Guarding the Gatekeepers: Ensuring the Security of Computation Hardware in Cloud Infrastructure

Kejun Chen, Xiaolong Guo
ECE Department
Kansas State University
Manhattan, KS, US
Email: kejun@ksu.edu, guoxiaolong@ksu.edu

Xuan Zhang
ESE Department
Northeastern University
Boston, MA, US
Email: xuan.zhang@northeastern.edu

Xianglong Feng
CSE Department
Miami University
Oxford, Ohio, US
Email: fengx17@miamioh.edu

***Abstract*—Physical computation devices, including CPUs, FPGAs, and GPUs, are integral to cloud computing but face unique security challenges. While cloud infrastructures are pivotal for service delivery, they are susceptible to threats. This paper introduces a novel hardware security framework to bolster cloud infrastructure resilience. Utilizing side-channel measurements from the power distribution network (PDN), the framework detects anomalies in computational devices. Leveraging Ring Oscillators and Time-to-Digital Converters, we design PDN sensors, further enhancing security with a co-processor for real-time checks based on Neural Network analysis.**

***Keywords*—Power Distribution Network; Co-Processor; Runtime Security Verification; Hardware Security**

I. Introduction

Physical computation devices, such as CPUs, FPGAs, and GPUs, are pivotal in cloud computing. They execute applications, store data, and provide essential computational resources. While cloud computing offers a plethora of benefits, it also presents distinct security challenges, particularly related to these physical computational devices. The physical infrastructure underpinning cloud computing is indispensable for service delivery, yet it remains vulnerable to threats. Unauthorized physical access to these devices can lead to dire consequences, ranging from data breaches to service interruptions. Threats can manifest through direct tampering, exploitation of hardware vulnerabilities, or interception of data transmissions, emphasizing the need for stringent physical security measures. It is imperative for cloud service providers to prioritize the protection of their hardware, from its manufacturing phase to its operational setting, to guarantee the data's integrity and confidentiality and the services they offer.

Current solutions predominantly rely on software-based approaches, where anomalies are detected using anti-virus software. However, [1] demonstrated the limitations of these solutions in detecting novel malware. Given that attackers can effortlessly alter software, hardware-based solutions have gained traction, as tampering with hardware presents significant challenges [2]. To mitigate these concerns, we introduce a hardware-based security solution designed to bolster the resilience of the cloud computing infrastructure. Our proposed framework ensures the robustness and integrity of computational devices, including CPUs, FPGAs, and GPUs, within the cloud environment. Specifically, the framework leverages side-channel measurements tied to the power distribution network (PDN) activity of the targeted computational devices. As depicted in Fig. 1, the framework addresses two scenarios: 1) where the targeted device and the PDN sensor share the same power source, and 2) where the targeted device and the PDN sensor share a PCIE communication channel.

To summarize, our contributions include:

- We propose a non-destructive approach to protect cloud computing infrastructure using the voltage fluctuation information.
- A specialized neural network model for anomaly detection is developed, and a corresponding dataset is constructed.
- We deliver a hardware platform designed for real-world testing and validation of our proposed framework.

II. Threat Model

The proposed framework is designed to allow cloud service providers to detect anomalous software or firmware behaviors at the runtime using hardware measurements within their multi-tenant cloud platforms. We posit that these anomalies are introduced by attackers with access to the computational devices. Furthermore, our framework considers the possibility that malicious firmware could be covertly embedded by a rogue or offshore vendor during the supply chain process. Potential consequences of these anomalies encompass data breaches, integrity violations, denial of service, account hijacking, man-in-the-middle attacks, ransomware attacks, and privilege escalation.

