Turn on, Tune in, Listen up: Maximizing Side-Channel Recovery in Time-to-Digital Converters

Colin Drewes
University of California San Diego
USA

Olivia Weng
University of California San Diego
USA

Keegan Ryan
University of California San Diego
USA

Bill Hunter
Georgia Tech Research Institute
USA

Christopher McCarty
Georgia Tech Research Institute
USA

Ryan Kastner
University of California San Diego
USA

Dustin Richmond
University of California, Santa Cruz
USA

ABSTRACT

Voltage fluctuation sensors measure minute changes in an FPGA power distribution network, allowing attackers to extract information from concurrently executing computations. Previous voltage fluctuation sensors make assumptions about the co-tenant computation and require the attacker have *a priori* access or system knowledge to tune the sensor parameters statically. We present the open-source design of the Tunable Dual-Polarity Time-to-Digital Converter, which introduces three dynamically tunable parameters that optimize signal measurement, including the transition polarity, sample window, frequency, and phase. We show that a properly tuned sensor improves co-tenant classification accuracy by 2.5× over prior work and increases the ability to identify the co-tenant computation and its microarchitectural implementation. Across 13 varying applications, our techniques yield an 80% classification accuracy that generalizes beyond a single board. Finally, our sensor improves the ability of a correlation power analysis attack to rank correct subkey values by 2×.

CCS CONCEPTS

• **Security and privacy → Side-channel analysis and countermeasures**.

KEYWORDS

Power Side-Channel Attacks, Information Leakage, Circuits

ACM Reference Format:
Colin Drewes, Olivia Weng, Keegan Ryan, Bill Hunter, Christopher McCarty, Ryan Kastner, and Dustin Richmond. 2023. Turn on, Tune in, Listen up: Maximizing Side-Channel Recovery in Time-to-Digital Converters. In *Proceedings of the 2023 ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA '23), February 12–14, 2023, Monterey, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3543622.3573193

FPGA '23, February 12–14, 2023, Monterey, CA, USA.

ACM ISBN 978-1-4503-9417-8/23/02.
https://doi.org/10.1145/3543622.3573193

1 INTRODUCTION

Cloud providers offer FPGAs as a service. FPGA's versatility makes them efficient compute engines for neural networks [8], genome sequencing [3], secure database transactions [1], networking [31], and homomorphic encryption [29]. These applications have strict requirements for data confidentiality and computational integrity.

FPGA cloud providers use strict time-sharing schemes where a user rents the entire FPGA. This can leave the FPGA under-utilized. FPGA virtualization maximizes utilization by supporting multiple concurrent users [41]. It can reduce costs and increase efficiency, making it an attractive option for cloud service providers.

Unfortunately, FPGA virtualization introduces a side channel observable by an attacker implementing a voltage fluctuation sensor within their programmable logic. Voltage fluctuation sensors measure minute voltage changes in the power distribution network that expose details about co-tenant computations. Voltage fluctuation sensors are used as a covert channel [35, 42] or a side channel to extract cryptographic keys of co-located encryption cores [35, 42].

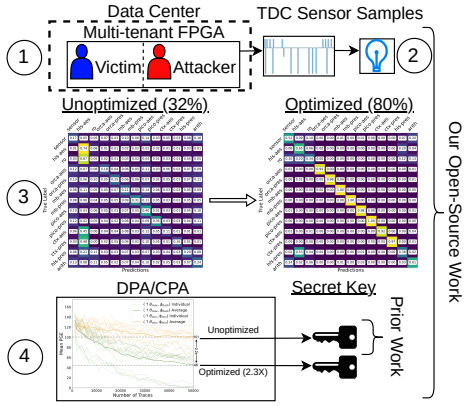


Figure 1: ① An attacker spatiotemporally locates with a victim. ② The attacker instantiates our Tunable Dual-Polarity TDC. ③ Our dynamic tuning techniques improve the ability to classify victim co-tenant computation by 2.5×. ④ After recognizing a cryptographic core, dynamic tuning increases the effectiveness of correlation power analysis by 2.2×.

Time-to-Digital Converters (TDCs) are a common voltage fluctuation sensor that measure the propagation delay through a linear

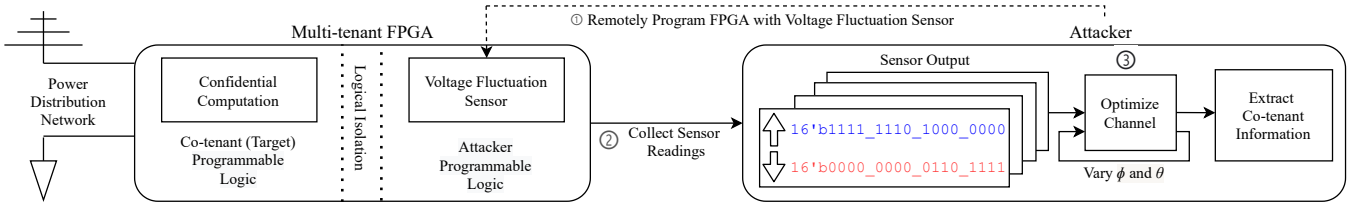


Figure 2: Remote TDC Threat Model: ① An attacker is given access to a remote multi-tenant FPGA and programs it with a voltage fluctuation sensor. ② The sensor readings are gathered and sent to the attacker for analysis. ③ The attacker tunes the parameters θ and ϕ to better extract co-tenant information. This paper studies the impact of θ and ϕ tuning.

array of logic elements, which is a function of the *power distribution network (PDN)* voltage. A slower propagation indicates that the PDN is stressed by some computation. These voltage fluctuations over time can be measured with consecutive TDC output captures.

Figure 1 shows our open-source pipeline. In Stage ①, an attacker co-locates temporospatially with a victim user. The attacker measures the shared power distribution network in Stage ②. Our open-source *Tunable Dual-Polarity TDC* allows the attacker to dynamically tune its sensing parameters, including transition polarity, sample window, frequency, and phase. Previously proposed TDC sensors are statically tuned in one or more of these parameters, which requires detailed knowledge of the computational environment and target computation. Utilizing these techniques in Stage ③, we demonstrate that a well-tuned sensor can improve classification accuracy by 2.5× over a statically-tuned sensor that incorrectly characterizes its environment or target computation. After successfully classifying an AES computation, we demonstrate in Stage ④ that proper sensor calibrations increase the ability to correctly rank subkey values by 2× in a Correlation Power Analysis (CPA) attack.

The contributions of this work are:

- An Open-Source Tunable Dual-Polarity TDC sensor for performing side-channel attacks on FPGAs
- A study of three metrics for measuring the propagation distance of rising and falling transitions
- A technique for maximizing channel information by adjusting capture window duration
- A method for tuning to the unknown phase of a co-tenant computation and isolating it from the environment
- A study characterizing the impact of these parameters on a 13-application, cross-board classification problem
- An application of our tuning methods to a multi-tenant Correlation Power Analysis attack

The paper is organized as follows: Section 2 presents the threat model. Section 3 describes our Tunable Dual-Polarity TDC and its tuning abilities. Section 4 experimentally verifies the tuning optimizations presented in the previous section, and then shows how this can be leveraged to perform our classification attack as well as a Correlation Power Analysis. We conclude in Section 6.

2 THREAT MODEL

Figure 2 describes the proposed threat model. The attacker is provided access to a cloud FPGA. The attacker has a design with a voltage fluctuation sensor and deploys it on the FPGA. We assume the system provides logical separation of the tenants [16, 21] and the attacker is restricted to system-defined interfaces, e.g., those provided by a shell. The attacker gathers the sensor readings, determines if a targeted co-tenant is present, and extracts confidential information from them. The attack is performed entirely remotely.

The attacker is a malicious adversary that aims to extract information from spatiotemporal co-tenants. This could be as simple as whether a co-tenant is currently using the FPGA, e.g., to know when to launch a fault attack [14, 19, 37]. The attacker could classify whether a specific type of computation is occurring on the shared FPGA, e.g., is the co-tenant performing encryption? It could infer details about the co-tenant's design, e.g., are they using a soft processor? Is it a RISC-V processor? The attacker could also learn information about the data being computed upon, e.g., extracting a cryptographic key [13, 34, 42], and leverage the architectural details learned about implementation to increase recovery speeds.

The attacker can implement a voltage fluctuation sensor. Our voltage fluctuation sensor is a variant of a TDC sensor [43]. We assume the sensor will pass bitstream analysis techniques that detect remote attacks [20]. As discussed later, our sensor passes the checks performed by Amazon AWS F1 instances.

We do not make any assumptions about where the sensor is placed, e.g., the victim computation does not need to have one of its wires running through it [12, 30, 32]. However, the sensors are more sensitive to computations that are spatially closer [17, 30], and so, as proximity decreases, demand for sensor tuning described in this increases. We consider only attacks within the same programmable logic. However, similar attacks have been shown from the FPGA to a CPU on the same die [42], across dies on a 2.5D integrated package [10], and across chips on the same board [11, 35].

