution [Kal03], and

are also used in chip testing [LAP+11, MGKT22]. In an abstract way, a *Start* and a *Stop* signal control the TDC, which measures the time between the signals, after which its data gets collected to measure that timing difference [Kal03]. When tapped delay lines are used, memory elements are added between a long line of buffers or inverters. The start signal is fed into the delay line, while the stop signal controls the memory elements (registers or latches). Like that, the progression of the start signal is an indication of how much time has passed between the signals.

By using the same signal for both start and stop, not the delay of the signal is measured, but instead it is an indicator of the power consumption of the device itself, which can be used for power analysis attacks [SGMT18a, SGMT18b, GDTM21, MDL+22]. Our variation of that sensor is shown in Figure 3. Delay lines or TDCs are available in many devices already [LAP+11, GDTM21] such that no external measurement device is needed, and for flexibility and experimental purposes can also be synthesized into FPGA fabric [SGMT18a, SGMT18b].

In detail, these TDCs are made up of two general parts: A delay line and the storage registers. When measuring an estimate for power, the clock is fed into the delay line and into the clock connections for the storage registers. For ideal components, the input signal (i.e. the clock signal in this case) would instantly propagate to the end of the delay line. However, in the real world, the time that the clock edge takes to travel through the delay line is non-zero. The storage registers all record the value of their corresponding latch at the moment the clock edge reaches their clock input, freezing the state of the delay line at that moment. Typically, the delay line is fine-tuned manually by changing its length, such that the previous clock-edge is being propagated through the delay line when the storage registers are clocked, thereby sampling the delay between the previous (negative) clock-edge and the current (positive) clock-edge. The result is a string of '1's followed by a string of '0's, marking the clock edge. The point in the string where the transition from 1's to 0's happen thus marks the length of the last negative pulse. The variation of this value was shown to be an estimator for the variation of the voltage level of the system, which can be used in power analysis attacks [SGMT18a, GDTM21, MDL+22].

2.6 Jitter and its Measurement

Typically, signal jitter is categorized into bounded and unbounded jitter, where bounded jitter is further broken down into correlated and uncorrelated jitter [Tel14]. Data Dependent jitter (DDj) is considered to be correlated to the transmitted data, but voltage fluctuations of the transmitter are typically addressed as a form of random or uncorrelated jitter.

Others have already explored to analyze timing variations of communication channels in the form of jitter, but not to extract secret data. More specifically, in [ODAF21] an identification approach is shown that measures jitter on an automotive CAN bus by using an FPGA-based TDC sensor. This jitter is used to uniquely identify the respective transmitter of the message by detecting its typical jitter characteristics, influenced by (systematic) manufacturing process variations. The respective input to their measurements come from a CAN transceiver module and from there are forwarded to a coarse time-sampling circuit. The coarse time sampler measures multiple CAN messages, which are then forwarded to a TDC circuit that measures the period between messages, i.e. the rise and fall times of the CAN signal. In the end, they report a minimum delay resolution of $219ps$, which is sufficient to measure the slow 500 kbps CAN bus signal with comparatively high jitter in [ODAF21]. However, that resolution would not be sufficient for this work, where jitter was observed to be usually less than $100ps$.

Alternatives to this measurement method would be to use an external spectrum analyzer and continuously measure the phase noise of the system, where care should be taken that the frequency of the local oscillator of the device is close to the frequency of the signal under test. Furthermore, repetitive measurements would be required. Digital oscilloscopes

on the other hand may have problems measuring such high resolution phase variation out of the box, since 11.5*ps* resolution relates to about 87 GHz.

In this work, we will use a delay line TDC FPGA design explained in Section 2.5, that has been reported with a timing resolution of about 11.5*ps* in one of the same FPGA platforms that we use [MGKT22]. It uses the delay from the input to the sampling registers of the delay line to catch a single period of the signal. The aim is also different to [ODAF21] in that we do not want to measure jitter that can be reproduced per transmitter, but jitter that changes its characteristic depending on the data being processed inside the transmitter. To measure jitter for side-channel analysis, we implement a configurable delay line based TDC following the design from [MGKT22], which adds flexibility regarding the time range that we want to measure. Further changes are described in Section 3.1.

3 Methodology

In this work, we measure the timing variations of a (clock) signal transmitted from a victim device to the attacker device, also known as jitter. An attacker receives this signal, and measures this signal's jitter. While the transmission happens, the victim is also running a cryptographic accelerator, which we assume causes minor voltage fluctuations in the entire power domain of the victim circuit. These voltage fluctuations also have an impact on the cycle-by-cycle jitter of the transmitted signal, sufficient for side channel-based key recovery attacks.

The attacker module aims to measure this signal jitter and derive the victim's power consumption and by that secret keys from the cryptographic operations. With an FPGA-based TDC, the attacker will measure the delay between two consecutive edges of the received signal, which can also be run continuously to get a reading on every change of the data signal. The variation of this delay is then a unit-less value of the jitter that can be used in side-channel analysis, where in this paper CPA and TVLA is used.

TDC sensors are also directly sensitive to voltage and temperature variations. This relation makes it possible, to use timing variations of a measured (clock) signal with a TDC sensor as power estimate for side-channel analysis. In related work, the TDC sensor is usually in the same power domain as the victim and the measured (clock) signal itself as well as the delay line components are affected by the voltage fluctuations in this power domain. Therefore, an adversary can measure voltage fluctuations caused by a running victim cryptographic circuit.

This section explains how we measure signal jitter with delay line-based TDCs, as well as how we designed our experiments to allow distinction between power in the Power Distribution Network (PDN) supplying the delay line versus the estimate we get through jitter from the victim, potentially even when the systems are galvanically isolated and no direct leakage through the PDN is possible.

3.1 Measuring Jitter for Side-Channel Analysis

As introduced earlier, we use delay line-based TDCs to measure signal jitter. Since TDC sensors are also sensitive to voltage and temperature variations, they are usable in power-based side-channel attacks, which we will separate based on the experimental setup.

The basic structure of the used TDC sensors is illustrated in Figure 3. The same (clock) signal is propagated through the delay line and is used to trigger the storage registers. With a suitable variation of the delay line length, which can be configured manually for every experiment, relative time differences from the actual rising edge to the falling edge of the previous clock cycle can be measured. The measured time window that a TDC is measuring is generally fixed. For clock signals, every rising edge is preceded by a falling edge in a relatively short time window for which the TDC is configured. This is not true

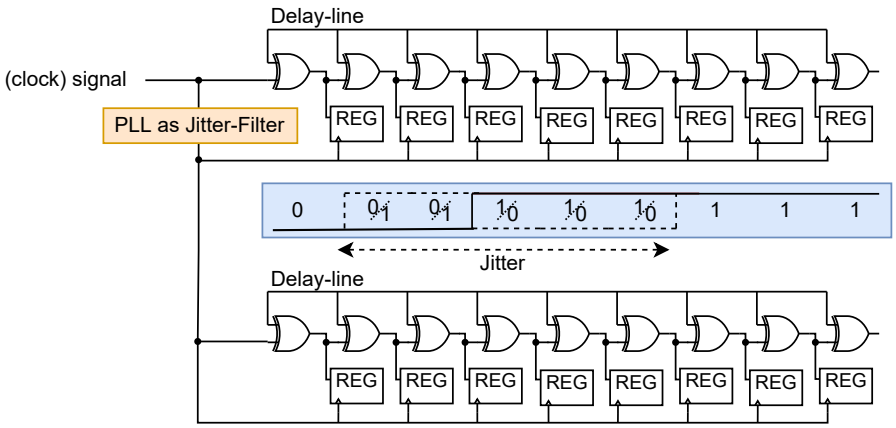


Figure 3: Delay line-based TDC sensor to measure voltage from variations in the speed of the XOR cells as well as from the clock signal. A clock enters the delay line while they are sampled by a previous edge from the same clock signal, which will include clock jitter. This is our interpretation of previously published FPGA-based voltage sensors [SGMT18a].

for general data signals. Therefore, when measuring jitter on a data signal, some traces may become useless and need to be sorted out while preprocessing the data.

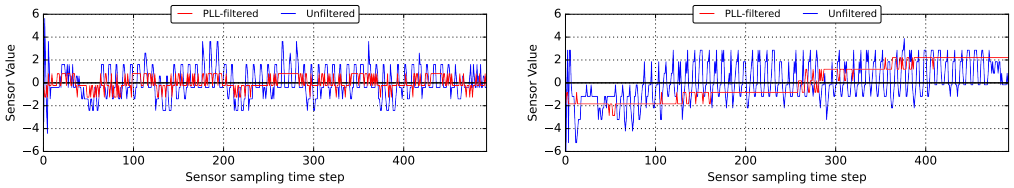
The (clock) signal transition itself is stored in the storage registers as a sequence of 0's followed by a sequence of 1's. This represents a relative time point of the actual (clock) signal transition in the previous clock cycle, with a discrete resolution of about 11.5 picoseconds for one of the used boards platforms (PYNQ-Z1) [MGKT22], but unknown for our other ULX3S platform. To receive the final sensor value, the values of the storage registers are summed up. This quantity reflects the duration of the latest negative clock pulse, which is the time of the nominal clock period affected by jitter. For that, we show exemplary measurement traces in Figure 4 for both of the systems used in our experiments.

