AmpereBleed: Exploiting On-chip Current Sensors for Circuit-Free Attacks on ARM-FPGA SoCs

Xin Zhang[124†], Yi Yang[12†], Jiajun Zou[12], Qingni Shen[12*],
Zhi Zhang[3*], Yansong Gao[3], Zhonghai Wu[1], Trevor E. Carlson[4]
[1]School of Software and Microelectronics, Peking University
[2]PKU-OCTA Laboratory for Blockchain and Privacy Computing, Peking University
[3]The University of Western Australia
[4]National University of Singapore
Email: {zhangxin00, tkike, jjzou2002}@stu.pku.edu.cn, qingnishen@ss.pku.edu.cn,
{zhi.zhang, garrison.gao}@uwa.edu.au, wuzh@pku.edu.cn, tcarlson@comp.nus.edu.sg

Abstract—**FPGAs offer superior energy efficiency and performance in parallel computing but are vulnerable to remote power side-channel attacks. Existing attacks rely on assumptions of co-resident crafted circuits and shared power delivery networks, limiting their practicality in real-world scenarios. In this paper, we present AmpereBleed, a novel `current`-based side-channel attack that exploits widely available `INA226` sensors in ARM-FPGA SoCs, bypassing the aforementioned two assumptions. AmpereBleed achieves 261× greater variations to victim activities compared to the popular ring oscillator (RO) circuit, fingerprints DNN models on the Xilinx Deep Learning Processor Unit (DPU) with 99.7% accuracy, and distinguishes the Hamming weights of RSA-1024 keys.**

I. Introduction

Over the past few decades, Field Programmable Gate Arrays (FPGAs) have gained widespread adoption due to their superior energy efficiency and performance in parallel computing applications [14], [19], [20]. For example, when performing inference tasks on Llama 2 (a large language model), the Xilinx Virtex UltraScale+ VU9P FPGA achieves up to a 12.75× reduction in energy consumption per token compared to the Intel Xeon Broadwell E5-2686 v4 CPU, and an 8.25× reduction compared to the NVIDIA RTX 3090 GPU. In addition, it delivers up to 2.46× faster inference speeds than the CPU and reaches 53% of the GPU's speed, despite the GPU's significantly higher base clock rate [20].

Despite these advancements, FPGAs face various security threats, particularly from remote power side-channel attacks, which can extract sensitive information from a victim FPGA circuit without physical proximity to it [16], [22], [34], [37], [38], [43]. Specifically, an attacker crafts a circuit that can co-reside with a victim circuit on the same FPGA board, sharing the power delivery network (PDN). As the PDN supplies power to both circuits, increases in the victim's voltage (thus power consumption) cause transient voltage drops in the crafted circuit. Further, the voltage fluctuations in the crafted circuit lead to variations in signal propagation delays within itself and can be reflected in its output. For instance, as a representative circuit design, ring-oscillator (RO) [43] uses a combinational loop to increment a counter and sample the counter at fixed time intervals, thereby observing voltage through the increments of this counter.

Critical assumptions of state-of-the-arts: Notably, all the aforementioned power side-channel attacks rely on the assumption of co-residence between malicious and benign circuits, which is feasible only in multi-tenant FPGA scenarios where the PDN is shared among tenants. While FPGA multi-tenancy has gained significant attention from academia and industry as a promising approach to virtualize FPGA resources in the cloud [25], it is not yet adopted in real-world commercial practice [15]. Also, these attacks assume a shared PDN that fluctuates when the victim circuit is actively running, with voltage fluctuations observable by the attacker. To mitigate them, PDN stabilizers are implemented to ensure the FPGA supply voltage fluctuates within a limited range (e.g., 0.825 V to 0.876 V on the Zynq UltraScale+ series FPGAs). Additionally, recent works [10], [35] have proposed isolated PDN for each tenant (e.g., ISO-TENANT [10]).

As the critical assumptions significantly raise the bar for remote power side-channel attacks, this leaves an open research question: *Can a remote power side-channel attack against FPGAs still be feasible if these assumptions are voided?*

Our work: In this paper, we answer this question affirmatively by presenting AmpereBleed, a novel attack that bypasses these limitations. Our key observation is that many embedding devices are equipped with `INA226` sensors that provide measurements of `current`, `voltage`, and `power` for various hardware components. In the context of ARM-FPGA SoCs, these sensors monitor the FPGA board and provide fine-grained `current` measurements for the FPGA which are the most sensitive to FPGA activities among the three available measurements. Further, these measurements are accessible to an unprivileged process running on the ARM processor via the `hwmon` subsystem. Leveraging this, AmpereBleed constructs a `current`-based side channel to infer FPGA activities without relying on either crafted circuits or a shared PDN.

In our evaluation, we first compare AmpereBleed's `current` with the ring oscillator (RO) circuit [43] (a well-

† Co-first authors. * Corresponding authors.

known crafted circuit that senses voltage fluctuations), regarding 161 distinct victim activation conditions. The results indicate that AmpereBleed-based current measurements achieve 261× greater variations than RO. Further, we validate that AmpereBleed's current is much more sensitive than voltage and power under the same settings. Last, we exploit AmpereBleed in two case studies. First, we mount a DNN model fingerprinting attack to show that AmpereBleed can distinguish among 39 DPU accelerators with a high accuracy of 99.7%. Second, we successfully apply AmpereBleed to distinguish Hamming weights of secret keys from an RSA-1024 circuit operating at 100 MHz.