3 TUNABLE DUAL-POLARITY TDC

Our Tunable Dual-Polarity TDC[1] has four key features: 1) it captures both rising and falling transition polarities (Dual-Edge); 2) it provides real-time adjustment of the sample window duration; 3) it provides real-time phase adjustment of the sample clock relative to the target computation; 4) it provides real-time frequency adjustment of the sample clock. We use these features to tune the sensor to the voltage fluctuations of the PDN caused by the target.

Figure 3 shows Tunable Dual-Polarity TDC architecture. The sensor's core is a pulse generator that induces rising and falling transitions through a delay line at a configurable frequency F_{sample}. A single pulse contains a positive ($0 \rightarrow 1$) and a negative ($1 \rightarrow$

[1]The sensor architecture and the sensor implemented alongside a PicoRV core executing AES has been open-sourced for the PYNQ-Z2. Additionally, we provide an easy-to-install PYNQ package for interacting with the sensor and an example Jupyter Notebook studying how ϕ shifting can be used for isolating relevant computation on the PicoRV running AES. All this is available at: https://github.com/KastnerRG/Tunable-TDC.

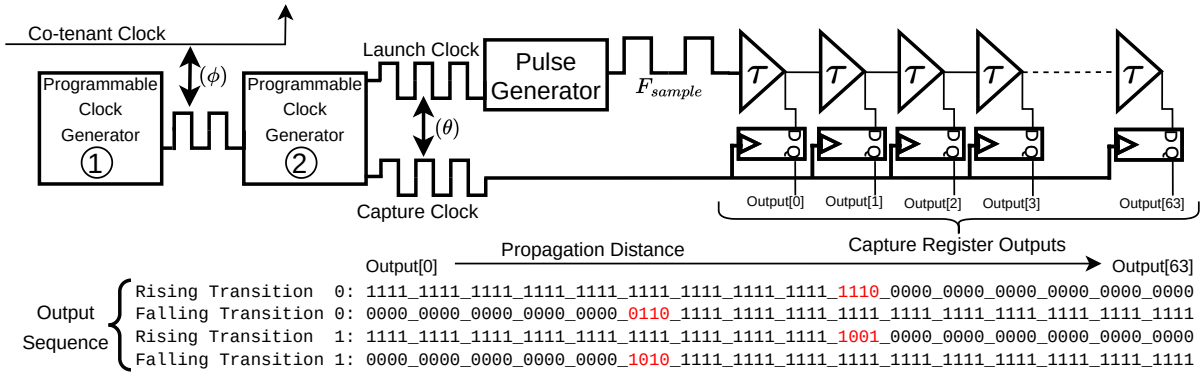


Figure 3: Tunable Dual-Polarity Time-To-Digital Converter Architecture: A pulse generator produces falling and rising transitions that propagate through delay line elements. Output capture registers record the transitions based on the capture clock. PDN voltage fluctuations affect the propagation speed and cause variations in the transition point (red).

0) pulse edge. Positive and negative pulse edges are issued sequentially in the `Launch Clock` domain. Pulse edges cause falling and rising transitions to propagate through a linear array of delay line elements to the `Output` capture register, which is controlled by the `Capture Clock`. θ is the phase difference between the `Launch Clock` and the `Capture Clock` – the time between the pulse launch and the subsequent capture of the transition in the output registers. When θ is set correctly, a transition will be propagating through the delay line when the output registers are clocked and record a metastable transition. The propagation distance is the number delay elements the transition has passed through.

An example output sequence from two consecutive pulses is shown at the bottom of Figure 3. Each pulse causes a falling and rising transition to be captured at the output. Rising Transition 0 shows that the $0 \rightarrow 1$ transition reached `Output[38]`.Falling Transition 0 shows that the $1 \rightarrow 0$ propagated to somewhere between `Output[21]` and `Output[23]`, with some metastability between the two points. In the next pulse, Rising Transition 1 propagates differently; the $0 \rightarrow 1$ transition propagates to between `Output[36]` and `Output[39]`. Similarly, Falling Transition 1 propagates to between `Output[20]` and `Output[23]`. These changes reflect PDN voltage fluctuations that change the delay line propagation. The variations provide potential information about the operation of the FPGA, including computation by co-tenants.

The sampling frequency is dictated by the length of the delay line and the speed of the underlying FPGA logic. If a higher effective sampling frequency is needed, multiple launch/capture clock pairs with a known phase offset can be generated by the clock generator as is done in related work [4, 36, 39].

Pulse Generator: The pulse generator produces positive ($0 \rightarrow 1$) and negative ($1 \rightarrow 0$) pulse edges that cause falling and rising transitions, respectively, in the delay line. Each sample produces a rising and a falling transition on the capture registers. A trace is a series of samples. The pulse generator has two configurable run-time parameters: the sampling frequency F_{sample}, which is an integer fraction of the launch clock frequency, and the number of pulses. Figure 3 demonstrates a trace length = 2, i.e., two rising and two falling transitions. We show that both transitions contain useful information in Section 4.

Programmable Clock Generators: The Tunable Dual-Polarity TDC has two programmable clock generators implemented using a Xilinx Mixed-Mode Clock Manager (MMCM). The first MMCM (Figure 3 ①) controls the input clock to the TDC and the phase relationship ϕ between the target clock and the sensor clock. Section 4.4 discusses the importance of tuning ϕ to capture relevant information about a co-located computation better.

A second MMCM (Figure 3 ②) generates the launch and capture clocks with a programmable phase offset, θ, between them. Changing θ modifies when the pulse generator generates an edge and when the capture clock fires and records the location of the subsequent transition in the output registers. Section 4.3 demonstrates the importance of tuning θ.

During compilation, the TDC sensor is configured to pass timing checks. The phase relationship ϕ is unconstrained, and θ is set to 2π. This means that the TDC sensor cannot be detected by tools that check for timing violations [20].

Delay Line and Capture Registers: The delay line in Figure 3 is a series of combinational logic elements that propagate the rising and falling transitions caused by the pulse generator. The delay elements are constructed from identical digital circuit elements that aim to provide a linear propagation delay, τ. The delay elements of the TDC should be placed and routed with uniform spacing to ensure consistent delay between each element and a uniform delay through the entire chain.

The Tunable Dual-Polarity TDC uses the fast look-ahead CARRY primitives in Xilinx FPGAs to create the delay line. The CARRY logic provides a relatively linear delay between each output bit within a single CARRY primitive. The carry logic is configured to compute `Output = 65'h0_ffff_ffff_ffff_ffff + input` so that when `input` changes from `65'h0` $\rightarrow$ `65'h1` on a positive pulse edge, the output of the delay line is a transition with falling polarity from `Output = 65'h0_ffff_ffff_ffff_ffff` to `Output = 65'h1_0000_0000_0000_0000`. A transition with rising polarity is produced on the negative pulse edge.

The interplay between the number of bits in the delay line and θ is also a critical TDC design consideration. The delay line length limits the maximum value of θ and the sampling frequency. A delay line that is too short may not capture all of the PDN variations

induced by a target, but a long delay line increases resource consumption. Characterizing how a target computation affects the PDN and the θ value that best measures variations are crucial for tuning the sensor to provide the most information.

The capture registers shown in Figure 3 record the output of each bit of the carry delay line in the capture clock domain. The path from the pulse generator to the high-order bit of the output meets timing constraints in the FPGA toolchain, and the launch clock and capture clock are configured to be in phase during compilation. This means that the TDC sensor cannot be detected by tools that check for timing violations [20].

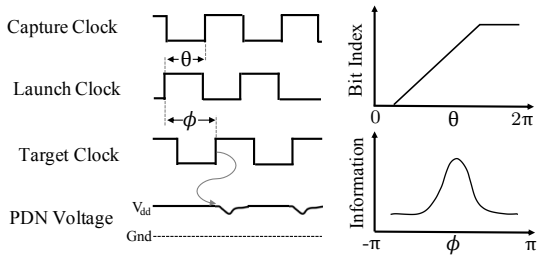


Figure 4: Tuning Parameters for our Tunable TDC Sensor. ϕ and θ define the relationship between the launch, capture and target clocks in our Tunable Dual-Polarity TDC. θ is the known phase relationship between the launch and capture clocks and affects the location of the transition bit index. ϕ is the unknown phase relationship between the launch and target clock. Variations in PDN voltage are caused by power consumption around the positive edge of the target clock. *Tuning* is the process of searching for $\phi \rightarrow 0$ such that power variations maximize the extracted information.

θ and ϕ Tuning: The programmable clock generators allow the Tunable Dual-Polarity TDC to tune its parameters to optimize the information sampled from the PDN. Figure 4 defines the relationship between the target clock, the capture clock, and the launch clock using θ and ϕ, the effect of the target computation on V_{dd}, and the effect of varying θ and ϕ on the extracted information. The PDN voltage V_{dd} varies in response to the target's rising clock edge.

The upper right graph in Figure 4 demonstrates the effect of varying θ from 0 to 2π. Increasing θ provides more time for the pulse to propagate through the delay line; as θ increases, the transition bit index increases. Section 4.3 experimentally demonstrates the importance of tuning θ.