For our experiments, we use two delay line based TDC sensors in parallel where the second sensor acts as reference measurement, as shown in Figure 3. While the first sensor measures the jitter in the signal as it is, for the second one we try to filter jitter first. For this, the signal is passed through a Phase Locked Loop (PLL) with an output clock of the same frequency, since PLLs have the property of filtering out jitter from the source clock, while maintaining a constant phase relationship to it. Thus, if we use the output of the PLL as an input signal to the TDC, this measurement should contain almost no jitter (and no side-channel leakage) from the original clock.

The PLL-filtered and unfiltered measurements are both included in Figure 4 as red and blue plot respectively, showing that the PLL in fact reduces jitter. To what extent that translates to side-channel attack success will be explored later. Filtering the jitter with a PLL can still be insufficient and include side-channel information, especially if the PLL itself is in the same PDN as the victim. Please note, in all cases, using the PLL as a filter only acts as an experimental vehicle applied by the attacker. It cannot work as a countermeasure, which is why we also need to use galvanic isolation in our experiments, which we discuss later.

3.2 Separation of Leakage from Voltage Influence on the TDC

We hypothesize in this paper that the signal-jitter carries a lot of the side-channel leakage, and thus we need to show that the leakage is only in part due to the electrical connection between the victim and attacker. In literature, the voltage fluctuations in the PDN that supplies the TDC are described as the main influence on the observed variations at its



(a) Single trace captured using PYNQ-Z1

(b) Single trace captured using ULX3S

Figure 4: Exemplary Traces of TDC measurements used in the experiments. The blue lines show traces captured while measuring the unfiltered signal. The red lines show traces captured while measuring a signal filtered using a PLL. The red and blue traces shown for each board are captured in parallel.

output [SGMT18a, GDTM21]. However, to our knowledge, these voltage variations have never been experimentally separated from clock jitter thus far.

We consider that an electrical signal and required ground-connection will still have a minor influence on the attacker's PDN that supplies the TDC, and thus we need to be able to reject this hypothesis of a still-existing power coupling which can be reached with galvanic isolation. Furthermore, we try to filter jitter out of the signal and through that evaluate if some leakage can still be measured.

In order to prove our assumptions, multiple experiments can be used to confirm each other. Here we explain the ideas behind the experiments, which are later detailed in the experimental setup:

- Single board in which victim and attack circuit share the same power supply and clock. The attacker measures the clock with a TDC sensor, reproducing related work such as [SGMT18a] and acting as a comparison baseline. This will be the **On-Chip** experiments (Section 4.1).
- Two boards that have a communication link. The victim operates AES while he is sending an independent message to the attacker through the communication link. A successful attack that uses the signal jitter of the communication link as a side channel will be a necessary precondition to confirm that signal jitter alone is a sufficient side channel. This will be done in the **Wire and HDMI** experiments (Section 4.2 and Section 4.3).
- When performing the previous experiment through a galvanically isolated communication channel, it will finally confirm our main hypothesis, this is done in the **Isolated HDMI** experiment (Section 4.3).
- To cross-check that our galvanic isolation does not allow for power side-channel attacks, we filter the signal jitter with a PLL on the attacker side and try to use this for attacks. While it is most certainly not a perfect filter, an unsuccessful attack can show that the most significant leakage was acquired through the signal jitter and not any other part of the TDC.

As a check, we will additionally perform each attack with PLL-filtering in all of the experiments, using the described two delay lines in Figure 3, to get a better picture of the filtering capabilities. However, the filtering with a PLL only acts as an experimental vehicle and cannot be a countermeasure. It will always be done on the side of the attacker. If it is applied in the same power domain as the victim, it would again be influenced from power supply leakage. Furthermore, it can only filter clock signals and no irregular communication signals.

3.3 Attack Flow

For the attack flow to be possible, the malicious actor must be able to prompt the victim device to perform a sensitive operation or the victim device itself must repeatedly perform some kind of sensitive operation. During this operation, the attacker FPGA must have access to a communication signal, such as a clock signal or transmitted HDMI signal of the victim. The attacker then repeatedly collects measurement traces. These traces are repeated measurements of the communication signal and are stored together with the artifacts generated by the sensitive operation on a workstation PC connected to the attacker FPGA. In our case, these artifacts are the ciphertexts computed by the AES module. After collecting enough traces, CPA and TVLA are performed off-line on the workstation PC.

4 Experimental Setup

The experiments in this paper are conducted with two different kinds of FPGA boards, the Xilinx PYNQ-Z1 that includes a Zynq SoC with ARM processor and FPGA logic with 53k Look-Up Tables (LUTs), and the Radiona ULX3S, which includes an Espressif ESP32 microcontroller and a Lattice ECP5 85F FPGA with 85k LUTs. However, we do not use the processors but only the FPGAs, which are both directly connected to the respective workstation PC through USB for programming, serial communication, and power. We use two ULX3S boards that we enumerate with #1 and #2 throughout the paper. For most of the experiments, the boards are powered from independent power sources, and in some of them we use an Analog Devices EVAL-CN0422 evaluation board for galvanic isolation of HDMI signals [Ana13].

The TDCs on the different attacker devices use fast carry-chain elements, which are either Carry4 elements on the PYNQ-Z1 or CCU2C elements on the ULX3S. The version of AES-128 that is used for the experiments is a simple implementation using 4 parallel S-Boxes, where each round is performed in 5 clock cycles. On both platforms, the TDC is operating at 100 MHz, while AES is always clocked at 12.5 MHz. Details on the individual setups are listed in the following subsections. Please note that 12.5 MHz might be considered low, but this does not mean that higher speeds do not leak, we just did not perform the respective experiments and wanted to show the general possibility of leakage.

4.1 On-Chip: Attacker and victim on a Single Device

This first experiment is performed on a ULX3S board and replicates and extends the concept of internal power analysis attacks from [SGMT18a], which we assume includes the effect of voltage fluctuations both on the chip-internal clock generator resulting in clock jitter, as well as the impact on transistor delay inside the delay line used for the measurements. With the results from this experiment we will have a baseline for the remaining results.

Figure 5 shows a block diagram, giving an overview of this setup. Two clocks drive the design. A fast clock of 100 Mhz is used for most of the design as well as the delay line sensors, and a slower clock with 12.5 Mhz drives the cryptographic module. Both clocks are derived from a 25 Mhz crystal oscillator that is mounted on the development board. The main components of the FPGA on the left hand side are the cryptographic module for computing Advanced Encryption Standard (AES) and the voltage sensor. These two modules send their data to the workstation PC on the right hand side, providing the attacker with the data they need for ciphertext-based CPA.

The plain text that acts as an input for the AES module is generated using a Linear Feedback Shift Register (LFSR) in order to keep the communication overhead small and enable fast trace collection during the experiments. For that same reason, the secret

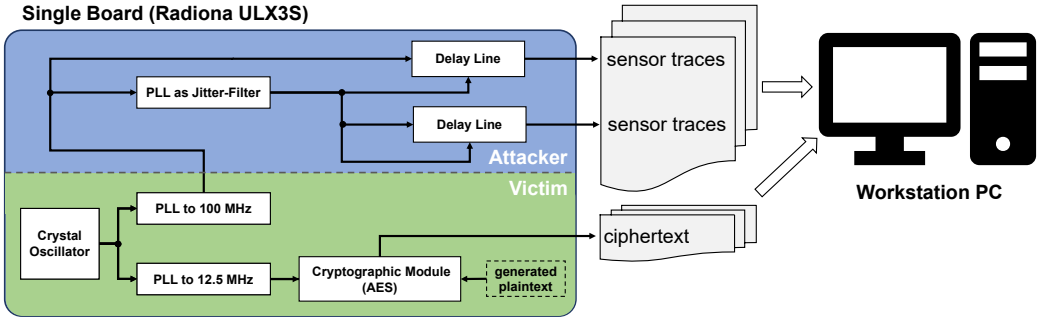


Figure 5: Block diagram for the experimental setup of the On-Chip case.

AES key is hard-coded into the device. Care has been taken to not have any overlap between communicating the computed ciphertext back to the workstation PC and voltage measurements, such that the traces can only depend on the actual encryption process, which is needed for fixed/random-testing of TVLA.