Summary of contributions: The main contributions of this paper are as follows:

- We present AmpereBleed, a new current-based remote power side-channel attack against ARM-FPGA SoCs, without relying on either crafted circuits or a fluctuating PDN. AmpereBleed highlights the need for secure power monitoring implementation.
- We characterize how current measurements leak fine-grained information about victim activities and compare them with crafted circuit (i.e., RO [43]), voltage measurements, and power measurements.
- We apply AmpereBleed to two case studies, which fingerprint different DPU accelerators and distinguish the Hamming weight of RSA keys.

The source code to reproduce our experiments is released at https://github.com/Skyofmine007/AmpereBleed-DAC.

II. Background and Related Work

A. ARM-FPGA SoCs

Recent years have seen a growing demand for both energy efficiency and high performance, which has significantly driven advancements in high-performance FPGAs. Unlike CPUs and GPUs, which accelerate by optimizing instruction streams, FPGAs provide substantial benefits for customized computing due to their inherent re-programmability and high-performance potential. For example, AMD Xilinx's high-end AI Adaptive Compute Acceleration Platform (ACAP) VCK5000 achieves 1.8× frames per second per watt compared to Nvidia's flagship Ampere GPU (A100 SXM) in a standard MLPerf [31] benchmark.

ARM-FPGA System-on-Chips (SoCs) integrate ARM processor cores with an FPGA to provide a versatile solution for embedded systems [14]. The ARM cores, akin to standalone ARM processors such as the STM32 series, are equipped with various peripherals including UART, CAN, and GPIO ports, while FPGA serves as a customizable peripheral. This tight integration allows for high-speed, low-latency communication between CPU and FPGA resulting in improved performance, reduced cost, and a smaller physical footprint compared to using separate chips. Today, ARM-FPGA SoCs have become increasingly popular in Internet of Things (IoT) applications for commercial purposes, including autonomous driving [4], [9], 5G telecommunications [11], and medical devices [3], [8]. Companies like Baidu [9] and Subaru [4] have adopted this architecture in their autonomous driving and driver-assistance systems, respectively. For instance, Subaru has sold over 1 million vehicles with over 50 different models, which are equipped with ARM-FPGA SoCs as a solution for driver-assistance systems [4], highlighting the practicality of these SoCs in real-world deployments.

B. Remote Power Side Channel Attacks against FPGAs

Power side channel attacks have long been recognized as a significant source of information leakage. By analyzing power consumption, an adversary can infer victim's activities and extract sensitive data. To enable remote power side channel attacks on FPGAs, existing studies have explored a number of crafted circuits [22], [34], [37], [38], [43] that can detect on-chip voltage fluctuations. Specifically, all these works assume that a crafted circuit co-resides with the victim circuit on the same FPGA board, thereby sharing the same PDN with the victim circuit. A fundamental vulnerability identified in these studies is the inverse correlation between signal propagation delay and voltage variations at gate level. When engineered with precision, a crafted circuit can exploit voltage fluctuations, resulting in varying bit flips in its output stream. This behavior enables the creation of a novel side channel. The adversary can then leverage this output to various end-to-end attacks, including extracting cryptographic keys [34], [38], [43], fingerprinting victim workloads [18], and stealing DNN model architectures [42].

III. AmpereBleed

A. Threat Model

Aligned with prior works [14], [30], [36], [44], we consider an unprivileged attacker controlling a user process without special privileges, achievable through malicious over-the-air (OTA) updates [28], [29] or malware installation [12], [24]. The attacker's process is co-located with the victim circuits on the same ARM-FPGA SoC equipped with INA226 sensors. The victim is a victim circuit with full control over the FPGA board for circuit deployment.

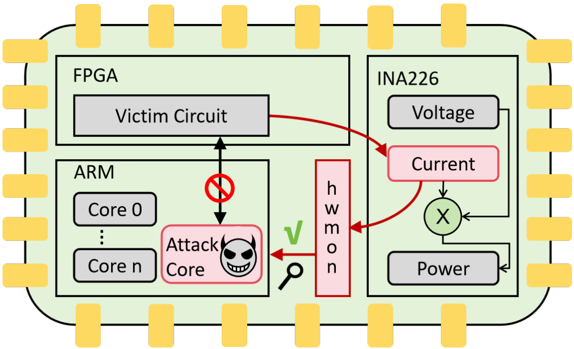


Fig. 1: Attack overview of AmpereBleed. The components inducing AmpereBleed's leakage are marked as red.

TABLE I: A comparison of the number of integrated INA226 sensors on various ARM-FPGA SoCs.