The lower right graph in Figure 4 demonstrates the effect of varying ϕ from $-\pi$ to π. Changing ϕ will change the sampling window with respect to the target computation. When the sampling window is correctly positioned to the target clock, the sensor output will maximally change in response to the variations in current drawn by the target. This will cause an increase in the information measured at the sensor. Section 4.4 demonstrates how our TDC sensor enables ϕ to be tuned to ensure that the sample window is optimized with respect to the target computation.

Propagation Metric: When θ is tuned correctly, the capture clock will record how far the signal has propagated through the delay elements. The signal propagation distance can be measured as the index in the capture register. The least significant bits generally have their post-transition value, and the most significant bits typically have their pre-transition value. This imprecise definition reflects the metastability around the transition point that can cause multiple bit flips. This metastability may contain useful information, and ignoring these flips could reduce the side-channel information. This behavior is shown in rising/falling Transition 1 of Figure 3.

We examine three propagation metrics:

- *First Index*: The index of the first bit in `Output` that is not equal to `Output[0]`
- *Last Index*: The index of the last bit in `Output` that is equal to `Output[0]`
- *Binary Hamming Distance*: For rising transitions, the binary Hamming distance from `64'h_0000_0000_0000_0000`, and for falling transitions, the binary Hamming distance from `64'h_ffff_ffff_ffff_ffff`.

In Figure 3, the *First Index* metric produces the sequence 38, 20, 36, 19. The *Last Index* sequence is 38, 23, 39, 23. The *Binary Hamming Distance* sequence is 39, 22, 38, 22. Section 4.3 explores the efficacy of each metric.

4 RESULTS

We now report results on the impact of θ, ϕ, and propagation metrics, as applied to the classification experiment to determine if the co-tenant is a cryptographic core and, if so, perform a correlation power analysis. The sensor, classification data on 13 applications, and classifier network are released as open source.

4.1 Experimental Setup

Our experimental platforms are Amazon Web Services (AWS) EC2 F1 instances with Xilinx UltraScale+ XCVU9P-FLGB2104 FPGAs and six PYNQ-Z2 boards with Xilinx ZYNQ XC7Z020-1CLG400C FPGAs. On the PYNQ systems, the device is programmed with our sensor and test designs through the Python Productivity for Zynq (PYNQ) infrastructure. The AWS EC2 F1 instances are launched through the EC2 interface and programmed with the unique AGFI identifier associated with our sensor designs. The AGFI is generated by Amazon's unmodified compilation flow with the design checkpoint we provide. Our sensor has passed all design analysis techniques performed by AWS.

A 64-bit Tunable Dual-Polarity TDC is instantiated on PYNQ-Z2 and a 256-bit Tunable Dual-Polarity TDC on AWS. The launch and capture clock domains operate at 100 MHz. This results in a sampling rate, F_{sample}, of 25 MHz. MMCM ①, which allows for the phase shifting of F_{sample}, produces a 100MHz output clock. The internal F_{vco} is maximized for the two MMCMs so that the step granularity of θ and ϕ is maximized with a step size of 11.16 ps on AWS and 14.88 ps on PYNQ.

4.2 Applications

Our experiments use our Tunable Dual-Polarity TDC to classify the characteristics of a co-tenant. We have 13 unique applications containing IP cores using different architectural features. The application IP core and the sensor are implemented on the same FPGA. They are logically and physically isolated. The characteristics of the applications are described in the following paragraphs.

Sensor Only: The primary goal of the sensor-only design is to model the lack of another co-tenant. This design only contains the voltage fluctuation sensor and associated data collection logic. This mimics a scenario where only the attacker is present on the FPGA.

Ring Oscillators: Ring Oscillators are a malicious circuit with the sole purpose of aggressively consuming power. These are implemented as banks of combinational loops, resulting in rapid switching and power consumption as the circuit cannot settle on a single output value. Such a circuit can cause voltage disruptions in the power distribution network and can be used as a covert channel or to induce faults [14, 19, 37].

Arithmetic-Heavy: FPGAs are particularly well suited for high-intensity signal processing tasks with arrays of digital signal processors (DSPs). As an approximation of these structures, we implement arrays of DSPs performing a pipelined fused multiply-add operation. All DSPs operate in a single clock domain and compute upon data generated by a randomly-seeded, linear-feedback shift register.

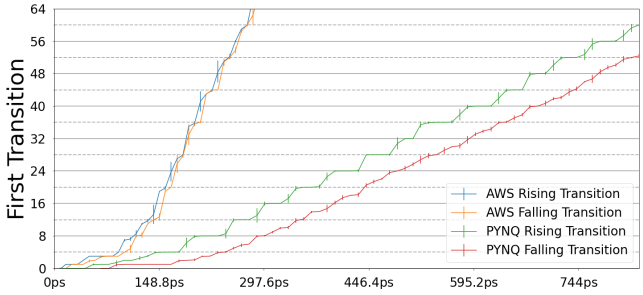
Cryptographic Cores: We study ten different implementations of cryptographic computations consisting of two algorithms (AES, PRESENT) implemented on five different architectures (Custom HLS IP core and as software running on Orca, MicroBlaze, PicoRV, and ARM CortexM3 soft processors).

4.3 θ Tuning and Metric Selection

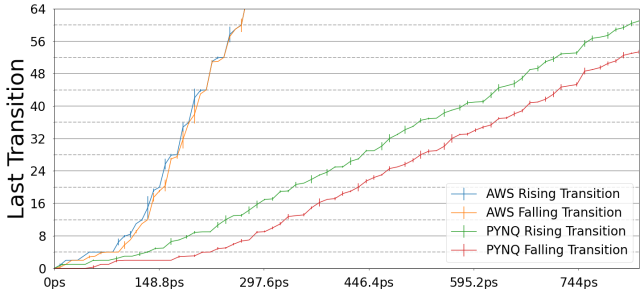
As shown in Figure 4, θ is the phase difference between the launch and capture clocks and dictates how long a transition is allowed to propagate through the delay line. It plays two important roles: first, θ determines the position of the transition in the output and can be used to avoid undesirable behavior caused by discontinuities in the FPGA architecture; second, θ defines the duration of the sampling window, when the delay line is measuring PDN variations.

Figure 5 demonstrates the effect varying θ has on the transition index as measured by the *First Index*, *Last Index*, and *Binary Hamming Distance* metrics across both falling and rising transition polarities. These experiments are performed on the *PYNQ-Z2 Sensor Only* and *AWS Sensor Only* designs. In the experiment θ is increased from 0 ps with a step size of 11.16 ps on AWS and 14.88 ps on PYNQ, as determined by the maximum F_{vco} frequency for the family and device speed grade. At each value of θ a trace of 2^{14} samples is captured, where a sample is one rising and one falling transition. This process is repeated until the transition index exceeds 64 bits, the maximum length of the delay line for our PYNQ-Z2 implementation. Next, we calculate the transition index using *First Index*, *Last Index*, and *Binary Hamming Distance* metrics, for each value of θ, for each trace, for both rising/falling transition polarities. The average value of the trace at each value of θ is plotted. Expressed as the error bar at each point is the standard deviation of the respective trace. Standard deviation, as we will show, is a good measure of the sensitivity of the sensor to voltage changes. The rising/falling transition polarities are shown in blue/orange for AWS and red/green for PYNQ. The three sub-graphs correspond to the three propagation metrics from Section 3 which are studied in the following sections.

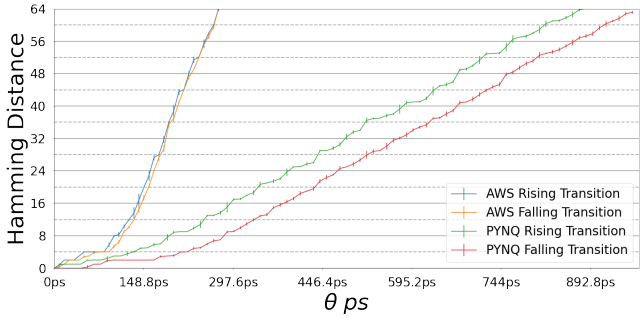
First Index: From Figure 5(a) we see that irregularities in the propagation of the rising transition are immediately apparent within both the PYNQ-Z2 and AWS. Plateaus where the transition index does not increase appear for ~40 ps (~5 ps) on PYNQ-Z2 (AWS) are divided by sloped regions where propagation is significantly



(a) *First Index* metric



(b) *Last Index* metric



(c) *Binary Hamming Distance* metric

Figure 5: Comparison of the three propagation metrics and two transition polarities as θ is increased from 0 ps. Vertical lines record the variance of a trace at each value of θ. The *First Index* metric is particularly susceptible to plateaus caused by the underlying CARRY4 (7-Series) and CARRY8 (UltraScale+) primitives that cause areas of low sensitivity but has high variance elsewhere. Falling transitions have fewer plateaus and less variance than their rising transition counterparts.

faster at $\sim.15\frac{bits}{ps}$ ($\sim.5\frac{bits}{ps}$) on PYNQ-Z2 (AWS). The sloped regions span four (eight) bits on PYNQ-Z2 (AWS) reflecting the underlying CARRY4 (CARRY8) primitives.