As mentioned before, we need to make sure that we are able to differentiate between jitter and power side-channel effects. For this, in parallel to the replication of internal power analysis attacks [SGMT18a], a second on-chip experiment is conducted that will serve as a control experiment for the experiments that follow. In this experiment, the input signal to the TDC is first filtered using a PLL. This filtering reduces the amount of jitter in the measured signal. Because the PLL is driven by the same PDN as the rest of the victim, this measurement should still show some amount of side-channel leakage. This is in contrast with the following experiments on multiple devices, where we expect that the measurements of the unfiltered signal does show side-channel leakage while the measurements of the filtered signal does not.

4.2 Wire: Attacker and victim connected via Twisted Pair Cable

The setup from the on-chip experiments was expanded to the cross-device condition, essentially by distributing the same design across two FPGAs. This setup has three different variations using different boards, which can be seen in Table 1. This time there is a clear distinction between the attacker and the victim, in contrast to the on-chip condition. One board (the victim) executes a design that simply computes a ciphertext from a random plaintext and hard-coded key using AES. The plaintext is generated by the LFSR module, just as for the on-chip condition, and also the key is identical to the on-chip experiments. Several control signals are shared between the attacker and victim boards in order to streamline and synchronize the measurement process.

Figure 6 displays the distinction between attacker and victim for the cross-device case and shows how different clocks are distributed across the devices. For these experiments, the devices do not share a power connection, but a ground connection. To keep the signal integrity high, we connect ground and the signal wire as a twisted pair from victim to attacker. Both devices are connected to the workstation PC in order to transfer traces (attacker) as well as the ciphertext (victim) to enable the ciphertext-based CPA. The victim device is powered using an external power supply, while the attacker is powered directly via the laptop's USB port.

4.3 (Isolated) HDMI: Attacker and victim connected via HDMI cable

For the second cross-device experiment, instead of using a single clock line, we send an Non-return-to-zero space (NRZS) encoded signal through HDMI, which is the only communication link between attacker and victim. NRZS is a form of Non-return-to-zero

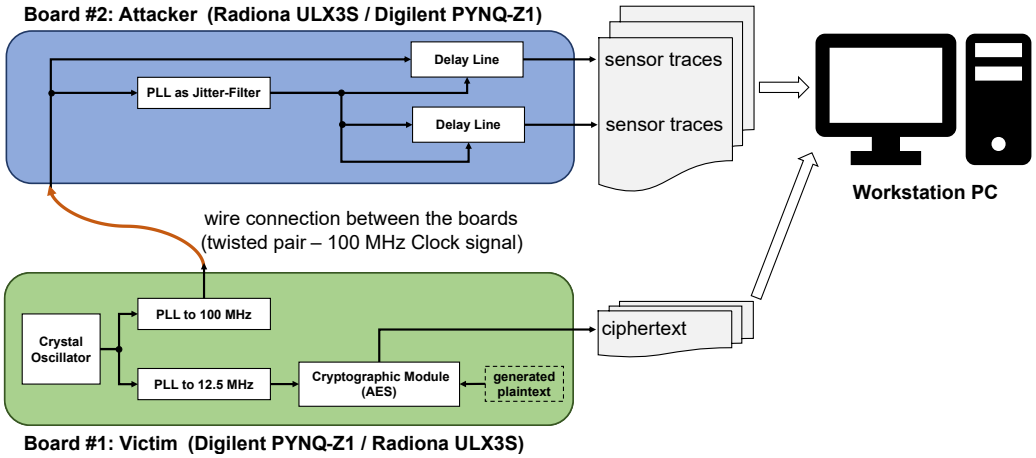


Figure 6: Block diagram for the experimental setup of the cross-device Wire case.

inverted (NRZI) encoding. In NRZI, one of the binary values ('0' or '1') is encoded by inverting the data signal, while the other value is encoded by keeping it constant. The *S* in NRZS means, that the inversion happens on a logic '0' while the signal stays constant for a logic '1'. NRZI is used in various technologies today, like USB and fiber optic communication [SAZT18]. The HDMI link is connected to a pair of Low-Voltage Differential Signaling (LVDS)-pins, which is terminated on the attacker side by differential input buffers. The ULX3S is equipped with an General Purpose Differential Interface (GPDI)-interface to support HDMI, which can only be used as an output. Because of that, we only use the PYNQ-Z1 Board as an attacker with HDMI.

The rest of the setup is similar to that of the twisted pair connection. However, the attack flow is changed to maintain complete galvanic isolation and address the limitations of unidirectional communication. The victim repeatedly computes AES on an internally generated random plaintext message. Before starting the encryption process, it sends an NRZS-encoded magic word through the HDMI-link, which triggers the attacker to start the measurement process. During the time AES is computing on the victim, it only sends NRZS-encoded 0's through the HDMI link, which is essentially a clock at 50 MHz. After the encryption process, the victim sends another magic word followed by the ciphertext to the attacker device, which then sends the measured jitter data (trace) and the associated ciphertext to the workstation PC for further side-channel analysis.

In detail, the victim first sends the magic word `0xBEEF`, then sends a series of 0's until AES finishes (always exactly the same constant time), and then `0xFACE` followed by the ciphertext. After a fixed amount of idle waiting during which more 0's are sent, this is repeated with `0xCAFE`, 0's, `0xFACE`, ciphertext. This means, it alternates between `0xBEEF` and `0xCAFE` as the starting magic word. The attacker has no handshaking possibility and needs to properly listen to these messages of the victim to record the needed data for performing CPA or TVLA.

To show our initial claim to be true, we additionally use an HDMI galvanic isolation module EVAL-CN0422 from Analog Devices [Ana13], which uses multiple isolator ICs. For us, the ADN4654 IC is relevant, that is used for the main HDMI TDMS signals. It is rated as an isolator up to 3.75 kV RMS, compliant with TIA/EIA-644-A and minimum 50-year continuous AC and DC working voltages of 424 V and 536 V respectively. It also has a jitter rating that is described to not include stimulus jitter, i.e. it is added on top of existing signal jitter, with unbounded jitter specified with 2.6-4.8 ps RMS and bounded jitter of 50-116 ps. The ADN4654 isolation mechanism is based on integrated transformer

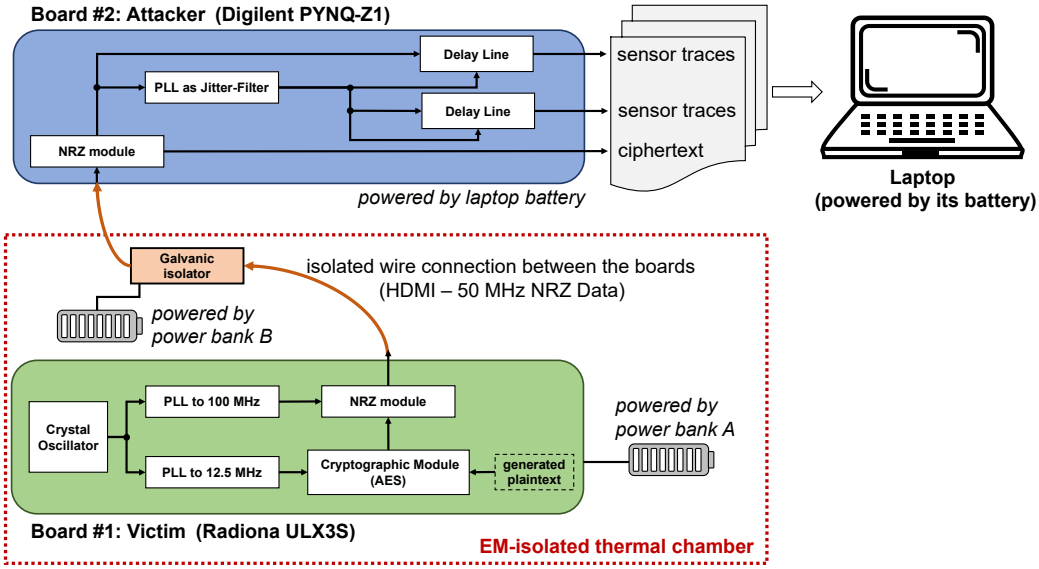


Figure 7: Block diagram for the experimental setup of the cross-device HDMI case, galvanically isolated with the EVAL-CN0422 [Ana13]. All devices are battery powered, so the attacker and victim do not share a common power domain. In addition, the victim device and galvanic isolator are housed in an EM-isolated thermal chamber.

coils. The signal is encoded into small ≈ 1 ns pulses, which on the other side sets or resets a bistable decoder bit, to indicate input transitions.

When connecting one HDMI-cable for the sender and one for the receiver on the EVAL-CN0422 board, this setup can be used just like a standard HDMI-cable. It guarantees galvanic isolation for both, the differential and the single-ended HDMI-signals. So instead of using a single HDMI-cable between victim and attacker, we now use two HDMI cables, one from ULX3S to the EVAL-CN0422, and another one from there to the PYNQ-Z1 board.