Property	ZCU102	ZCU111	ZCU216	ZCU1285	VEK280	VCK190	VHK158	VPK180
FPGA Family	Zynq UltraScale+	Zynq UltraScale+	Zynq UltraScale+	Zynq UltraScale+	Versal	Versal	Versal	Versal
FPGA Voltage (V)	0.825∼0.876	0.825∼0.876	0.825∼0.876	0.825∼0.876	0.775∼0.825	0.775∼0.825	0.775∼0.825	0.775∼0.825
CPU Model	Cortex-A53	Cortex-A53	Cortex-A53	Cortex-A53	Cortex-A72	Cortex-A72	Cortex-A72	Cortex-A72
DRAM	4 GB	4 GB	4 GB	8 GB	12 GB	8 GB	32 GB	12 GB
INA Sensors	18	14	14	21	20	17	22	19
Price ($)	3,234	14,995	16,995	32,394	6,995	13,195	14,995	17,995

B. Current-based Side Channel

As a first step for our analysis, we set out to review how existing attacks [16], [34], [43] monitor dynamic power consumption. Specifically, in a shared PDN that ideally ignores the effect of the stabilizers and can be modeled as an equivalent RLC matrix, an increase in the victim circuit's power causes transient voltage drops in neighboring circuits:

$$V_{drop} = I \cdot R + L \cdot \frac{\Delta I}{\Delta t}, \tag{1}$$

where V_{drop} depends on both steady-state current ($I \cdot R$ drop) as well as short transients ($\frac{\Delta I}{\Delta t}$ drop) caused by switching logic on the FPGA. It can be seen that current serves as an intermediate quantity of voltage change in these studies.

While previous studies have crafted various circuits [16], [22], [34], [37], [38], [43] to capture the voltage drop in the shared PDN, we note that dedicated PDN stabilizers [21] have been implemented to ensure the FPGA supply voltage fluctuates within a limited range, e.g., from 0.825 V to 0.876 V on the board with Zynq UltraScale+ series FPGAs (as shown in Table I). Besides, there are improvements in academia [10], [35] that aim at maintaining a stable voltage supply to each tenant circuit. However, it remains unclear whether the remote power side-channel threats can be fully mitigated after the voltage is completely stabilized.

Foundation: The foundation behind our work is rooted in an essential law of physics, where the overall dynamic power can also be measured by the supply voltage and current [1], as represented in Equation 2.

$$P_{dyn} = V_{dd} \cdot \sum I(LE, RAM, DSP, Clocks, \ldots), \tag{2}$$

where P_{dyn} is the dynamic power, V_{dd} is the supply voltage, and I is the current drawn by different types of computing elements, e.g., logic elements (LE), random access memory (RAM), digital signal processing (DSP), and clocks. Thus, even if the voltage remains stable, significant changes in power still cause noticeable changes in current.

To this end, we propose AmpereBleed, a novel remote power side channel attack that exploits current measurement to infer victim activities. As shown in Figure 1, AmpereBleed works on an ARM-FPGA SoC platform. Our attacker is located on one of the ARM cores without any co-resident circuit involved. To monitor on-chip current, we exploit INA226 sensors, which are integrated into a wide range of embedding devices for system monitoring, including ARM-FPGA SoCs. By accessing unprivileged current driver (i.e., hwmon) of INA226 sensors, AmpereBleed infers the activities of victim circuits through the red path.

TABLE II: Sensitive sensors that allow unprivileged access through the hwmon interfaces on the ZCU102 board [5].

Sensor	Description
ina226_u76	current, voltage, and power for full-power domain of the ARM processor cores.
ina226_u77	current, voltage, and power for low-power domain of the ARM processor cores.
ina226_u79	current, voltage, and power for FPGA's logic and processing elements.
ina226_u93	current, voltage, and power for DDR memory.

C. Unprivileged Sampling of On-chip Current

In this section, we provide implementation details about the unprivileged sampling of current. First, to demonstrate the widespread availability of the exploited INA226 sensors, we have examined a range of SoC boards across two FPGA families (i.e., Zynq UltraScale+ and Versal). Table I lists 8 representative boards, all of which include INA226 sensors. Notably, AmpereBleed's applicability extends well beyond these examples. For instance, the ZCU106 board, featuring 14 INA226 sensors, is not listed due to its similarity in price and release date to the ZCU102. Besides, Table II showcases four sensitive INA226 sensors out of the 18 integrated on the ZCU102, providing on-chip monitoring for various components including CPU, FPGA, and DRAM.

The INA226 sensors provide current and voltage measurements at specific monitoring points. They also calculate power indirectly from these two measurements when properly calibrated [2].

We identify current as the most potentially exploitable for two key reasons: First, for the victim FPGA workload, its voltage measurements are constrained by a fixed resolution of 1.25 mV and exhibit minimal changes (as shown in Table I, resulting in a limited range of voltage variations. Second, the power measurements are derived from current and voltage, with their resolution fixed at a ratio of 25 relative to the current resolution [2]. Given the limited variability in voltage, the power measurements are almost synchronized to the current measurements, but the low bits are truncated. In our evaluation, we compare current measurements against voltage and power measurements to validate the identification.

We use the hardware monitoring (also known as hwmon) subsystem to access the INA226 sensors from the ARM side without requiring any privileges. The subsystem is also available in x86 processors [23]. We access the

Each INA226 sensor is equipped with a dedicated shunt resistor, enabling current to be measured based on its voltage and resistance.

current measurements on each INA226 sensors through /sys/class/hwmon/hwmon[0-*]/curr1_input in the Linux file system. Notably, these interfaces provide a resolution of ±1 mA and a configurable updating interval between 2 and 35 ms. The resolution is determined by the hardware sensors, and the default updating interval is set to 35 ms. Although the updating interval can be adjusted at runtime, modifying it requires root privileges. Since AmpereBleed assumes an unprivileged attacker, we use the default updating interval throughout this paper.