Last Index: Figure 5(b) demonstrates the behavior of the rising and falling transitions as measured by the *Last Index* metric on both PYNQ-Z2 and AWS. As demonstrated, plateaus are still present but they are less prominent and the standard deviation is more consistent across values of θ on the line. Noticeable plateaus still exist where the rising transition resembles that of the *First Index*. Plateaus are obvious on the AWS device and less on the PYNQ-Z2.

Binary Hamming Distance: Figure 5(c) demonstrates the behavior of the rising and falling transitions as measured by the *Binary Hamming Distance* metric on both PYNQ-Z2 and AWS. No prior method accounts for metastability within TDC output. For example,

if a rising transition falls within `Output[36:40]` as in Figure 3, neither prior metric is able to discern between `4b'0101` and `4b'0111`, potentially missing important information. The data in Figure 5(c) demonstrates that there are few plateaus when using the *Binary Hamming Distance* metric, and that the standard deviation is relatively consistent across the delay line.

The variable θ provides the ability to choose where in the delay line a transition falls, and therefore the ability to avoid plateaus we have observed in this section. For the remainder of the paper we use the *Binary Hamming Distance* metric for measuring rising and falling polarities due to its improved characteristics.

The delays of the carry outputs do not monotonically increase due to the use of carry lookahead adders in the FPGA architecture. Permuting the outputs allows the timing to be maintained [13]. This would change the behavior of the first/last index metric, making them more linear. It does not effect the *Binary Hamming Distance*.

4.4 ϕ Tuning and Background Subtraction

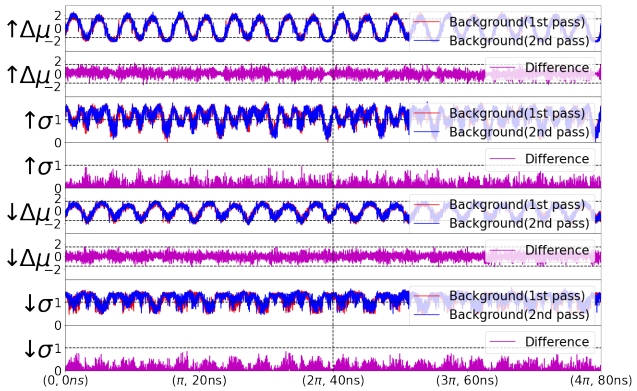
ϕ is the phase relationship between the target clock and the launch clock of the sensor. Our Tunable Dual-Polarity TDC can dynamically adjust ϕ to tune to the target clock and maximize measured information. This provides the ability to reliably isolate where information channel is maximized between the co-tenant and sensor. This has a significant impact on the side-channel information.

To demonstrate this, we sweep ϕ through two complete phase rotations (4π). For F_{sample} equal to 25 MHz this corresponds to 80ns. This process is performed twice: once as a measure of the background environment when the computation is disabled, and again when a co-tenant has been enabled. At each position of ϕ, two traces of 1024 samples are captured. One trace records the rising transition polarity ($\uparrow$) where θ maximizes the rising transition standard deviation samples and the other trace records the falling transition polarity ($\downarrow$) where θ maximizes the falling transition standard deviation samples. The *Binary Hamming Distance* is computed for each of these transition types. The average (μ) as well as standard deviation (σ) of each trace is calculated.

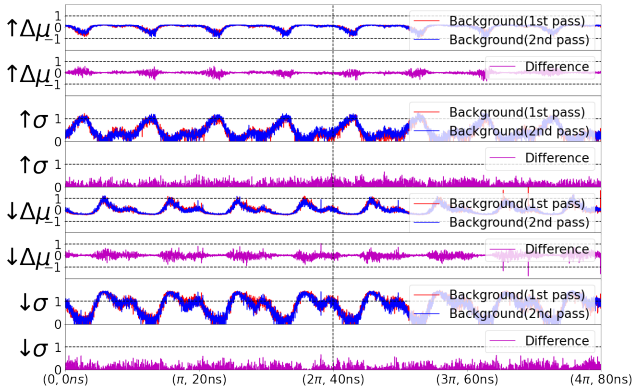
Figures 6(a), 6(c), and 6(b) demonstrate the result of sweeping ϕ over the range of 4π on three different designs: *AWS Sensor Only*, *PYNQ-Z2 Sensor Only* and *PYNQ-Z2 PicoRV AES*. The first and fifth row in each subfigure plot the zero-centered trace average for the rising transition ($\uparrow \Delta\mu$) and falling transition ($\downarrow \Delta\mu$). The raw offset in the *Binary Hamming Distance* is unimportant, so we consider the deviations from the average across all values of ϕ. The blue line is the data recorded when the computation is off (Background), and the red line is the data recorded when the target was on (if applicable). The second and the sixth row plot the point-wise difference between the red and the blue line in their respective preceding plots. The third and the seventh row in each subfigure plot the trace *Binary Hamming Distance* standard deviation (σ) for the rising transition ($\uparrow \sigma$) and falling transition ($\downarrow \sigma$).

The fourth and eighth plots are point-wise difference between the red and blue line in their respective preceding plots.

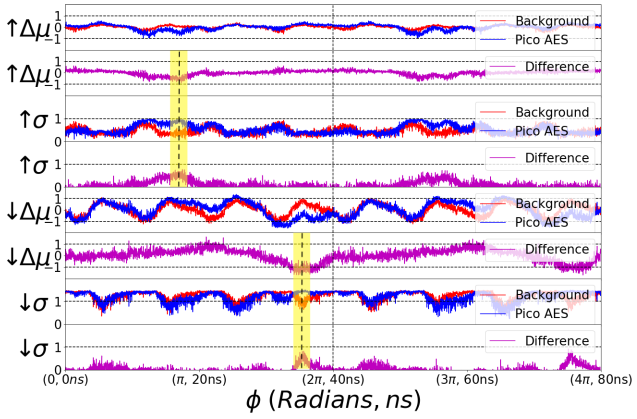
AWS Sensor Only: Figure 6(a) shows the behavior of the Sensor Only design on the AWS platform. The Background sweep is performed to record the environment. A second background sweep is then performed to determine whether background is consistent across multiple sweeps of ϕ. A clear signal emerges in the *Binary Hamming Distance* of both edges on both background sweeps (rows 1 and 5, $\uparrow \Delta\mu$ and $\downarrow \Delta\mu$). When the difference of the two ϕ sweeps is taken, the *Binary Hamming Distance* ($\uparrow \Delta\mu$ and $\downarrow \Delta\mu$) as well as the standard deviation ($\uparrow \sigma$ and $\downarrow \sigma$), is reduced to a flat line.



(a) AWS Sensor Only

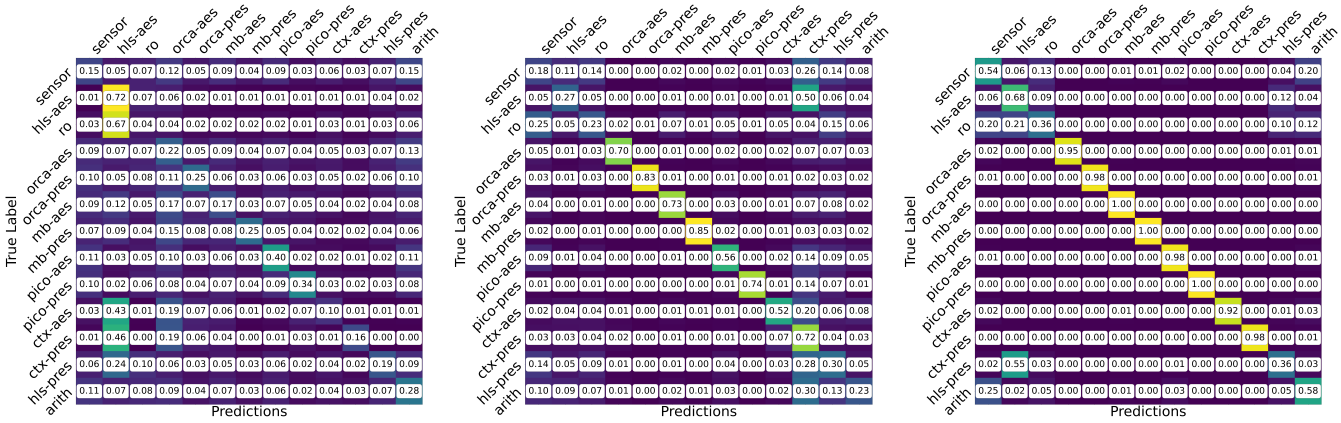


(b) PYNQ-Z2 Sensor Only



(c) PYNQ-Z2 PicoRV AES

Figure 6: Measuring sensor output as ϕ is swept from 0 to 4π. Background noise is consistent across multiple sweeps of ϕ, as demonstrated rows 3 and 6 of Figure 6(a) and 6(b). Background subtraction is critical to isolate the variance caused by a target from other information sources on the system and determine the value of ϕ that maximizes the information leakage (yellow). The rising ($\uparrow$) and falling ($\downarrow$) transition variance maxima are offset by π.