To eliminate even more potential information side-channels, each of attacker system, isolation module and victim device run from their own respective battery, making it impossible to have leakage through mains power [GZBE19]. In addition, the victim and the isolation module are placed inside an EM-isolated thermal chamber (Weisstechnik LabEvent T/210/40/EMC [Wei21]). Since it isolates very well, we can not keep it off, since the temperature would increase over the time of collecting traces (about 4 hours). Thus, we run it at 22°C, which was also close to the room temperature. Otherwise, the experimental setup stays identical. This final setup is summarized in Figure 7.

4.4 Summary of Setups

We summarize our experimental setups in Table 1, listing the respective platforms and connection mechanisms. We differentiate between **1.** on-chip attacks, **2.** attacks through a normal wire, i.e. a single-ended twisted pair cable with a signal/ground pair on multiple platforms, and **3.** attacks based on a differential NRZS-encoded signal through an HDMI cable, also with additional isolation through an isolator module. For all of the setups, we will measure with the two sensor variants described in Figure 3.

Table 1: Overview of the attacker and victim platforms used for the various experiments.

Basic Setup	Victim / Attacker Platform	Connection	Signal
On-Chip	ULX3S #1	*on-chip*	Clock
Wire	ULX3S #1 / ULX3S #2	Twisted Pair	Clock
Wire	ULX3S #1 / PYNQ-Z1	Twisted Pair	Clock
Wire	PYNQ-Z1 / ULX3S #1	Twisted Pair	Clock
HDMI	ULX3S #1 / PYNQ-Z1	HDMI cable	NRZ all-0's
HDMI	ULX3S #1 / PYNQ-Z1	Isolated HDMI cables	NRZ all-0's

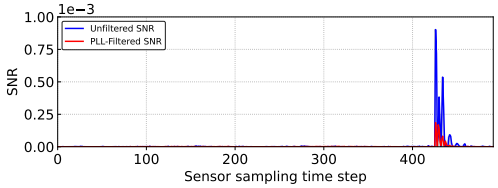


Figure 8: Plot showing the unfiltered and PLL-filtered SNR of the On-Chip experiment

5 Results

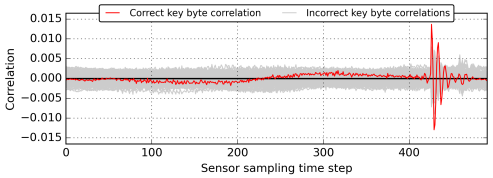
In the same order of presenting our experimental setup, we will go through the respective results. We first show the on-chip results for the reproduction of previous side-channel attacks using TDCs as well as the control condition with a PLL-filtered input signal, which acts as a comparison for the later results, followed by an extension to multiple boards and galvanically isolated HDMI communication. For SNR results and CPA-based attacks, we will always show the correlation result for byte 0 for comparability.

5.1 On-Chip Attacks

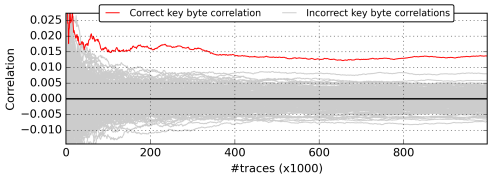
Figure 8 shows the SNR for the unfiltered condition and the condition using the PLL-filter for the on-chip condition. When comparing the SNR values, the PLL-filtered condition shows lower SNR. Thus, compared to the original clock, the PLL-filter seems to be less affected by voltage variations that contain side-channel information. However, the SNR is still considerably higher during the last round of AES as compared to the rest of the time. This means that both signals definitely carry leakage, considering that we are performing a known-ciphertext attack.

Figures 9 and 10 show the results for the CPA attack for the on-chip condition. Figures 9a and 10a show the correlation between the byte hypothesis and all 1 million collected traces for every timestep. Wrong byte hypotheses are displayed in gray, whereas the correct byte hypothesis is represented in red. This kind of plots will be called *CPA total-plots* from here, as they show the correlation for every timestep over the total of traces.

Figures 9b and 10b relate the correlation of a certain byte hypothesis with the number of traces employed in the analysis. These progress plots always refer to a single time step, so they can be thought of as a depth-slice of the corresponding total-plot at that time step. The time step was chosen to be the one for which the correlation of the correct byte was at a maximum and located at the end of the trace, such that it must be during the last round of AES. If the total-plot showed no points of interest for one of the delay lines, the same time step was chosen for the progress-plot of both delay lines. Again, the wrong hypotheses are shown in gray and the correct hypothesis is shown in red. We will be calling this kind of plot *CPA progress-plot*, as they show the progress of the correlation

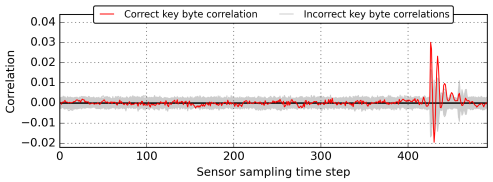


(a) CPA total-plot for the unfiltered delay line

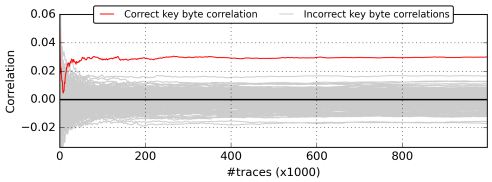


(b) CPA progress-plot for the unfiltered delay line; the correct key byte leaks after 27k traces

Figure 9: CPA plots for over 1 million traces for the delay lines measuring the unfiltered signal for the on-chip experiment on ULX3S. The red lines represent the correct key guesses, while the gray lines represent the incorrect key guesses.



(a) CPA total-plot for the PLL-filtered delay line



(b) CPA progress-plot for the PLL-filtered delay line; the correct key byte leaks after 40k traces

Figure 10: CPA plots over 1 million traces for the delay lines measuring the PLL-filtered signal for the on-chip experiment on ULX3S. The red lines represent the correct key guesses, while the gray lines represent the incorrect key guesses.

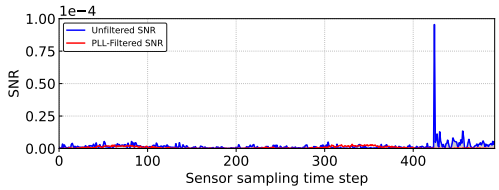
coefficient over all the collected traces.

Figure 9 shows the CPA results for the delay line measuring the unfiltered clock signal. This is essentially the reproduction of previous power side-channel attacks and serves as a control for the experiments that follow. The figure clearly shows that the correct key byte can be recovered easily, as the red line stands out strongly in both plots. Not all bytes could be recovered during our experiments, however if any byte could be recovered, byte 0 usually showed the strongest leakage, which must be due to the specific AES implementation. High absolute values of the red line around timestep 420 on the total-plot indicate the computation of the S-Box during the last round of AES. As the red line separates from the rest of the lines after about 40k traces in the progress plot, we can say that the attack is successful after 27k traces. This quantity is also reflected in Table 2.

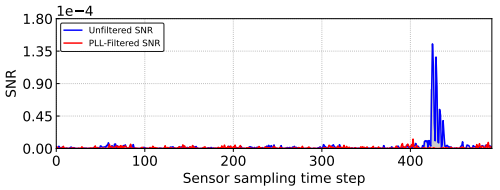
The results in Figure 10 from the on-chip separation condition are very similar to those from the on-chip baseline condition. In our experiment, the correct key byte for byte 0 could only be recovered after 40k traces. This difference to the previous experiment is not significant enough for a conclusion and is probably due to random effects. This result is expected, since the PLL we use for filtering resides in the same power domain and is thus again affected by the running AES module inside the chip, adding no benefit in terms of side-channel security. However, in the following results, this PLL-filtering serves as a valuable control experiment.

5.2 Wire-connected Attacks through a Twisted Pair Cable

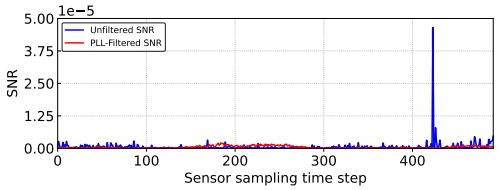
In order to be more general, the results for the wire-based experiments have been collected on three different setups. In the first setup, victim and attacker are both ULX3S boards. In the second setup, the attacker is an ULX3S while the victim is a PYNQ-Z1 and for the



(a) SNR plot for attacker ULX3S #1 and victim ULX3S #2



(b) SNR plot for attacker PYNQ-Z1 and victim ULX3S #1



(c) SNR plot for attacker ULX3S #1 and victim PYNQ-Z1

Figure 11: SNR plots for the wire-connected experiments.

last setup, the roles are swapped such that the PYNQ-Z1 is the attacker and the ULX3S is the victim board.