IV. Evaluation

Experimental machine: We evaluate AmpereBleed on a Xilinx ZCU102 SoC, which features four Cortex-A53 cores with a base frequency of 1200 MHz and an FPGA fabric operating at 300 MHz. The FPGA fabric includes 274,080 lookup tables, 548,160 flip-flops, and 2,520 digital signal processing blocks. We use PetaLinux to customize Linux-based operating system for a specific SoC, which is a popular approach for Xilinx ARM-FPGA SoCs. All system configurations, including hwmon settings and dynamic voltage and frequency scaling (DVFS) policies, are kept by default.

A. Characterizing AmpereBleed

As outlined above, our AmpereBleed attack leverages the unprivileged hwmon interface to acquire current measurements that are related to the victim's behavior in theory. Here, we demonstrate that these current measurements are sufficient to distinguish various activities of the victim FPGA.

Varying computing workloads: Then, we deploy 160 k power virus instances described in Gand et al. [17] to cover major routing places of our ZCU102 board. They serve as a victim to stress FPGA logic, aligned with Zhao et al. [43]. We then divide them into 160 groups and each group has 1 k evenly-distributed instances. After deploying the bitstream onto the FPGA board, we can dynamically activate different numbers of these groups from the ARM processor to make the current vary by 161 levels.

Distinguishing different victim activities: Our first experiment is to characterize the hwmon's output under different FPGA workloads. To this end, we activate a varying number of power virus instances, making the victim activities vary from 161 levels. To construct a baseline, we follow Zhao et al. [43] to reproduce RO circuits and distribute them throughout the FPGA board to average dependence on spatial proximity to activated power virus instances. At each level, we collect 10 k samples about current, voltage, and power of FPGA logic from hwmon and the RO circuits, respectively, and compute the mean of these samples as the final value to derive the Pearson correlation coefficient which quantifies the strength and direction of the linear relationship.

Figure 2 illustrates the correlation between these measurements and the number of active power virus instances. FPGA current and power exhibit a strong linear relationship with the number of activated power virus instances, with a Pearson correlation coefficient of 0.999. Additionally, FPGA voltage achieves a Pearson correlation of 0.958, and RO achieves -0.996. This suggests that FPGA current and power readings, accessed via hwmon, correlate more strongly with victim activities under this experimental setup. Furthermore, we compute the linear function for each type of measurements. Due to power readings being updated at a maximum resolution of 25 mW, the difference between consecutive settings is limited to 1-2 least significant bits (LSBs). voltage measurements also show a limited linear correlation coefficient of 0.006 and support a fixed and coarse-grained resolution of 1.25 mV, leading to only slight LSB changes even under high workloads of 160 k activated power virus instances. In contrast, current measurements support a fine-grained resolution of 1 mA and vary approximately 40 LSBs per setting, enhancing their sensitivity to different victim activities and, consequently, improving attack performance as detailed in the following subsections. We clarify that the reason why current measurements do not start from 0 is due to the static workloads [26] caused by inactivated but deployed power virus instances.

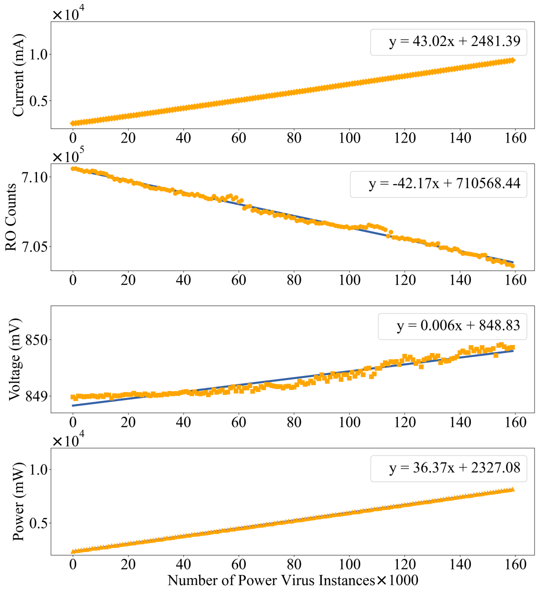
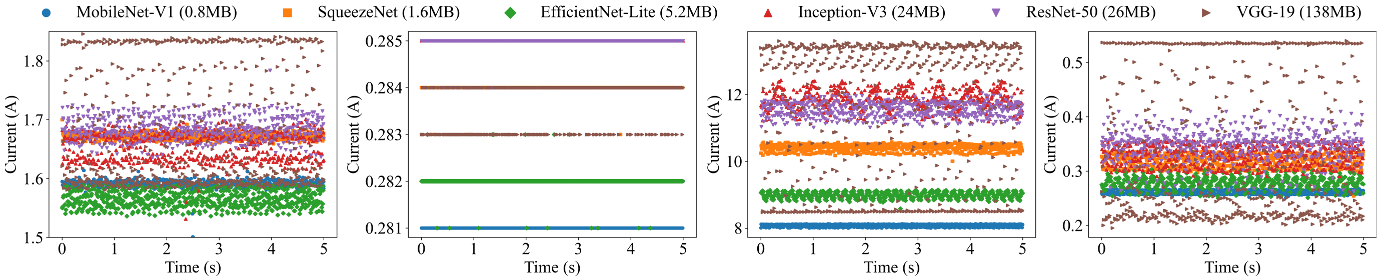


Fig. 2: The FPGA current, voltage, and power accessed via hwmon, along with RO counts, versus the number of activated power virus instances.