(a) ($\uparrow \theta_{min}, \phi_{min}$): 32% accuracy, Baseline (b) ($\downarrow \theta_{max}, \phi_{min}$): 51% accuracy, 1.6× (c) ($\downarrow \theta_{max}, \phi_{back}$): 75% accuracy, 2.3×

Figure 7: Our Tunable Dual-Polarity TDC is employed in a 13-way classification task where an attacker extracts the type of co-located computation. The ability to distinguish co-tenant computations is a measure of side-channel information contained in the sensor's traces. 7(a) represents the worst-case where a TDC cannot reconfigure ϕ and θ and achieves 32% accuracy. In 7(b) the TDC can tune θ and improves to 51% accuracy. In 7(c) both θ and ϕ have been tuned with background subtraction to isolate co-tenant information and achieve 75% accuracy, a 2.3× improvement.

The results demonstrate that there is significant background noise that has an effect on both the *Binary Hamming Distance* as well as the standard deviation of a trace. 20 peaks of equal amplitude appear over a range of 80 ns within the *Binary Hamming Distance*, which implies the existence of 250 MHz logic on the FPGA, likely the AWS shell logic which. Using background subtraction techniques [27] it can be removed to isolate the target.

PYNQ-Z2 Sensor Only: Figure 6(b) demonstrates the same experiment on the PYNQ-Z2 platform. We now observe background peaks that indicate 100 MHz synchronous logic. As on AWS, this information is consistent across multiple sweeps. When the background is subtracted, all variation in *Binary Hamming Distance* and standard standard deviation is reduced to a noisy flat signal.

PYNQ-Z2 PicoRV AES: Figure 6(c) demonstrates the same experiment performed on PYNQ-Z2 platform when the PicoRV AES design is operating at 25 MHz. In contrast to the previous two experiments, we take a single background sweep of ϕ with the PicoRV core deactivated, then another sweep of ϕ with the processor activated. The difference between the deactivated/activated ϕ sweeps produces a peak (yellow) that highlights the correct ϕ tuning.

Background subtraction produces a single distinct peak over a range of 2π in the *Binary Hamming Distance* ($\Delta\mu$) and standard deviation (σ) plots. We attribute this single peak to the PicoRV AES core running at 25 MHz. This behavior is consistent across designs, algorithms, and architectures. This position of ϕ represents not just where standard deviation is maximized (which may be muddied by the presence of background information), but where the channel contains maximum information about the co-tenant. We show in Section 4.5 that this is the best location for tuning the sensor and recovering side-channel information.

4.5 Effects of Tuning on Classification

As a precursor to cryptographic key recovery attacks, like a Correlation Power Analysis, an attacker must be able to determine what and when a cryptographic core is executing. We fill this void by demonstrating an attack where we accurately classify a co-tenant computation in a multi-tenant system.

Setup: As described in Section 2, we assume an attacker uploads a voltage fluctuation sensor to a remote multi-tenant FPGA environment to extract the architecture and algorithm of co-tenant computation through the comparison of captured power traces to a known body of labeled training data. This training data can be generated in two possible ways. First, a malicious actor, utilizing two separate user accounts, can instantiate a voltage fluctuation sensor with one user and attempt to co-locate with the second user, which instantiates a known design. This would allow an attacker to build a data set on the same architecture where the attack will be performed. The second option is to create a data set using local boards of the same type as the cloud environment. Because this choice depends on the implementation details of the multi-tenant model, we will not consider this in our analysis. Such an attack serves as a violation of the application anonymity guaranteed by such a multi-tenant system. The attack is performed on each of the 13 applications on 5 PYNQ-Z2 platforms.

θ Tuning: In the following experiment we consider four configurations of θ: $\uparrow \theta_{max}$, $\uparrow \theta_{min}$, $\downarrow \theta_{max}$, and $\downarrow \theta_{min}$. We sweep through the 64-bit delay line and record a trace's average and standard deviation at each point. The position where standard deviation has been maximized for a particular rising/falling transition polarity we call $\uparrow \theta_{max}$ / $\downarrow \theta_{max}$, and the point where standard deviation has been minimized for a particular rising/falling transition polarity we call $\uparrow \theta_{min}$ / $\downarrow \theta_{min}$.

ϕ Tuning: The sensor's ϕ will be configured three ways: first at a state of absolute maximum standard deviation (ϕ_{max}), at its absolute minimum standard deviation (ϕ_{min}), and finally the maximum standard deviation under the background subtraction (ϕ_{back}) process of Section 4.4. Just as in Section 4.4, ϕ is shifted in 14.88 ps increments 2688 times at each point, capturing a trace of 128 samples—once to capture background noise (σ_{back}), and again once the target

Table 1: Summary of Prior Work and Quantitative Comparison. Prior works implement a subset of our sensor's capabilities, which can be summarized as a tuning tuple (θ, ϕ) of our sensor. Each tuning tuple is tested in our classification experiment to determine accuracy and loss and perform a CPA Attack on a soft processor running AES to report the GSR @ 50K, PSR (Min, Avg, Max) @ 50K, and PGE (Min, Avg, Max) @ 50K traces. No prior experiments have measured how information learned on one board generalizes to other boards (called Cross-Board) —a crucial consideration for cloud attacks. Our tuning methodology and TDC Sensor improve co-tenant classification accuracy by 2.5× and increase the rate that correct subkey values are ranked as most likely (PSR) in a CPA attack by 2.2× relative to an un-tuned sensor.

Literature	($\uparrow \theta_{min}, \phi_{min}$)	($\downarrow \theta_{max}, \phi_{min}$)	($\downarrow \theta_{max}, \phi_{max}$)	($\downarrow \theta_{max}, \phi_{back}$)	($\uparrow \theta_{max}, \phi_{back}$)	Cross-Board	Class. %	Class. Loss	Class. Gain	GSR@50K	PSR@50K	PGE@50K	PGE@50K Gain
[14, 15, 34, 35]	✓						32.146%	2.159	0×	0%	(2%, 15%, 35%)	(66, 101, 124)	0×
[6, 7, 13, 17, 40, 43]		✓					51.160%	1.644	1.591×	NA	NA	NA	NA
This Work	✓					✓	32.146%	2.159	0×	0%	(2%, 15%, 35%)	(66, 101, 124)	0×
This Work		✓				✓	51.160%	1.644	1.591×	NA	NA	NA	NA
This Work			✓			✓	75.788%	0.834	2.358×	NA	NA	NA	NA
This Work				✓		✓	75.552%	0.733	2.350×	NA	NA	NA	NA
This Work					✓	✓	80.943%	0.668	2.518×	4%	(12%, 46%, 90%)	(3, 46, 73)	2.2×

application has been enabled (σ_{active}). In the tuning process, ϕ_{max} (ϕ_{min}) is the maximum (minimum) standard deviation of the σ_{active} sweep for a given transition type. The position of maximum standard deviation after background tuning (ϕ_{back}) is the maximum standard deviation of $\sigma_{active} - \sigma_{back}$.

Data Collection: The target design and sensor are loaded onto the device. Then, θ is positioned at one of $\uparrow \theta_{max}$, $\downarrow \theta_{max}$, $\uparrow \theta_{min}$, or $\downarrow \theta_{min}$. Finally, ϕ is configured to one of ϕ_{max}, ϕ_{min}, or ϕ_{back}.

We examine the following tuning combinations of (θ, ϕ): ($\uparrow \theta_{min}$, ϕ_{min}) emulates the worst-case of a non-tunable TDC. ($\downarrow \theta_{max}$, ϕ_{min}) introduces θ tuning to demonstrate how the mitigation of carry-chain non-linearity improves the sensor's ability to resolve co-tenant information. ($\downarrow \theta_{max}$, ϕ_{max}) demonstrates the significance of ϕ tuning on classification accuracy. ($\downarrow \theta_{max}$, ϕ_{back}) demonstrates how background subtraction improves our ability to optimize the co-tenant side channel. ($\uparrow \theta_{max}$, ϕ_{back}) is used to determine which transition polarity captures the most information, as it can be directly compared against ($\downarrow \theta_{max}$, ϕ_{back}).

After the sensor is configured, the target computation is launched, and a trace of 2^{16} samples are gathered. This process is repeated 100 times on each application for a total of 1300 traces per tuning combination per board. [2]

Post-processing: For a group of 1300 traces from a single tuning configuration (θ, ϕ) on a single board, we randomly segment each trace into ten sub-traces of 2^{13} samples. Each sub-trace is de-trended to remove the DC offset. The Fourier transform of the processed trace is then computed. From an original set of 1300 traces, we are left with 1000 rising transition Fourier transforms and 1000 falling transition Fourier transforms for each application, amounting to 26000 Fourier transforms per board per configuration.

Network Architecture: We train a simple neural network of only one fully connected layer, a fast and simplistic starting point. We evaluate the classification accuracy (how accurately our network can classify among the 13 classes of computation) and cross-entropy loss (how well our network generalizes to unseen data) on all configurations of training on four boards and testing on a 5th.