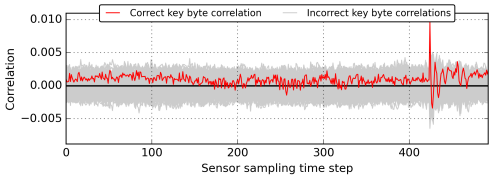
The SNR plots for the wire-connected attacks can be seen in Figure 11. The plots show a much stronger SNR for the unfiltered case during the last round of AES for all conditions compared to the PLL-filtered case. After evaluating the SNR results, we will now look at the CPA results.

First, we will look at the condition with the unfiltered input signal. The total-plots for the delay lines measuring the unfiltered signal Figure 12a, 12c and 12e show relatively large correlation values during the last round of AES. The corresponding progress-plots Figure 12b, 12d and 12f show the correct key byte being recovered after 318k traces for the experiment using two ULX3S boards and 317k trace for the experiment where the ULX3S is the attacker and the PYNQ-Z1 is the victim. For the last experiments where the PYNQ-Z1 is the attacker and the ULX3S is the victim, the key can already be recovered after 165k traces, which seems that the PYNQ-Z1 might have more side-channel leakage but also be able to observe leakage better.

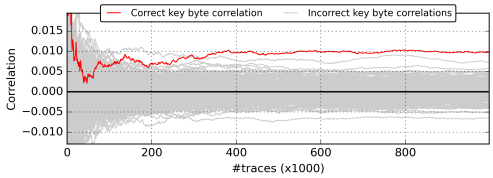
In contrast to the unfiltered signal, the total-plots for the delay line measuring the PLL-filtered signal in Figures 13a, 13c and 13e do not show clear signs of correlation. The red lines in the progress-plots in Figures 13b, 13d and 13f do not show any tendency and the correlation between the correct key byte power hypothesis and the measurements at the given time step seem to converge towards 0. This is true for all key bytes, and not only the shown byte 0. These traces could not be used to reconstruct the key, even after collecting 1 million traces.

In order to investigate further, TVLAs were conducted by measuring 1M trace pairs, where 1M fixed messages and 1M random messages are alternately encrypted. Please note that typically more traces are recommended to show an absence of leakage, but this mainly acts as a crosscheck. The results for these can be found in Figure 14. The threshold in the total-plots for whether or not the signals is leaking is computed following a mini-p procedure. Traditionally, a significance level of 10^{-5} is required for the signal to be classified as leaky, so the threshold amounts to 5.61 for traces of length 494. For the progress-plots, the HC-procedure is employed. Here, the threshold is 334.

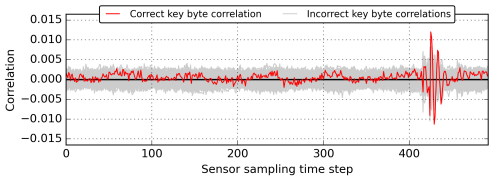
For the condition with two ULX3S boards shown in Figure 14a and its progress-plot in Figure 14b, we notice values well above the significance levels of 10^{-5}, thus the leakage



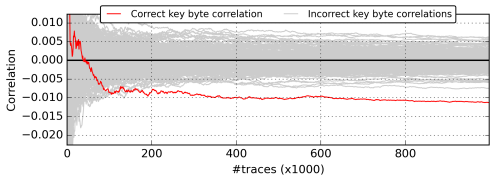
(a) CPA total-plot for attacker ULX3S #1 and victim ULX3S #2



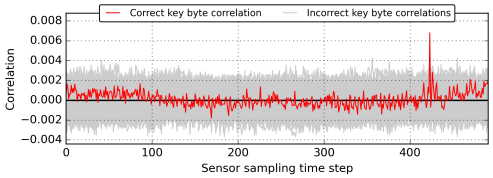
(b) CPA progress-plot for attacker ULX3S #1 and victim ULX3S #2; the correct key byte leaks after 318k traces



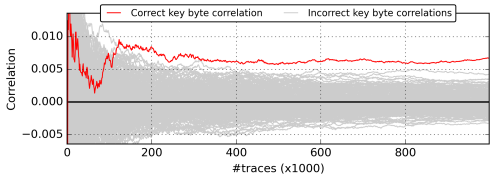
(c) CPA total-plot for attacker PYNQ-Z1 and victim ULX3S #1



(d) CPA progress-plot for attacker PYNQ-Z1 and victim ULX3S #1; the correct key byte leaks after 165k traces



(e) CPA total-plot for attacker ULX3S #1 and victim PYNQ-Z1



(f) CPA progress-plot for attacker ULX3S #1 and victim PYNQ-Z1; the correct key byte leaks after 317k traces

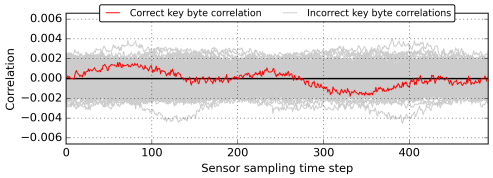
Figure 12: CPA plots over 1 million traces for the delay lines measuring the unfiltered signal transmitted over single wires. The red lines represent the correct key guesses, while the gray lines represent the incorrect key guesses.

assessment is successful. The total-plot Figure 14a even shows some of the structure of the victim's computation. Before and after the encryption process, the victim does idle waiting. This can be seen between time steps 0 and about 80 and at the end of the total-plot. The reason for this assessed leakage could well be the fact that the PLL is no perfect jitter filter, furthermore there is still a ground connection between attacker and victim, even though they do not share a power supply.

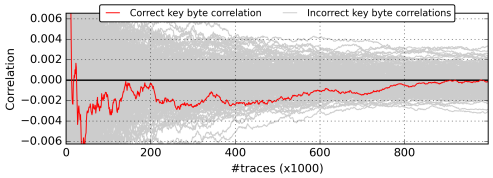
5.3 HDMI and isolated HDMI Attacks

Finally, we look at results for the HDMI setup. Here, the PYNQ-Z1 is used as an attacker and the ULX3S #1 is the victim. We will discuss the results of the attack over a standard HDMI-cable and the attack over the galvanic HDMI isolation module EVAL-CN0422 in tandem, as both are similarly successful. In addition to the EVAL-CN0422 for galvanic isolation, the victim board as well as the isolation board are placed inside an EM-isolated thermal chamber. Attacker, isolation board and victim are each powered by their own battery in order to eliminate any possibility of interaction through power lines.

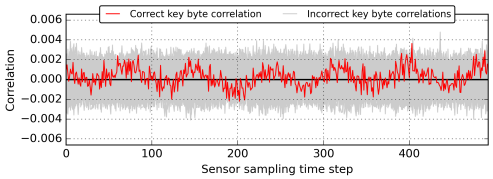
For the HDMI based attacks, the SNR plots can be seen in Figure 15. Similar to the wire-connected experiments, the plots show a much stronger SNR for the unfiltered case during the last round of AES for all conditions compared to the PLL-filtered case. The



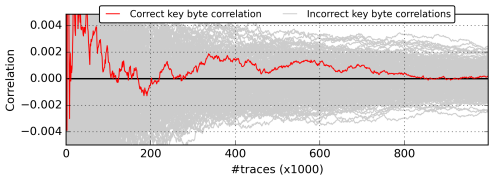
(a) CPA total-plot for attacker ULX3S #1 and victim ULX3S #2



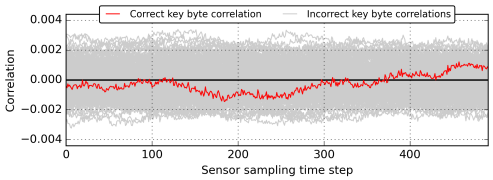
(b) CPA progress-plot for attacker ULX3S #1 and victim ULX3S #2



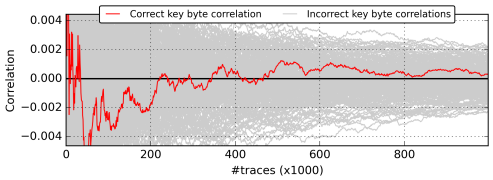
(c) CPA total-plot for attacker PYNQ-Z1 and victim ULX3S #1



(d) CPA progress-plot for attacker PYNQ-Z1 and victim ULX3S #1



(e) CPA total-plot for attacker ULX3S #1 and victim PYNQ-Z1



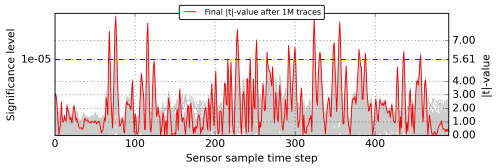
(f) CPA progress-plot for attacker ULX3S #1 and victim PYNQ-Z1

Figure 13: CPA plots over 1 million traces for the delay lines measuring the PLL-filtered signal transmitted over single signal/GND twisted pair wires. The red lines represent the correct key guesses, while the gray lines represent the incorrect key guesses.