B. Fingerprinting DPU Accelerators

Xilinx DPU is a commercial framework for deploying pre-trained DNN models on FPGA boards. To protect its confidentiality and intellectual property, DPU encrypts its hardware description language (HDL) files at the source code level, following IEEE-1735-2014 V2 standard [6]. Due to this encryption, attackers cannot determine how the model inference is performed, such as the sequence of matrix computations and intensive memory read-write activities, increasing the difficulty of attack. Till now, the only one attack against DPU is to reverse-engineer the encrypted implementation



(a) Full-power CPU (b) Low-power CPU (c) FPGA (d) DRAM

Fig. 3: `current` patterns leaked from four sensors during different DNN model inferences.

details about DPU, which requires a high-resolution probe and physical proximity to sample electromagnetic signals [19].

In this subsection, we demonstrate a DNN model fingerprinting attack against the DPU framework, notably achieved without any specialized equipment. Our key insight is that distinct `current` variation patterns during inference can expose information about the underlying neural network operations. Knowledge of the DNN architecture poses a significant intellectual property risk [13], and can facilitate further AI attacks (e.g., adversarial example attack [27] and membership inference attack [40]).

Victim DPU accelerator: To deploy DPU framework on ZCU102 board, we utilize its official image provided by AMD-Xilinx [7]. This image incorporates Python version 3.9.9, Linux kernel version 5.15, and Vitis AI version 3.0. To minimize the impact of process scheduling interference, we schedule the task of triggering DPU inference onto CPU core 0, and the sampling tasks onto CPU core 3. We select a complete suite of image recognition models from Vitis AI Library as victim accelerators, including 39 architectures over 7 diverse architecture families. We conduct inference using the ImageNet ILSVRC Test set, resizing the input images to comply with the specifications of the victim accelerators. By default, the victim runs each model in series for 5 seconds.

Experimental setup: Aligned with previous work [27], [41], our model fingerprinting attack has two distinct phases: an *offline preparation* phase and an *online classification* phase. In the *offline preparation* phase, we collect traces of `current`, `voltage`, and `power` by separately sampling each interface to build a series of well-trained classifiers, each of which can classify a specific type of side channel traces to their corresponding model architectures. In the *online classification* phase, we issue queries to a black-box accelerator running on our FPGA board, triggering its model inference. Concurrently, we collect side-channel traces from `hwmon`. With the collected traces, we then use the corresponding classifier to fingerprint the architectures of encrypted DPU accelerators.

Since our fingerprinting attack is essentially a classification task with straightforward features, we use random forest (RForest) to conduct the evaluation, due to its suitability for handling high-dimensional data and identifying feature importance. We configure RForest with 100 trees and set the maximum depth to 32. The model uses Gini impurity as the splitting criterion, measuring class impurity within nodes to assess the quality of splits. We apply bootstrap sampling to build each decision tree, ensuring each tree is trained on a unique subset of data by selecting samples with replacement. For validation, we perform a 10-fold cross-validation where, in each iteration, 9 folds serve as training data and the remaining fold is used for testing.

TABLE III: Classification accuracy for encrypted accelerator fingerprinting. The baseline of random guess is 0.0256.

Duration Sensor		1 s	2 s	3 s	4 s	5 s (Full-length)
Current (Full-power CPU)	Top-1	0.823	0.830	0.832	0.834	0.837
	Top-5	0.981	0.980	0.982	0.981	0.982
Current (Low-power CPU)	Top-1	0.429	0.498	0.548	0.557	0.557
	Top-5	0.870	0.905	0.914	0.915	0.915
Current (DRAM)	Top-1	0.953	0.961	0.960	0.959	0.958
	Top-5	0.998	0.999	0.999	0.999	0.999
Current (FPGA)	Top-1	**0.980**	**0.985**	**0.997**	**0.997**	**0.997**
	Top-5	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Voltage (FPGA)	Top-1	0.080	0.099	0.116	0.118	0.116
	Top-5	0.317	0.323	0.327	0.326	0.330
Power (FPGA)	Top-1	0.929	0.973	0.989	0.991	0.989
	Top-5	0.997	0.997	0.997	0.996	0.996

Experimental results: Figure 3 illustrates examples of `current` traces collected via four `hwmon` interfaces while the DPU executes 6 DNN models separately, including MobileNet-V1, SqueezeNet, EfficientNet-Lite, Inception-V3, ResNet-50, and VGG-19, with their respective model sizes indicated. Each of the selected models produces a unique `current` patterns, influenced by the computing activities during the inference phase. The DPU's activity, involving the CPU, FPGA, and DRAM, is reflected in these `current` traces from these components, effectively captured by the `hwmon` interfaces and highlighting significant model fingerprinting vulnerabilities.