Classification Results: The results of our experiments are shown in Table 1, and select confusion matrices from our 13-way experiment are shown in Figure 7. The results are summarized:

($\uparrow \theta_{min}$, ϕ_{min}): The baseline dataset exhibits predictably poor performance in our classification task as shown in Table 1 with a 32% accuracy. The confusion matrix in Figure 7(a) demonstrates that the classifier struggles across all applications.

($\downarrow \theta_{max}$, ϕ_{min}): With the introduction of θ tuned to the maximum position, we see an immediate improvement in classification accuracy from 32% to 51% in Table 1, with the confusion shown in Figure 7(b). This shows that with proper θ tuning to avoid plateaus, measured information increases based on its ability to distinguish between soft processors and their applications.

($\downarrow \theta_{max}$, ϕ_{max}): With the introduction of ϕ tuning, accuracy improves to 75% in Table 1. The confusion matrix for this data set is shown in Figure 7(c) and robustly determines co-tenant application.

($\downarrow \theta_{max}$, ϕ_{back}): To evaluate the effects of background subtraction, we report our network's average accuracy and loss for ($\downarrow \theta_{max}$, ϕ_{max}) and ($\downarrow \theta_{max}$, ϕ_{back}). As seen in Table 1, the network performs 0.236% better without background subtraction; however, background subtraction decreases the loss (0.733 vs. 0.834). This indicates that our network generalizes better with background subtraction.

We expand our cross-validation configuration to investigate this result and understand how well the network generalizes to boards it has not trained, i.e., cross-board generalization. We train on all possible $\binom{5}{n} * (5-n)$ configurations, where $n \in [0, 5]$ is the number of training boards of data, and test on the remaining $(5-n)$ boards on both ($\downarrow \theta_{max}$, ϕ_{max}) and ($\downarrow \theta_{max}$, ϕ_{back}). We also train and test on the same board, as standard in prior work [15]. The results in Figure 8 show that when training and testing on the same board as in 'S', the network fits to artifacts of the dataset rather than the computation itself and does not generalize beyond the training board. As the number of training boards increases, the median accuracy increases, and the median loss decreases, demonstrating increased generalization.

[2] All of the data sets for each of the 13 applications, for each of the five boards, for each of the five tuning methods, along with the entire classification pipeline and network details are available at: https://github.com/KastnerRG/multitenant-classification

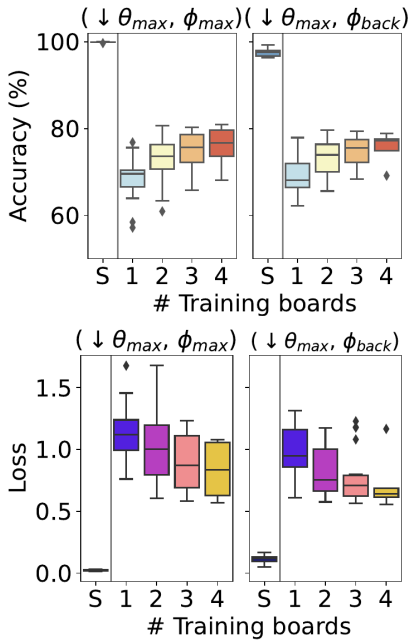


Figure 8: Multi-board Training and Inference Results. Plots show Accuracy and Loss when training on 1-4 boards with background subtraction ($\downarrow \theta_{max}, \phi_{back}$) and without ($\downarrow \theta_{max}, \phi_{max}$). Testing always occurs on data from a separate board, except for when data from the same board is used (denoted "S"). Background subtraction decreases the cross-board accuracy's interquartile range (IQR) by 2.3× and the loss's IQR by 5.8×. Multi-board training and background subtraction greatly improve cross-board generalization.

The distributions of accuracy and loss as we train on more boards behave differently when the network trains on data with background and without background subtraction. As seen in Figure 8(b), the interquartile range (IQR) decreases when background subtraction is added. When we train on four boards and test on a 5th, the Loss IQR without background subtraction is 0.429, whereas the IQR with background subtraction is 0.074, an improvement of 5.8×. The network is more likely to generalize to unseen boards with a smaller distribution. This is also reflected in the accuracy distribution in Figure 8(a). In the same four training board setup, the IQR of the accuracy with background subtraction is 2.3× smaller than without background subtraction.

($\uparrow \theta_{max}, \phi_{back}$): The use of the rising transition increases the accuracy from 75% to 80% and decreases the loss from .733 to .626. This indicates that the rising and falling transitions contain different information and that both transitions, when properly tuned, perform well in this classification task.

4.6 Effects of Tuning on CPA

After recognizing a cryptographic core with our classification procedure, we launch a Correlation Power Analysis (CPA) attack [2] to extract the key values. Because the values affect power consumption during encryption [34], our tuning techniques decrease the number of traces needed to extract the key.

Setup: We perform our CPA attack on the PYNQ-Z2 Orca AES application and consider the configurations ($\uparrow \theta_{max}, \phi_{back}$) for the well-optimized sensor and ($\uparrow \theta_{min}, \phi_{min}$) as a worst case un-tunable TDC comparison. The attack is repeated 50 times for each tuning strategy. We randomly generate a 128-bit AES key each time the attack is performed. This key is used by the Orca AES application to encrypt 50000 randomly generated plaintexts known to the attacker. During the encryption of each plaintext, the attacker collects a trace of length 8192, aligned by the measurement setup, so the beginning of the trace coincides with the beginning of the encryption.

A large body of work exists on performing attacks on reducing traces needed [5], alignment methods [9, 24, 28, 38], or filtration methods [25, 33]. We have kept our CPA attack as conventional as possible for a fair comparison of the different sensor configurations.

Following prior work, we analyze the results of the CPA attacks using multiple metrics [5]. After processing some traces, the CPA method returns a list for each subkey that ranks the possible subkey values from most likely to least likely. Partial Guessing Entropy (PGE) is the position of the correct subkey value in the list, where lower is better. Partial Success Rate (PSR) is the frequency with which the correct subkey value is ranked as most likely. Global Success Rate (GSR) is the frequency with which all correct subkey values are ranked as most likely. We also consider the mean PGE as a function of the number of traces. This is frequently used to compare the performance of CPA attacks [22, 23, 26].

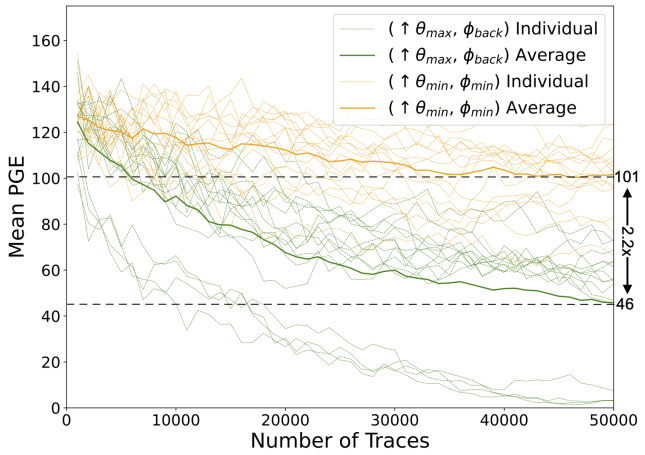


Figure 9: Performance of a CPA attack for ($\uparrow \theta_{max}, \phi_{back}$) and ($\uparrow \theta_{min}, \phi_{min}$) tuning parameters. Lower average PGE indicates that the attack is performing better, as the correct subkey values are ranked as more likely after processing fewer traces. Our tuning method has a significant impact on key recovery, lowering PGE by 2.2× at 50,000 traces.

CPA Attack Results: Our results demonstrate that the optimized sensor configuration ($\uparrow \theta_{max}, \phi_{back}$) outperforms the worst-case of an un-tunable sensor ($\uparrow \theta_{min}, \phi_{min}$). Qualitative results are given in Figure 9, which show that the traces obtained from ($\uparrow \theta_{max}, \phi_{back}$) exhibit lower PGE on average. This indicates that fewer traces are needed to recover the key when using the ($\uparrow \theta_{max}, \phi_{back}$) configuration, lowering the overall cost of the attack.

Numerical (Min, Avg, Max) results are given in Table 1. The GSR statistic shows that the optimized sensor configuration ($\uparrow \theta_{max}$,

ϕ_{back}) was able to recover all 16 subkeys in a single trial. In contrast, a poorly tuned sensor ($\uparrow \theta_{min}$, ϕ_{min}) never recovered the entire key. The higher PSR values for the well-tuned sensor demonstrate that individual subkeys were recovered around 2× more frequently given the same number of traces as with a poorly tuned sensor. These results show that optimized sensor configuration is crucial for identifying co-tenant computation and significantly increases the rate at which a cryptographic key can be recovered.

5 RELATED WORK

TDCs for Power Side-Channels: The Tunable Dual-Polarity TDC allows us to rapidly compare the performance of our tuning techniques to different "classes" of prior work. Table 1 summarizes prior and related work as configurations of our sensor.