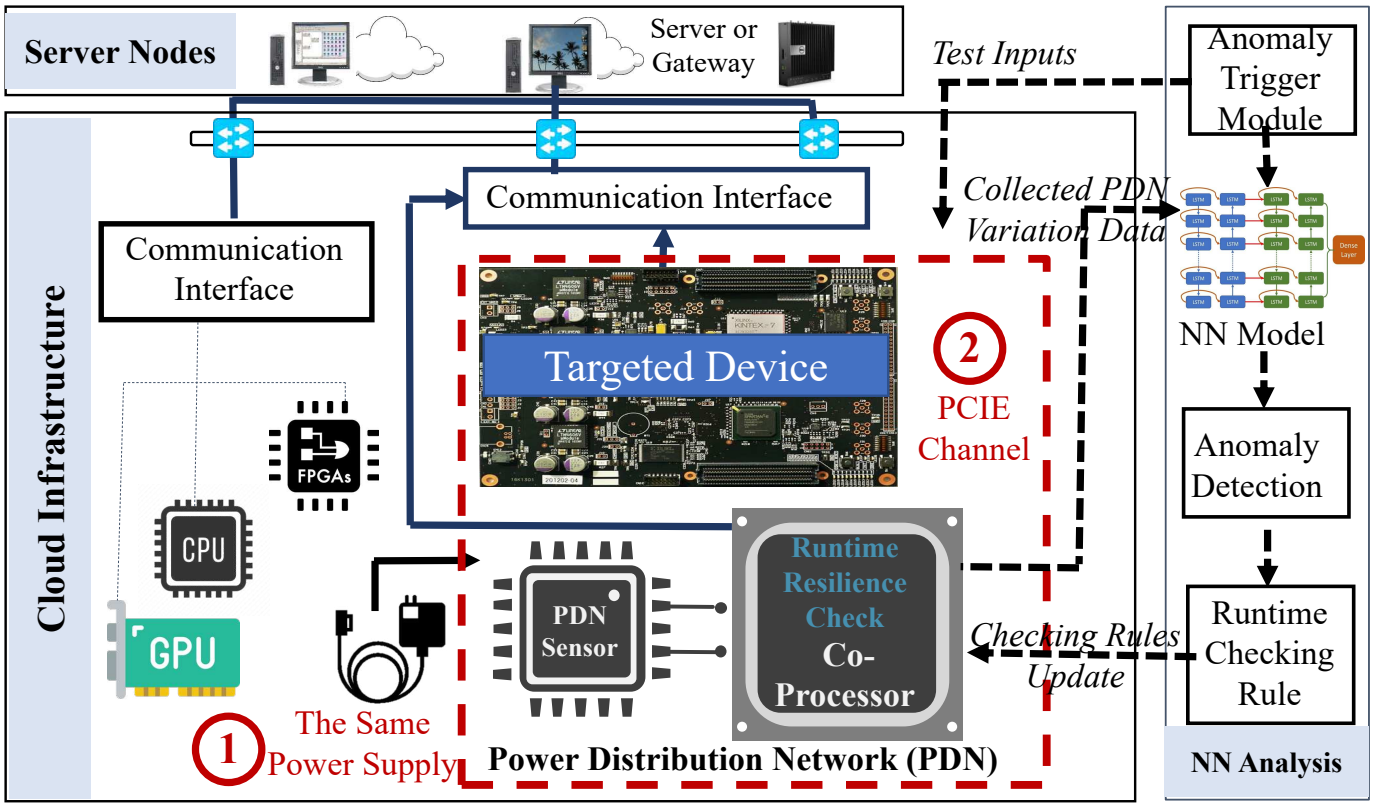

Fig. 1: The overview of the proposed framework.

III. Preliminary Work

The PDN, at the hardware level, is a sophisticated passive network responsible for delivering power to each computing unit within the silicon substrate. Simultaneously, all PDN components are susceptible to process variations [3]. While attackers often use Ring Oscillators (RO)- or Time-to-Digital Converters (TDC)-based sensors to monitor power supply fluctuations for side-channel analysis attacks [4], our proposed framework aims to utilize the sensitivity of these PDN sensors to detect attacks or anomalies.

In our proposed framework, we hypothesize that device anomalies will manifest as fluctuations in the PDN. Conversely, specific patterns in PDN fluctuations can be indicative of sensor anomalies. This hypothesis has been corroborated in our preliminary studies [5], [6]. To achieve this, we designed a PDN sensor to capture power trace vibrations. We utilize Neural Networks (NN) to associate these PDN vibrations with sensor anomalies. A dedicated co-processor, connected to the PDN sensor, is conceived to facilitate real-time anomaly detection. Drawing from our initial research [7], [8], the proposed co-processor has been realized. The comprehensive solution is envisioned as a plug-in hardware module, facilitating seamless integration with the existing structure of cloud devices.

IV. Methodology

A. Framework Architecture

We use Figure 1 to elucidate the architecture of our proposed framework.

We develop two types of PDN sensors to measure the voltage fluctuations caused by the operations of the targeted computation devices. Specifically, we leverage the characteristics of Ring Oscillators (ROs) and Time-to-Digital Converters (TDCs) for the design of the PDN sensors. Figure 2(a) and (b) illustrate the schematics of the RO- and TDC-based sensor designs, respectively. The frequency of oscillation of an RO is directly proportional to the supply voltage; thus, we have designed a counter to periodically record the frequency of the RO. In the case of the TDC, the timing of switches can be correlated with the stability of a power supply. Therefore, we have configured a ones-counter to record the number of "1"s, which serves as an indicator of power fluctuations.