Table III presents the fingerprinting results of four sensitive `current` sensors, an FPGA `voltage` sensor, and an FPGA `power` sensor captured by `hwmon`, with each sensor's top-1 accuracy displayed in the first row and top-5 accuracy in the second row. We observe that all the `current` sensors achieve a high accuracy when the duration exceeds 3 s. Notably, FPGA `current` measurements reach a high accuracy of 99.7%, indicating a strong correlation between FPGA `current` variations and DPU activities. Given the greatly varied workloads of different DPU accelerators on the FPGA logic (as shown in

Figure 3(c)), indirect power measurements that are computed from current and voltage also achieve a high success rate of 98.9%. In contrast, the FPGA voltage measurements achieve a significantly lower accuracy of 11.6%.

C. Example Attack on an RSA

Beyond fingerprinting attacks, we present a concrete attack example that shows how AmpereBleed can infer the Hamming weight of secret keys from an RSA module implemented on an FPGA. Specifically, we assume an unprivileged attacker running on ARM cores targets a victim circuit performing RSA encryption. Till now, the only one attack against RSA-1024 circuit is Zhao et al. [43], where a malicious process samples the RO outputs with a high sampling frequency of 2 MHz. However, their victim RSA circuit operates at a low frequency of 20 MHz and RO circuits have been banned by commercial cloud providers (e.g., AWS [33]).

Victim RSA accelerator: We follow Zhao et al. [43] to implement an RSA-1024 circuit as the victim, and modify it to operate at 100 MHz. This circuit employs the Square-and-Multiply algorithm, which consists of two dedicated modular multiplication modules and a state machine. The state machine iterates through each bit of the 1024-bit input exponent, starting from the least-significant bit. One multiplication module is used for computing the square term and the other module is used for the multiplication operation. For every iteration, if the current least-significant bit of the iteratively-shifted exponent is ‘1’, both the two modules are activated to compute the square, inducing high switching activities. Otherwise, only a square operation will be activated. Both multipliers are synchronized to complete computation within the same cycle for each iteration of the loop. To protect the secret key, this implementation embeds the key within the encrypted bitstream. Once the circuit is deployed on an FPGA, the private key remains inaccessible, even to privileged users.

Experimental setup: To perform power side channel attack, our attack program continuously records the FPGA current measurements from the hwmon interface, using the sampling frequency of 1 kHz to collect 100,000 samples. During the current collecting, the RSA modular repeatedly encrypts a random plaintext. To demonstrate that FPGA current measurements can indeed leak information about private keys, we construct 17 distinct keys whose Hamming weights increase in intervals of 64, except for the first key that is chosen as 1 (since the RSA circuit does not support exponentiation with a value of 0). To make a comparison, we also use the FPGA power measurements to repeat the above procedure.

Experimental results: Figure 4 shows the statistical distribution of FPGA current and power measurements during RSA-1024 execution. Clearly, the attacker can use the FPGA current measurements to infer the Hamming weights of these private keys. As a comparison, the power measurements could only categorize the 17 keys into 5 groups. We note that knowledge of the Hamming weight can greatly reduce the search space of RSA's key brute force attack, and serves as a precursor for statistical analysis attacks [32].

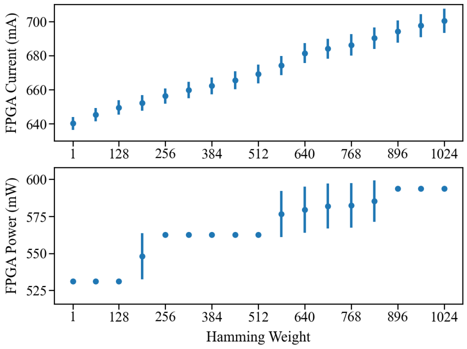


Fig. 4: The impact of the Hamming weight of RSA-1024 keys on FPGA current and power measurements.

V. Discussion

Mitigation: Since AmpereBleed exploits unprivileged access to INA226 sensors, restricting their access to privileged users can effectively mitigate the unprivileged attacks that are demonstrated in this paper. However, this restriction may badly affect benign programs that rely on these interfaces for performance monitoring, fault detection, and system management purposes. Besides, it requires driver or kernel updates for all affected devices, and cannot protect the legacy systems.

Future work: While this study primarily focuses on the Xilinx ARM-FPGA platforms, it raises the question of whether similar vulnerabilities exist in other FPGA SoCs (e.g., Intel's ARM-FPGA SoC products) as more FPGA boards are integrating a CPU as well as on-chip sensors within a single die. Besides, this paper only considers unprivileged attack scenarios, whether these INA226 sensors could be exploited to attack trusted execution environments (TEEs) implemented on FPGA [39] remains an open question. Last, as the on-chip current can also be measured by physical devices [26], we are also interested in its security implications as a physical side channel.

VI. Conclusion and Acknowledgments

In this work, we propose AmpereBleed, a new type of remote power side channel attack to infer FPGA activities. Specifically, AmpereBleed targets the ARM-FPGA SoC architectures and exploits the unprivileged hwmon subsystem to acquire current measurements. To demonstrate the viability of AmpereBleed, we first characterize the behavior of current measurements under varying power consumption conditions, experimental results of which show that fine-grained victim activities can be distinguished. We then successfully apply AmpereBleed to fingerprint DPU accelerators and infer the Hamming weights of different RSA private keys.