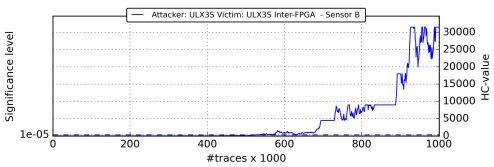
SNR plots support our hypothesis that there exists leakage for the unfiltered case, while there seems to be no (or much less) leakage for the PLL-filtered case. After evaluating the SNR results, we will now look at the CPA results.

Both the results on direct HDMI in Figure 16a and galvanically isolated HDMI in Figure 16c show relatively large correlation values during the last round of AES, even more than those observed for twisted pair wires in Section 5.2, which might be due to an increased signal quality through differential signalling in HDMI. The corresponding progress-plots in Figure 16b and Figure 16d clearly show the correct key byte in red, separating from the gray lines of incorrect keys, indicating a successful recovery of the 0th byte of the key. Even though our effective sampling rate is only half of the previous experiments (50 MHz vs 100 MHz), the correct key can be recovered after about 48k in both cases. Thus, the galvanic isolation chip does not seem to reduce the jitter side-channel leakage in the original signal at all.

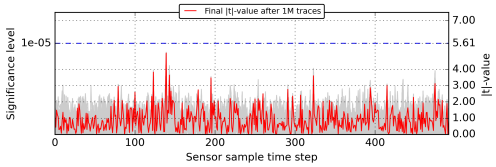
Despite we receive a NRZ-signal, we also put this signal through a PLL, which should work for clock-like phases (but not during communication phases). Here, in contrast, the total-plots for the PLL-filtered delay line in Figure 17a and Figure 17c do not show any signs of correlation. Also, the red lines of the correct key bytes in the progress-plots in Figure 17b and Figure 17d do not display any tendency and the correlation between the correct key byte power hypothesis and the measurements at the expected time step of the last AES round seem to converge towards 0. These traces could not be used to reconstruct the key, even after collecting 1 million traces. This confirms results from Section 5.2



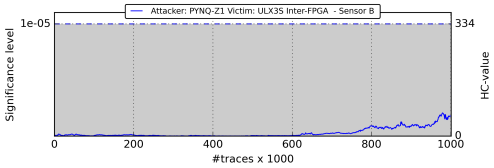
(a) TVLA total-plot for attacker ULX3S #1 and victim ULX3S #2



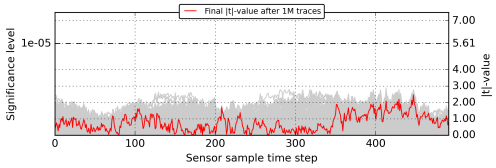
(b) TVLA progress-plot for attacker ULX3S #1 and victim ULX3S #2, leakage after about 550k



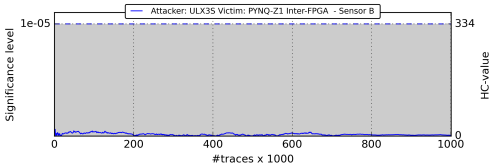
(c) TVLA total-plot for attacker PYNQ-Z1 and victim ULX3S #1



(d) TVLA progress-plot for attacker PYNQ-Z1 and victim ULX3S #1, no leakage can be reported



(e) TVLA total-plot for attacker ULX3S #1 and victim PYNQ-Z1



(f) TVLA progress-plot for attacker ULX3S #1 and victim PYNQ-Z1, no leakage can be reported

Figure 14: TVLA plots over 1 million traces pairs fixed/random for the delay lines measuring the PLL-filtered signal transmitted over single signal/GND twisted pair wires.

that PLL-filtering could effectively reduce the chance for a successful CPA if we are only analyzing a clock signal, but as TVLA has shown before, it cannot completely remove it. Furthermore, please note that the filter is implemented on the side of the attacker, the victim itself cannot effectively use a PLL to filter within their own power domain, as we have shown in Section 5.1.

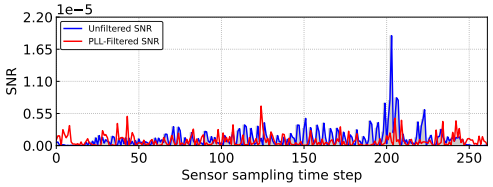
Since we already confirmed in Section 5.2 that the PLL cannot perfectly filter all side-channel leakage, we leave TVLA out in this experiment, since it would also add some difficulty to our experimental setup. These difficulties are the unidirectional communication on one hand, as well as the PLL that is connected to an actual data signal which would also contain fixed/random handshaking information that might keep the PLL in a state where it does not oscillate correctly for multiple clock cycles.

In total, this last experiment confirms our main hypothesis: Signal jitter in communication links leak side-channel information from the transmitter of this signal and can become a victim to this new class of side-channel attacks.

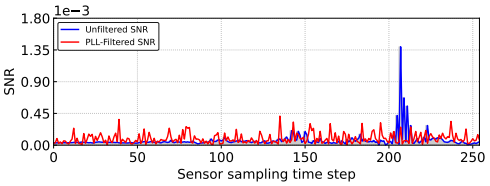
5.4 Summary of Results

In Table 2 we summarize our results. This overview shows that on-chip attacks are equally successful whether jitter is filtered with a PLL or not, which is probably due to the PLL again being influenced by on-chip voltage fluctuations in the same way the original clock source is. For all the experiments that were cross device, filtering the jitter can reduce the chance of a successful attack.

These results clearly confirm our original hypothesis. In fact the jitter in the signal is the reason for side-channel leakage across devices, while this cannot be prevented if a



(a) SNR plot for the signal transmitted using a standard HDMI-cable



(b) SNR plot for the signal transmitted via HDMI using an EVAL-CN0422 for galvanic isolation; victim and isolation board are placed inside an EM-isolated thermal chamber with only the HDMI cable exiting the chamber

Figure 15: SNR plots for the experiments connected via HDMI.

Table 2: Overview of our results for performing CPA with 1M traces and TVLA with 1M fixed/random trace pairs.

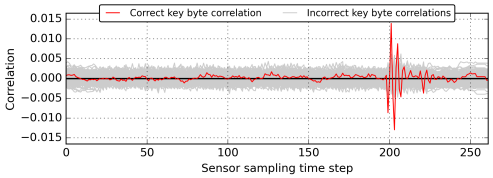
			Wire (Victim / Attacker)			**HDMI**	
		On-Chip	**ULX3S/ ULX3S**	**ULX3S/ PYNQ**	**PYNQ/ ULX3S**	**non-isolated**	**isolated**
CPA	unfiltered	∼27k	∼318k	∼165k	∼317k	∼47k	∼47k
	PLL-filtered	∼40k	failed	failed	failed	failed	failed
TVLA	unfiltered	Yes	Yes	Yes	Yes	n/a	n/a
	PLL-filtered	Yes	No	No	No	n/a	n/a

differential signal, or galvanically isolated signal is used. Since the differential HDMI link leads to faster key recovery than the single-ended twisted pair wire, we would even assume, that high quality data links will also transmit the jitter at higher precision.

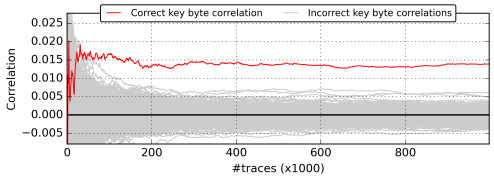
Table 3 shows descriptive statistics for the measured signal jitter values per experiment. We show the unit-less jitter from the sensors in which the mean is subtracted on a device-basis, as explained in Section 3.1. In all our experiments except for the on-chip condition the jitter's standard deviation decreases when filtering the signal using a PLL, indicating the voltage fluctuations having the most effect on the jitter. For the PYNQ-Z1 we can relate the results to a 11.5 ps estimate per bit, leading to a mean jitter of about 54±45 ps in the worst-case setup (isolated HDMI). Anyway, these statistical values of jitter cannot directly indicate whether a side-channel attack would be successful or not, when comparing the results in Table 3 to Table 2.