The majority of these efforts never considered either θ tuning or ϕ tuning [14, 15, 34, 35]. Such sensors cannot achieve the signal resolution gains observed in Section 4. These sensors will not apply well to cloud-FPGA environments or across different FPGAs because they assume a random θ and ϕ on each device and therefore do not extract general computation elements.

Some prior work has introduced the concept of θ tuning [6, 7, 13, 17, 43]. However, many of these are limited in ability and applicability to the cloud-FPGA environment. For example, the method proposed in [7] involves re-configuring the number of delay elements to change where the transition reaches in the delay line. This does not generalize well to the cloud-FPGA model, as the delay elements need to be compiled into the design or updated with partial reconfiguration, making it difficult to respond quickly to changing conditions in a multi-tenant FPGA.

The authors of [17] consider another primitive variant of θ tuning, where they connect the pulse generator to several different places in the delay line through a set of multiplexers. This allows a user to shift θ by configuring the input location to the delay line. This is more runtime configurable but adds complexity to the design as the clock input must be duplicated and significantly limits the range of configurability as each position of θ needs to be predefined.

A more configurable approach for θ tuning is considered in [6], which leverages partial reconfiguration for modifying the routing between each delay element. This relies on that feature being exposed to the end user, which is potentially not an option in a cloud environment. This will be slower than our approach, which makes it difficult to adjust to rapidly changing conditions in cloud FPGAs (i.e., another user's design is allocated to the board).

The authors of [43] propose that if calibration of the TDC is required, the clock to the delay line can be phase-shifted using a programmable clock generator, as we have in this work. They do not expand on this nor consider how θ tuning can be leveraged to avoid irregularities in the delay line or adjust to the PDN load created by other users' designs.

The classification experiment we examined in this work is an essential consideration, as multi-tenant FPGA side-channel attacks often presuppose when and what computation is running alongside the attacker, e.g., the attacker assumes that a victim is performing a cryptographic operation. The first work to propose this [15] fails to generalize across FPGAs that training data was not collected on, making it incompatible with the cloud model. It cannot be generalized because it does not consider θ and ϕ, so it leans heavily on the architectural features of the device for its classification. Our work rectifies this limitation and addresses a fundamental optimization step that must be taken with power fluctuation sensors. The classification network used in [15], ResNet50, is significantly more complex yet performs worse than the simplistic single-layer network used in this paper. This is an important consideration if the network is to be implemented in hardware for fast recognition to launch a CPA attack if a cryptographic device is recognized.

To the best of our knowledge, there is no prior work demonstrating ϕ as we have done in Section 4.4. Alternative solutions that increase the sampling frequency of TDCs can reduce the importance of ϕ tuning (by increasing the likelihood a transition falls when there is activity in the PDN); this remains imprecise and limited as co-tenant frequency increases.

Mitigations: Physical isolation of co-tenants on the FPGA programmable logic [16, 21] mitigates attacks that require close physical access [12, 32]. Many remote attacks do not have such constraints on sensor placement [34, 42]. Our work reduces the benefits of physical isolation with a sensor designed to improve the signal-to-noise ratio through ϕ and θ optimization.

Active fences surround the co-tenant IP core with ring oscillators or other heavy-power-draw circuits [18], which induces noise into the PDN, making it harder to extract the signal. These techniques increase power and area consumption. Our sensor improves the signal-to-noise ratio making attacks more effective, through ϕ and θ optimization, as demonstrated in our results.

Krautter et al. [20] describe techniques that check the design for structures that resemble side-channel sensors. They focus on detecting sensors using ring oscillators, those that induce timing violations, data to clock paths, and high fanouts, which they argue are indicative circuits used in these threat models. Our sensor is resistant to these detection techniques as it has low fanout, no timing violations, and no combinational loops.

6 CONCLUSION

We present the Tunable Dual-Polarity TDC, which enables a first of its kind pipeline for recognizing co-tenant computation, maximizing recovered leaked information, and effectively extracting confidential information from a victim co-tenant. In a classification experiment with 13 applications, our techniques yield an 80% classification accuracy on 5-board, leave-one-out cross-validation, a 2.5× improvement over prior work. In addition, our sensor and tuning methodology improves the rate at which all correct subkey values are ranked as most likely by 2.2× in a CPA attack.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2038238. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Arvind Arasu, Ken Eguro, Manas Joglekar, Raghav Kaushik, Donald Kossmann, and Ravi Ramamurthy. 2015. Transaction Processing on Confidential Data using Cipherbase. In *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 435–446. https://doi.org/10.1109/ICDE.2015.7113304

[2] Eric Brier, Christophe Clavier, and Francis Olivier. 2004. Correlation Power Analysis with a Leakage Model. In *International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2004*, Marc Joye and Jean-Jacques Quisquater (Eds.). Springer, Berlin, Heidelberg, 16–29. https://doi.org/10.1007/978-3-540-28632-5_2

[3] Yu-Ting Chen, Jason Cong, Zhenman Fang, Jie Lei, and Peng Wei. 2016. When Spark Meets FPGAs: A Case Study for Next-Generation DNA Sequencing Acceleration. In *2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. 29–29. https://doi.org/10.1109/FCCM.2016.18

[4] Kyu-Jin Choi and Dong-Woo Jee. 2020. Design and Calibration Techniques for a Multichannel FPGA-Based Time-to-Digital Converter in an Object Positioning System. *IEEE Transactions on Instrumentation and Measurement* 70 (2020), 1–9. https://doi.org/10.1109/TIM.2020.3011490

[5] Christophe Clavier, Jean-Luc Danger, Guillaume Duc, M. Elaabid, Benoît Gérard, Sylvain Guilley, Annelie Heuser, Michael Kasper, Yang Li, Victor Lomné, Daisuke Nakatsu, Kazuo Ohta, Kazuo Sakiyama, Laurent Sauvage, Werner Schindler, Marc Stöttinger, Nicolas Veyrat-Charvillon, Matthieu Walle, and Antoine Wurcker. 2014. Practical Improvements of Side-Channel Attacks on AES: Feedback from the 2nd DPA Contest. *Journal of Cryptographic Engineering* 4 (11 2014), 259–274. https://doi.org/10.1007/s13389-014-0075-9

[6] Marc-Andre Daigneault and Jean Pierre David. 2011. A High-Resolution Time-to-Digital Converter on FPGA Using Dynamic Reconfiguration. *IEEE Transactions on Instrumentation and Measurement* 60, 6 (2011), 2070–2079. https://doi.org/10.1109/TIM.2011.2115390

[7] Claudio Favi and Edoardo Charbon. 2009. A 17ps Time-to-Digital Converter Implemented in 65nm FPGA Technology. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays* (Monterey, California, USA) *(FPGA '09)*. Association for Computing Machinery, New York, NY, USA, 113–120. https://doi.org/10.1145/1508128.1508145

[8] Jeremy Fowers, Kalin Ovtcharov, Michael Papamichael, Todd Massengill, Ming Liu, Daniel Lo, Shlomi Alkalay, Michael Haselman, Logan Adams, Mahdi Ghandi, Stephen Heil, Prerak Patel, Adam Sapek, Gabriel Weisz, Lisa Woods, Sitaram Lanka, Steven K. Reinhardt, Adrian M. Caulfield, Eric S. Chung, and Doug Burger. 2018. A Configurable Cloud-Scale DNN Processor for Real-Time AI. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. 1–14. https://doi.org/10.1109/ISCA.2018.00012

[9] Catherine H. Gebotys, Simon Ho, and C. C. Tiu. 2005. EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA. In *Cryptographic Hardware and Embedded Systems – CHES 2005*, Josyula R. Rao and Berk Sunar (Eds.). Springer, Berlin, Heidelberg, 250–264. https://doi.org/10.1007/11545262_19

[10] Ilias Giechaskiel, Kasper Rasmussen, and Jakub Szefer. 2019. Reading Between the Dies: Cross-SLR Covert Channels on Multi-Tenant Cloud FPGAs. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*. 1–10. https://doi.org/10.1109/ICCD46524.2019.00010

[11] Ilias Giechaskiel, Kasper Rasmussen, and Jakub Szefer. 2020. C3APSULe: Cross-FPGA Covert-Channel Attacks through Power Supply Unit Leakage. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. 1728–1741. https://doi.org/10.1109/SP40000.2020.00070

[12] Ilias Giechaskiel, Kasper B. Rasmussen, and Ken Eguro. 2018. Leaky Wires: Information Leakage and Covert Communication Between FPGA Long Wires. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (Incheon, Republic of Korea) *(ASIACCS '18)*. Association for Computing Machinery, New York, NY, USA, 15–27. https://doi.org/10.1145/3196494.3196518

[13] Ognjen Glamočanin, Louis Coulon, Francesco Regazzoni, and Mirjana Stojilović. 2020. Are Cloud FPGAs Really Vulnerable to Power Analysis Attacks?. In *Proceedings of the 23rd Conference on Design, Automation and Test in Europe* (Grenoble, France) *(DATE '20)*. EDA Consortium, San Jose, CA, USA, 1007–1010. https://doi.org/10.23919/DATE48585.2020.9116481