We thank the anonymous reviewers for their feedback. In addition, we thank Yuhui Zhang and Yusi Feng for their valuable comments that helped us to improve the work. This work was supported by and the Science and Technology Development Program of Two Districts in Xinjiang Province, China under Grant No. 2024LQ03004, the National Key R&D Program of China (No. 2022YFB2703301), and the Singapore Ministry of Education (Grant No. T1251RES2403).

References

[1] "FPGA Total Power Components Introduction." [Online]. Available: https://www.intel.com/content/www/us/en/support/programmable/support-resources/power/pow-overview.html

[2] "INA226 36V, 16-Bit, Ultra-Precise I2C Output Current, Voltage, and Power Monitor With Alert." [Online]. Available: https://www.ti.com/lit/ds/symlink/ina226.pdf

[3] "Pneumonia and COVID-19 Detection from X-Ray Images using Vitis-AI and Deployed by IoT GreenGrass." [Online]. Available: https://github.com/splineai-cloud/COVID-XS

[4] "Subaru Selects AMD MPSoCs for EyeSight System." [Online]. Available: https://iot-automotive.news/baidu-launches-global-first-mass-produced-autonomous-driving-computing-platform/

[5] "ZCU102 Evaluation Board User Guide (UG1182)." [Online]. Available: https://docs.amd.com/v/u/en-US/ug1182-zcu102-eval-bd

[6] "IEEE Recommended Practice for Encryption and Management of Electronic Design Intellectual Property (IP)," *IEEE Std 1735-2014 (Incorporates IEEE Std 1735-2014/Cor 1-2015)*, pp. 1–90, 2015.

[7] "Quick Start Guide for Zynq UltraScale+," 2024. [Online]. Available: https://xilinx.github.io/Vitis-AI/3.0/html/docs/quickstart/mpsoc.html

[8] "Xilinx Medical Clinical Solutions," 2024. [Online]. Available: https://www.xilinx.com/publications/prod_mktg/medical-clinical-brochure.pdf

[9] "Xilinx SoC FPGA Powers Baidu's Apollo Driverless Platform," 2024. [Online]. Available: https://www.electronicdesign.com/markets/automotive/article/21119589/xilinx-soc-fpga-powers-baidus-apollo-driverless-platform

[10] M. K. Ahmed and C. Bobda, "ISO-TENANT: Rethinking FPGA Power Distribution Network (PDN): A Hardware Based Solution for Remote Power Side Channel Attacks in FPGA," in *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2024.

[11] N. Bartzoudis, J. Rubio Fernández, D. López-Bueno, A. Román Villarroel, and A. Antonopoulos, "Agile FPGA Computing at the 5G Edge: Joint Management of Accelerated and Software Functions for Open Radio Access Technologies," *Electronics*, vol. 13, 2024.

[12] W. Diao, X. Liu, Z. Li, and K. Zhang, "No Pardon for the Interruption: New Inference Attacks on Android Through Interrupt Timing Analysis," in *IEEE Symposium on Security and Privacy*, 2016, pp. 414–432.

[13] Y. Gao, H. Qiu, Z. Zhang, B. Wang, H. Ma, A. Abuadbba, M. Xue, A. Fu, and S. Nepal, "DeepTheft: Stealing DNN Model Architectures through Power Side Channel," *IEEE Symposium on Security and Privacy*, 2023.

[14] J. Ge and F. Zhang, "SnapMem: Hardware/Software Cooperative Memory Resistant to Cache-Related Attacks On ARM-FPGA Embedded SoC," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024.

[15] I. Giechaskiel, K. B. Rasmussen, and J. Szefer, "C3APSULe: Cross-FPGA covert-channel attacks through power supply unit leakage," in *IEEE Symposium on Security and Privacy*, 2020.

[16] O. Glamočanin, L. Coulon, F. Regazzoni, and M. Stojilović, "Are cloud FPGAs really vulnerable to power analysis attacks?" in *Design, Automation & Test in Europe Conference & Exhibition*, 2020.

[17] D. R. Gnad, F. Oboril, and M. B. Tahoori, "Voltage drop-based fault attacks on FPGAs using valid bitstreams," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2017, pp. 1–7.

[18] M. Gobulukoglu, C. Drewes, W. Hunter, R. Kastner, and D. Richmond, "Classifying Computations on Multi-Tenant FPGAs," in *Design Automation Conference*, 2021, pp. 1261–1266.

[19] C. Gongye, Y. Luo, X. Xu, and Y. Fei, "Side-Channel-Assisted Reverse-Engineering of Encrypted DNN Hardware Accelerator IP and Attack Surface Exploration," in *IEEE Symposium on Security and Privacy*.

[20] A. He, D. Key, M. Bulling, A. Chang, S. Shapiro, and E. Lee, "HLSTransform: Energy-Efficient Llama 2 Inference on FPGAs Via High Level Synthesis," 2024. [Online]. Available: https://arxiv.org/abs/2405.00738

[21] H. Homulle, S. Visser, B. Patra, and E. Charbon, "Design techniques for a stable operation of cryogenic field-programmable gate arrays," *Review of Scientific Instruments*, vol. 89, no. 1, 2018.

[22] D. Jayasinghe, B. Udugama, and S. Parameswaran, "1LUTSensor: Detecting FPGA Voltage Fluctuations using LookUp Tables," *Cryptographic Hardware and Embedded Systems*, 2024.