6 Discussion and Countermeasures

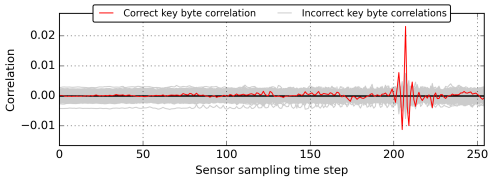
Our results clearly show that galvanic isolation through a differential signaling link still allow side-channel leakage in the form of signal jitter being transported across these devices. In contrast to a previous work by Schmidt et al. [SPK+10] who perform a voltage measurement on a constant high signal that is connected through an optocoupler, we rather perform timing measurements on a signal that varies its level, i.e. clock or NRZ data, significantly lowering the attack barrier. We thus observe a new side channel, which



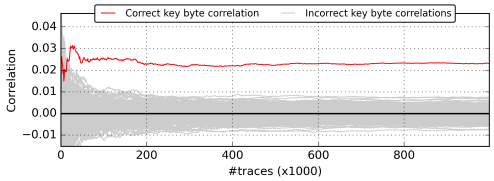
(a) CPA total-plot for the signal transmitted using a standard HDMI-cable,



(b) CPA progress-plot for the signal transmitted using a standard HDMI-cable; the correct key byte leaks after 47k traces



(c) CPA total-plot for the signal transmitted via HDMI using an EVAL-CN0422 for galvanic isolation; victim and isolation board are placed inside an EM-isolated thermal chamber with only the HDMI cable exiting the chamber



(d) CPA progress-plot for the signal transmitted via HDMI using an EVAL-CN0422 for galvanic isolation; victim and isolation board are placed inside an EM-isolated thermal chamber with only the HDMI cable exiting the chamber; the correct key byte leaks after 47k traces

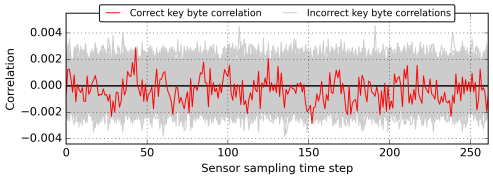
Figure 16: CPA plots over 1 million traces for the delay lines measuring the unfiltered signal transmitted over HDMI. The red lines represent the correct key guesses, while the gray lines represent the incorrect key guesses.

derives the supposedly power side channel leakage from the signal jitter it causes in an outgoing signal.

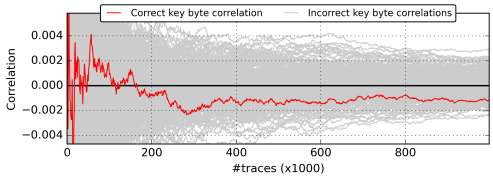
We showed that direct-to-device communication is clearly at risk, which goes beyond HDMI and can affect other interfaces such as USB 3.0, Ethernet, and even interfaces within a device, such as PCIe or SDRAM. Any synchronized communication channel that contains jitter may also be at risk, potentially also fiber-channel interfaces. Furthermore, we believe that even circuit-switched networks might not entirely prevent this attack from being successful, even over multiple hops, due to the absence of buffering that would eliminate jitter through time discretization.

We also tried to separate the effects from jitter and power by using a PLL that has a certain level of jitter rejection as a filter of the jitter at its input. This is relatively successful in that CPA does not work anymore on the filtered signal. Like that, we can show that the component of power leakage that affects the delay line directly was prevented by separating the power domains. However, please note that this filter is only an experimental vehicle that cannot work as a countermeasure, since it can only filter clock signals. Furthermore, so far, it was applied on the attacker side, a practical design would need to add a third galvanically isolated domain that acts as an independently powered filter for the victim.

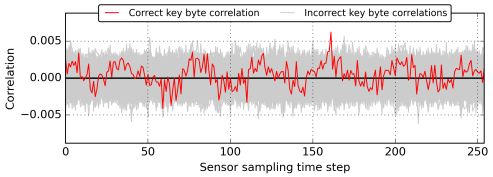
Since all of our experiments were performed using similar measurement designs and the same AES design, we can conclude that jitter leaks a very high degree of power side-channel information across boards. The required traces for successful key recovery are just about 2–10× more than those needed for FPGA-based on-chip power measurements, which is less of a difference than we expected. Interestingly, a higher quality connection through a differential link on the HDMI cable seems to allow for easier attacks than a simple twisted pair wire connection. While it was not the focus of this work, future experiments might be able to show that a significant degree of side-channel leakage in the on-chip attacks benefit from clock jitter, and not just the impact of voltage fluctuations on the delay line itself,



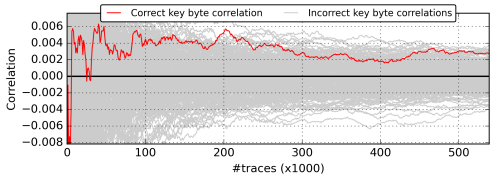
(a) CPA total-plot for the PLL-filtered signal transmitted using a standard HDMI-cable



(b) CPA progress-plot for the PLL-filtered signal transmitted using a standard HDMI-cable



(c) CPA total-plot for the PLL-filtered signal transmitted via HDMI using an EVAL-CN0422 for galvanic isolation; victim and isolation board are placed inside an EM-isolated thermal chamber with only the HDMI cable exiting the chamber



(d) CPA progress-plot for the PLL-filtered signal transmitted via HDMI using an EVAL-CN0422 for galvanic isolation; victim and isolation board are placed inside an EM-isolated thermal chamber with only the HDMI cable exiting the chamber

Figure 17: CPA plots over 1 million traces for the delay lines measuring the PLL-filtered signal transmitted over HDMI. The red lines represent the correct key guesses, while the gray lines represent the incorrect key guesses.

which is usually the assumption.

In summary, signal jitter can leak sufficient (power) side-channel information across boards, even without an actual power connection. By that, our work adds another new side-channel attack vector that was previously not addressed. Countermeasures that directly prevent this new side channel vector will need significant effort, as galvanic isolation cannot prevent it. If this attack vector gets developed further, millions of devices can be affected.

As a countermeasure, if actual messages and not a clock is transferred, we assume that a certain-level of buffering and re-transmitting the message to be transmitted seems to be a feasible suggestion to prevent the side channel, which is hard to do for circuit-switched communication networks. The countermeasure may be using only package-switched networks with at least one hop, like in most ethernet switches. After a hop, the jitter may be lost, which would prevent an adversary from carrying out a successful side-channel attack based on jitter. A very simple variant for such a countermeasure could also be adding a FIFO with depth 1 and two clocks with separated clock domains to decouple the jitter of the incoming signal from the outgoing signal. However, that FIFO would at least need to be on its own power domain, if not entirely EM-shielded.

Another aspect to discuss as a countermeasure is the use of data diodes [Fib23, Ind16, Gen23], which secure a system by allowing information to flow in only one direction, and are already used in medical and military domains. However, here it should be noted that some implementations of data diodes might still allow jitter side-channel attacks. While any data diode would still enforce an information flow in only one direction, it might not prevent that more data (in the form of jitter) is sent in that same direction. That is essentially shown by our last experiments over unidirectional HDMI. However, we think that more complex data diodes which also buffer data [Gen23], may effectively prevent jitter side-channels.

Overall, some of these isolation methods might work as a countermeasure, but further research is still required. Furthermore, it might still not be trivial to apply them in existing systems due to either practicability or cost issues.

Table 3: Overview of the jitter characteristics for all the experiments. All values are based on unit-less delay line sensor measurements. Jitter values are computed as the absolute difference between the sensor values and their mean for a given setup.

			Wire (Victim / Attacker)			**HDMI**	
		On-Chip	**ULX3S/ ULX3S**	**ULX3S/ PYNQ**	**PYNQ/ ULX3S**	**non-isolated**	**isolated**
unfiltered	min/max	0.36/6.64	0.11/16.11	0.15/16.15	0.42/23.58	0.41/26.59	0.38/27.38
	mean ± std.dev.	1.49±0.88	1.63±1.39	1.16±0.91	2.17±1.29	2.27±1.58	4.97±3.73
PLL-filtered	min/max	0.43/5.57	0.33/3.66	0.09/13.10	0.48/4.52	0.46/12.46	0.14/10.14
	mean ± std.dev.	1.86±1.31	1.56±0.62	0.51±0.45	1.46±0.75	0.63±0.39	0.27±0.43

7 Conclusion

Communication signals should not carry more information than what their specified bitrate allows. However, in this work we could show how minuscule timing variations of the rising and falling edges of a digital communication signal, also known as jitter, can carry sensitive information. First, we came up with a way to separate voltage- and jitter-based side-channel information. After that, we confirmed that we were able to reconstruct a secret key used for encryption using the Advanced Encryption Standard by performing Correlation Power Analysis on the traces carrying only jitter-based side-channel information. Finally, we showed that this side-channel information is available even if the signal in question is fed through a galvanic isolator and both devices have no common potential reference. The results of this work solidify the assumption that galvanic isolation is no countermeasure for side-channel attacks. In addition, it points out that digital signals contain side-channel information that could potentially be sent over various high speed signals such as optical links. This could become an additional security threat for millions of devices. Finally, it opens up a new medium for side-channel attacks that has not been explored before.

References

[Ana13] Analog Devices. Galvanic Isolation of the HDMI 1.3a Protocol Using iCoupler® Isolation Technology, 2013. https://www.analog.com/en/design-center/reference-designs/circuits-from-the-lab/CN0422.html.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. *Cryptographic Hardware and Embedded Systems (CHES)*, pages 16–29, 2004.

[BT11] Billy Bob Brumley and Nicola Tuveri. Remote timing attacks are still practical. In *European Symposium on Research in Computer Security*, pages 355–371. Springer, 2011.