[14] Dennis R. E. Gnad, Fabian Oboril, and Mehdi B. Tahoori. 2017. Voltage drop-based fault attacks on FPGAs using valid bitstreams. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. 1–7. https://doi.org/10.23919/FPL.2017.8056840

[15] Mustafa Gobulukoglu, Colin Drewes, William Hunter, Ryan Kastner, and Dustin Richmond. 2021. Classifying Computations on Multi-Tenant FPGAs. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. 1261–1266. https://doi.org/10.1109/DAC18074.2021.9586098

[16] Ted Huffmire, Brett Brotherton, Gang Wang, Timothy Sherwood, Ryan Kastner, Timothy Levin, Thuy Nguyen, and Cynthia Irvine. 2007. Moats and Drawbridges: An Isolation Primitive for Reconfigurable Hardware Based Systems. In *2007 IEEE Symposium on Security and Privacy (SP '07)*. 281–295. https://doi.org/10.1109/SP.2007.28

[17] Jonas Krautter, Dennis Gnad, and Mehdi Tahoori. 2020. CPAmap: On the Complexity of Secure FPGA Virtualization, Multi-Tenancy, and Physical Design. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 3 (Jun. 2020), 121–146. https://doi.org/10.13154/tches.v2020.i3.121-146

[18] Jonas Krautter, Dennis R.E. Gnad, Falk Schellenberg, Amir Moradi, and Mehdi B. Tahoori. 2019. Active Fences against Voltage-based Side Channels in Multi-Tenant FPGAs.. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–8. https://doi.org/10.1109/ICCAD45719.2019.8942094

[19] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. 2018. FPGAhammer: Remote voltage fault attacks on shared FPGAs, suitable for DFA on AES. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018, 3 (Aug. 2018), 44–68. https://doi.org/10.13154/tches.v2018.i3.44-68

[20] Jonas Krautter, Dennis R. E. Gnad, and Mehdi B. Tahoori. 2019. Mitigating Electrical-Level Attacks towards Secure Multi-Tenant FPGAs in the Cloud. *ACM Trans. Reconfigurable Technol. Syst.* 12, 3, Article 12 (aug 2019), 26 pages. https://doi.org/10.1145/3328222

[21] Mark McLean and Jason Moore. 2007. FPGA-based single chip cryptographic solution. *Military Embedded Systems* (2007), 34–37.

[22] Colin O'Flynn and Zhizhang Chen. 2016. Power Analysis Attacks Against IEEE 802.15.4 Nodes. In *Constructive Side-Channel Analysis and Secure Design*, François-Xavier Standaert and Elisabeth Oswald (Eds.). Springer, Cham, 55–70. https://doi.org/10.1007/978-3-319-43283-0_4

[23] Colin O'Flynn and Zhizhang David Chen. 2015. Side channel power analysis of an AES-256 bootloader. In *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 750–755. https://doi.org/10.1109/CCECE.2015.7129369

[24] David Oswald and Christof Paar. 2011. Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World. In *Cryptographic Hardware and Embedded Systems – CHES 2011*, Bart Preneel and Tsuyoshi Takagi (Eds.). Springer, Berlin, Heidelberg, 207–222. https://doi.org/10.1007/978-3-642-23951-9_14

[25] David Oswald, Bastian Richter, and Christof Paar. 2013. Side-Channel Attacks on the Yubikey 2 One-Time Password Generator. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 204–222. https://doi.org/10.1007/978-3-642-41284-4_11

[26] Colin O'Flynn and Alex Dewar. 2019. On-Device Power Analysis Across Hardware Security Domains. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019, 4 (Aug. 2019), 126–153. https://doi.org/10.13154/tches.v2019.i4.126-153

[27] M. Piccardi. 2004. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, Vol. 4. 3099–3104 vol.4. https://doi.org/10.1109/ICSMC.2004.1400815

[28] Thomas Plos, Michael Hutter, and Martin Feldhofer. 2008. Evaluation of Side-Channel Preprocessing Techniques on Cryptographic-Enabled HF and UHF RFID-Tag Prototypes. In *Workshop on RFID Security*. Citeseer, 114–127.

[29] Thomas Pöppelmann, Michael Naehrig, Andrew Putnam, and Adrian Macias. 2022. Accelerating Homomorphic Evaluation on Reconfigurable Hardware. In *Cryptographic Hardware and Embedded Systems – CHES 2015*. Springer-Verlag, Berlin, Heidelberg, 143–163. https://doi.org/10.1007/978-3-662-48324-4_8

[30] George Provelengios, Daniel Holcomb, and Russell Tessier. 2019. Characterizing Power Distribution Attacks in Multi-User FPGA Environments. In *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 194–201. https://doi.org/10.1109/FPL.2019.00038

[31] Andrew Putnam, Adrian M. Caulfield, Eric S. Chung, Derek Chiou, Kypros Constantinides, John Demme, Hadi Esmaeilzadeh, Jeremy Fowers, Gopi Prashanth Gopal, Jan Gray, Michael Haselman, Scott Hauck, Stephen Heil, Amir Hormati, Joo-Young Kim, Sitaram Lanka, James Larus, Eric Peterson, Simon Pope, Aaron Smith, Jason Thong, Phillip Yi Xiao, and Doug Burger. 2015. A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services. *IEEE Micro* 35, 3 (2015), 10–22. https://doi.org/10.1109/MM.2015.42

[32] Chethan Ramesh, Shivukumar B. Patil, Siva Nishok Dhanuskodi, George Provelengios, Sebastien Pillement, Daniel Holcomb, and Russell Tessier. 2018. FPGA Side Channel Attacks without Physical Access. In *2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 45–52. https://doi.org/10.1109/FCCM.2018.00016

[33] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. 2017. IoT Goes Nuclear: Creating a ZigBee Chain Reaction. In *2017 IEEE Symposium on Security and Privacy (SP)*. 195–212. https://doi.org/10.1109/SP.2017.14

[34] Falk Schellenberg, Dennis R.E. Gnad, Amir Moradi, and Mehdi B. Tahoori. 2018. An inside job: Remote power analysis attacks on FPGAs. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1111–1116. https://doi.org/10.23919/DATE.2018.8342177

[35] Falk Schellenberg, Dennis R.E. Gnad, Amir Moradi, and Mehdi B. Tahoori. 2018. Remote Inter-Chip Power Analysis Side-Channel Attacks at Board-Level. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–7. https://doi.org/10.1145/3240765.3240841

[36] Zhipeng Song, Zhixiang Zhao, Hongsen Yu, Jingwu Yang, Xi Zhang, Tengjie Sui, Jianfeng Xu, Siwei Xie, Qiu Huang, and Qiyu Peng. 2020. An 8.8 ps RMS

Resolution Time-To-Digital Converter Implemented in a 60 nm FPGA with Real-Time Temperature Correction. *Sensors* 20, 8 (4 2020). https://doi.org/10.3390/s20082172

[37] Takeshi Sugawara, Kazuo Sakiyama, Shoei Nashimoto, Daisuke Suzuki, and Tomoyuki Nagatsuka. 2019. Oscillator without a Combinatorial Loop and its Threat to FPGA in Data Center. *Electronics Letters* 55 (04 2019). https://doi.org/10.1049/el.2019.0163

[38] Jasper G. J. van Woudenberg, Marc F. Witteman, and Bram Bakker. 2011. Improving Differential Power Analysis by Elastic Alignment *(CT-RSA'11)*. Springer-Verlag, Berlin, Heidelberg, 104–119. https://doi.org/10.1007/978-3-642-19074-2_8

[39] Yonggang Wang, Peng Kuang, and Chong Liu. 2016. A 256-channel multi-phase clock sampling-based time-to-digital converter implemented in a Kintex-7 FPGA. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*. IEEE, 1–5. https://doi.org/10.1109/I2MTC.2016.7520401

[40] Jun Yeon Won, Sun Il Kwon, Hyun Suk Yoon, Guen Bae Ko, Jeong-Whan Son, and Jae Sung Lee. 2016. Dual-Phase Tapped-Delay-Line Time-to-Digital Converter With On-the-Fly Calibration Implemented in 40 nm FPGA. *IEEE Transactions on Biomedical Circuits and Systems* 10, 1 (2016), 231–242. https://doi.org/10.1109/TBCAS.2015.2389227

[41] Yue Zha and Jing Li. 2020. Virtualizing FPGAs in the Cloud. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (Lausanne, Switzerland) *(ASPLOS '20)*. Association for Computing Machinery, New York, NY, USA, 845–858. https://doi.org/10.1145/3373376.3378491

[42] Mark Zhao and G Edward Suh. 2018. FPGA-Based Remote Power Side-Channel Attacks. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 229–244. https://doi.org/10.1109/SP.2018.00049

[43] Kenneth M. Zick, Meeta Srivastav, Wei Zhang, and Matthew French. 2013. Sensing Nanosecond-Scale Voltage Attacks and Natural Transients in FPGAs. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays* (Monterey, California, USA) *(FPGA '13)*. Association for Computing Machinery, New York, NY, USA, 101–104. https://doi.org/10.1145/2435264.2435283