[23] T. Kim and Y. Shin, "ThermalBleed: A Practical Thermal Side-Channel Attack," *IEEE Access*, 2022.

[24] R. Li, W. Diao, Z. Li, J. Du, and S. Guo, "Android Custom Permissions Demystified: From Privilege Escalation to Design Shortcomings," in *IEEE Symposium on Security and Privacy*, 2021, pp. 70–86.

[25] J. M. Mbongue, D. T. Kwadjo, A. Shuping, and C. Bobda, "Deploying Multi-tenant FPGAs within Linux-based Cloud Infrastructure," *ACM Trans. Reconfigurable Technol. Syst.*, 2021.

[26] A. Moradi, "Side-Channel Leakage through Static Power – Should We Care about in Practice?" in *CHES*, 2014.

[27] K. Patwari, S. M. Hafiz, H. Wang, H. Homayoun, Z. Shafiq, and C.-N. Chuah, "DNN Model Architecture Fingerprinting Attack on CPU-GPU Edge Devices," in *European Symposium on Security and Privacy (EuroS&P)*, 2022, pp. 337–355.

[28] C. Plappert and A. Fuchs, "Secure and Lightweight ECU Attestations for Resilient Over-the-Air Updates in Connected Vehicles," in *Annual Computer Security Applications Conference*, 2023, p. 283–297.

[29] Plappert, Christian and Fuchs, Andreas, "Secure and Lightweight Over-the-Air Software Update Distribution for Connected Vehicles," in *Annual Computer Security Applications Conference*, 2023, p. 268–282.

[30] G. D. RE, J. Krautter, and M. B. Tahoori, "Leaky noise: New side-channel attack vectors in mixed-signal IoT devices," in *Cryptographic Hardware and Embedded Systems*, 2019, pp. 305–339.

[31] V. J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J. S. Gardner, I. Hubara, S. Idgunji, T. B. Jablin, J. Jiao, T. S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A. T. R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, and Y. Zhou, "MLPerf Inference Benchmark," in *International Symposium on Computer Architecture*, 2020, pp. 446–459.

[32] S. Sarkar and S. Maitra, "More on correcting errors in RSA private keys: Breaking CRT-RSA with low weight decryption exponents," *Cryptographic Hardware and Embedded Systems*, 2012.

[33] A. W. Services. (2024) AWS EC2 FPGA HDK+SDK errata. [Online]. Available: https://github.com/aws/aws-fpga/blob/master/ERRATA.md

[34] D. Spielmann, O. Glamočanin, and M. Stojilović, "RDS: FPGA Routing Delay Sensors for Effective Remote Power Analysis Attacks," *Cryptographic Hardware and Embedded Systems*, pp. 543–567, 2023.

[35] B.-H. Su, J.-S. Tang, H.-J. Lee, and C.-B. Tzeng, "Noise Analysis and Improvement of Power Supply Network Based on Power Integrity," in *International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, 2023.

[36] X. Tan, Z. Ma, S. Pinto, L. Guan, N. Zhang, J. Xu, Z. Lin, H. Hu, and Z. Zhao, "SoK: Where's the "up"?! A Comprehensive (bottom-up) Study on the Security of Arm Cortex-M Systems," 2024.

[37] B. Udugama, D. Jayasinghe, H. Saadat, A. Ignjatovic, and S. Parameswaran, "A power to pulse width modulation sensor for remote power analysis attacks," *Cryptographic Hardware and Embedded Systems*, 2022.

[38] Udugama, Brian and Jayasinghe, Darshana and Saadat, Hassaan and Ignjatovic, Aleksandar and Parameswaran, Sri, "VITI: A tiny self-calibrating sensor for power-variation measurement in FPGAs," *Cryptographic Hardware and Embedded Systems*, pp. 657–678, 2022.

[39] K. Xia, Y. Luo, X. Xu, and S. Wei, "SGX-FPGA: Trusted Execution Environment for CPU-FPGA Heterogeneous Architecture," in *Design Automation Conference*, 2021, pp. 301–306.

[40] S. Zhai, H. Chen, Y. Dong, J. Li, Q. Shen, Y. Gao, H. Su, and Y. Liu, "Membership Inference on Text-to-Image Diffusion Models via Conditional Likelihood Discrepancy," in *Annual Conference on Neural Information Processing Systems*, 2024.

[41] X. Zhang, Z. Zhang, Q. Shen, W. Wang, Y. Gao, Z. Yang, and Z. Wu, "ThermalScope: A Practical Interrupt Side Channel Attack Based On Thermal Event Interrupts," in *Design Automation Conference*, 2024.

[42] Y. Zhang, R. Yasaei, H. Chen, Z. Li, and M. A. Al Faruque, "Stealing neural network structure through remote FPGA side-channel analysis," *IEEE Transactions on Information Forensics and Security*, 2021.

[43] M. Zhao and G. E. Suh, "FPGA-Based Remote Power Side-Channel Attacks," in *IEEE Symposium on Security and Privacy*, 2018.

[44] W. Zhou, Y. Jia, Y. Yao, L. Zhu, L. Guan, Y. Mao, P. Liu, and Y. Zhang, "Discovering and understanding the security hazards in the interactions between IoT devices, mobile apps, and clouds on smart home platforms," in *USENIX Security Symposium*, 2019.