[CPM+18] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. Screaming channels: When electromagnetic side channels meet radio transceivers. In *Conference on Computer and Communications Security (CCS)*. ACM, October 2018.

[DBN+01] Morris Dworkin, Elaine Barker, James Nechvatal, James Foti, Lawrence Bassham, E. Roback, and James Dray. Advanced encryption standard (aes), 2001-11-26 2001.

[DZD+18] A. Adam Ding, Liwei Zhang, Francois Durvaux, Francois-Xavier Standaert, and Yunsi Fei. Towards sound and optimal leakage detection procedure. In Thomas Eisenbarth and Yannick Teglia, editors, *Smart Card Research and Advanced Applications*, pages 105–122, Cham, 2018. Springer International Publishing.

[Fib23] Fibersystem. Ethernet isolator, January 2023. https://www.fibersystem.com/product-category/ethernet-isolator/.

[GDTM21] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia, and Philippe Loubet Moundi. Sideline: How delay-lines (may) leak secrets from your soc. In *Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, pages 3–30. Springer, 2021.

[Gen23] Genua. Highly Secure Industrial Monitoring. An industrial data diode for especially critical plants and processes, April 2023.

[GGJR+11] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[GKT19] Dennis R. E. Gnad, Jonas Krautter, and Mehdi B. Tahoori. Leaky noise: New side-channel attack vectors in mixed-signal iot devices. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2019(3):305–339, May 2019.

[GNST22] Daniel Genkin, Noam Nissan, Roei Schuster, and Eran Tromer. Lend me your ear: Passive remote physical side channels on PCs. In *USENIX Security Symposium (USENIX Security)*, pages 4437–4454, 2022.

[GRS20] Ilias Giechaskiel, Kasper Bonne Rasmussen, and Jakub Szefer. C^{3}apsule: Cross-fpga covert-channel attacks through power supply unit leakage. In *Symposium on Security and Privacy (SP)*, pages 1728–1741. IEEE, 2020.

[GYCH18] Qian Ge, Yuval Yarom, David Cock, and Gernot Heiser. A survey of microarchitectural timing attacks and countermeasures on contemporary hardware. *Journal of Cryptographic Engineering*, 8(1):1–27, 2018.

[GZBE19] Mordechai Guri, Boris Zadov, Dima Bykhovsky, and Yuval Elovici. Powerhammer: Exfiltrating data from air-gapped computers through power lines. *IEEE Transactions on Information Forensics and Security*, 15:1879–1890, 2019.

[Ind16] Industrial Control Systems Cyber Emergency Response Team. *Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies*. U.S. Department of Homeland Security, September 2016.

[Kal03] Jozef Kalisz. Review of methods for time interval measurements with picosecond resolution. *Metrologia*, 41(1):17, 2003.

[KGG+18] Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre attacks: Exploiting speculative execution. *ArXiv e-prints*, January 2018.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology*, pages 388–397. Springer Berlin Heidelberg, 1999.

[KMMS21] David Knichel, Amir Moradi, Nicolai Müller, and Pascal Sasdrich. Automated generation of masked hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):589–629, November 2021.

[Koc96] Paul C. Kocher. *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, pages 104–113. Springer, Berlin, Heidelberg, 1996.

[KP13] Thomas Korak and Thomas Plos. Applying remote side-channel analysis attacks on a security-enabled nfc tag. In *Cryptographers' Track at the RSA Conference*, pages 207–222. Springer, 2013.

[LAP+11] Charles Lamech, James Aarestad, Jim Plusquellic, Reza Rad, and Kanak Agarwal. REBEL and TDC: Two embedded test structures for on-chip measurements of within-die path delay variations. In *International Conference on Computer-Aided Design (ICCAD)*, pages 170–177. IEEE/ACM, 2011.

[LKO+21] Moritz Lipp, Andreas Kogler, David Oswald, Michael Schwarz, Catherine Easdon, Claudio Canella, and Daniel Gruss. PLATYPUS: Software-based power side-channel attacks on x86. In *Symposium on Security and Privacy (SP)*, pages 355–371. IEEE, 2021.

[MDL+22] Shayan Moini, Aleksa Deric, Xiang Li, George Provelengios, Wayne Burleson, Russell Tessier, and Daniel Holcomb. Voltage sensor implementations for remote power attacks on fpgas. *Transactions on Reconfigurable Technology and Systems (TRETS)*, 2022.

[MGKT22] Sergej Meschkov, Dennis R E Gnad, Jonas Krautter, and Mehdi B Tahoori. New approaches of side-channel attacks based on chip testing methods. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 2022.

[MKP12] Amir Moradi, Markus Kasper, and Christof Paar. Black-box side-channel attacks highlight the importance of countermeasures. In *Cryptographers' Track at the RSA Conference*, pages 1–18. Springer, 2012.

[MSEH20] Daniel Moghimi, Berk Sunar, Thomas Eisenbarth, and Nadia Heninger. TPM-FAIL: TPM meets timing and lattice attacks. In *USENIX Security Symposium*, pages 2057–2073, 2020.

[OD19] Colin O'Flynn and Alex Dewar. On-device power analysis across hardware security domains. *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, pages 126–153, 2019.

[ODAF21] Shuji Ohira, Araya Kibrom Desta, Ismail Arai, and Kazutoshi Fujikawa. PLI-TDC: Super fine delay-time based physical-layer identification with time-to-digital converter for in-vehicle networks. In *Asia Conference on Computer and Communications Security (AsiaCCS)*, pages 176–186. ACM, 2021.

[SAZT18] Eman Salem, Hossam Labeb Abdelhalim Zekry, and Radwa Tawfik. Fpga implementation of 1000base-x ethernet physical layer core. *International Journal of Engineering & Technology*, 7(4):2106–2112, 2018.

[SGMT18a] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi, and Mehdi B. Tahoori. An inside job: Remote power analysis attacks on FPGAs. In *Design, Automation & Test in Europe (DATE)*, March 2018.

[SGMT18b] Falk Schellenberg, Dennis R. E. Gnad, Amir Moradi, and Mehdi B. Tahoori. Remote inter-chip power analysis side-channel attacks at board-level. In *International Conference on Computer-Aided Design (ICCAD)*, pages 1–7, Nov 2018.

[SPK+10] Jörn-Marc Schmidt, Thomas Plos, Mario Kirschbaum, Michael Hutter, Marcel Medwed, and Christoph Herbst. Side-channel leakage across borders. In *International Conference on Smart Card Research and Advanced Applications*, pages 36–48. Springer, 2010.

[Sze19] Jakub Szefer. Survey of microarchitectural side and covert channels, attacks, and defenses. *Journal of Hardware and Systems Security*, 3(3):219–234, 2019.

[Tel14] Teledyne LeCroy. *Understanding Jitter Calculations: Why Dj Can Be Less than DDj (or Pj)*, July 2014. https://teledynelecroy.com/doc/understanding-dj-ddj-pj-jitter-calculations.

[Wei21] Weisstechnik. EMC test chambers LabEvent, 2021. https://backend.weiss-technik.com/webapp/weisstechnik/multimedia-center/brochures/2021/Weiss-Technik-LabEvent-EMV-EN-1.pdf.

[WPH+22] Yingchen Wang, Riccardo Paccagnella, Elizabeth Tang He, Hovav Shacham, Christopher W Fletcher, and David Kohlbrenner. Hertzbleed: Turning power side-channel attacks into remote timing attacks on x86. In *USENIX Security Symposium (USENIX Security 22)*, pages 679–697, 2022.

[WXL+21] Meizhi Wang, Shanshan Xie, Ping Na Li, Aseem Sayal, Ge Li, Vishnuvardhan V Iyer, Aditya Thimmaiah, Michael Orshansky, Ali E Yilmaz, and Jaydeep P Kulkarni. Galvanically isolated, power and electromagnetic side-channel attack resilient secure aes core with integrated charge pump based power management. In *Custom Integrated Circuits Conference (CICC)*, pages 1–2. IEEE, 2021.

[YSG+19] Mengjia Yan, Read Sprabery, Bhargava Gopireddy, Christopher Fletcher, Roy Campbell, and Josep Torrellas. Attack directories, not caches: Side channel attacks in a non-inclusive world. In *Symposium on Security and Privacy (SP)*, pages 888–904. IEEE, 2019.

[ZS18] Mark Zhao and G. Edward Suh. FPGA-based remote power side-channel attacks. In *Symposium on Security and Privacy (SP)*, pages 805–820. IEEE, May 2018.

[ZZL+22] Zihao Zhan, Zhenkai Zhang, Sisheng Liang, Fan Yao, and Xenofon Koutsoukos. Graphics peeping unit: Exploiting em side-channel information of gpus to eavesdrop on your neighbors. In *Symposium on Security and Privacy (SP)*. IEEE, 2022.