The co-processor then gathers the measured data and transfers it to a host machine for preprocessing. To identify anomalies in the sensor data, we use a Neural Network analysis method. To accumulate sufficient data for training the NN model, an anomaly trigger module is developed. We simulate various attacks while concurrently collecting data via the PDN sensor and co-processor. After that, both computational logic and memory are incorporated into the co-processor, enabling

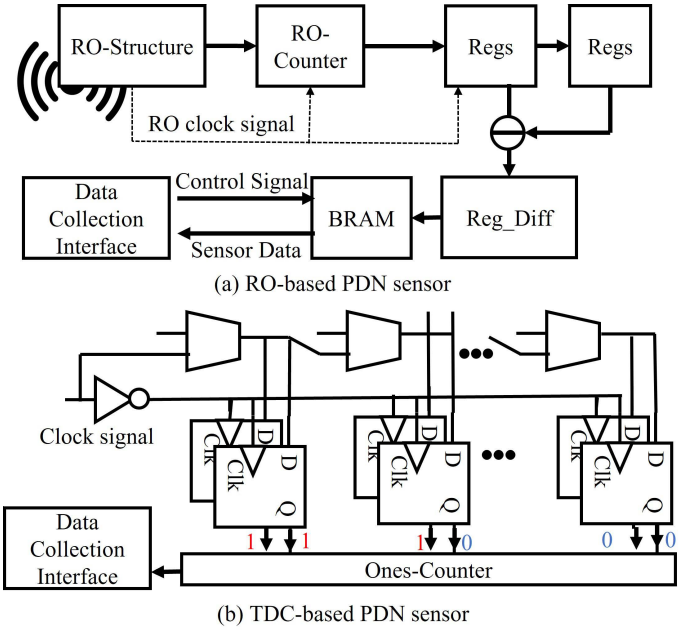

Fig. 2: PDN sensors: (a) Ring-oscillator (RO) based structure; (b) Time-to-Digital (TDC) based structure.

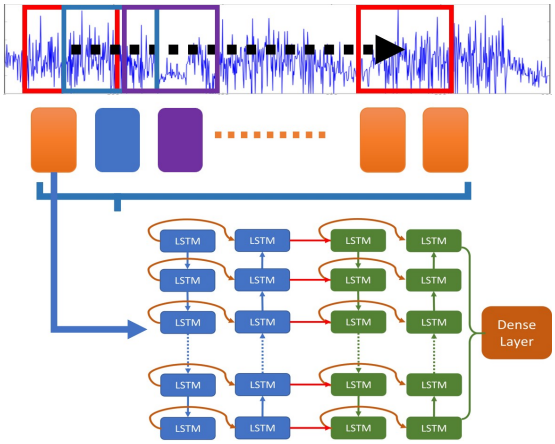

Fig. 3: Sensor data prepossessing & sequential neural network framework.

it to conduct real-time security checks. Rules for these runtime checks are formulated based on the anomaly detection results obtained from the trained NN model. The PDN sensor will continue to monitor the power traces from the targeted devices, allowing the co-processor to process and compare this information against the established checking rules.

B. Neural Network for Anomaly Detection

Among various neural network models, we chose the Long Short-Term Memory (LSTM) module as our primary tool for extracting sequential features from the time-series data collected by the sensors. Initially, this data is segmented using a sliding window technique. These segments are then labeled and compiled to form the dataset, which is subsequently divided into training, validation, and test sets. Within each data segment, two bidirectional LSTM models process the information, extracting sequential features in both forward and backward directions. Each LSTM maintains a consistent output vector size. The output from the final LSTM module is then fed into a dense layer to produce the classification result.

V. Experimental Results

To assess the efficacy of our proposed security framework, this paper's experiments are divided into two case studies. The first case study presents system-level performance data, demonstrating enhanced security for scenarios 1 and 2 as depicted in Figure 1. The second case study supports the hypothesis that variances in software behavior, such as device anomalies, contribute to fluctuations in power voltage, with the PDN sensor exhibiting sufficient resolution to detect these variations.

A. Case Study One: System-Level Behaviors Check

The architecture of our current experimental environment is depicted in Figure 4. This platform will primarily consist of three main components: a Xilinx Virtex-7 FPGA VC707 board, a targeted device, and a high-performance host workstation. Both the PDN sensor and the co-processor are implemented on the FPGA core. In this case study, the RO-based PDN sensor records measurement data and transmits it to the host workstation for analysis. As illustrated in Figure 2 (a), the frequency of the *RO-Counter* is governed by the *RO-Structure*. The pipeline registers operate at a sampling frequency of 62.5 MHz. The collected data are stored in *BRAM* before being packaged and transferred to the host workstation via the Peripheral Component Interconnect Express (PCIE) interface.

We have implemented three test settings to collect the ROs sensor data and further evaluate the performance of our proposed deep learning based sensing approach. The three test settings are: *Setting A:* we run five benchmarks on the same chip where the ROs sensor is implemented; *Setting B:* we run two benchmarks on the PC and the developing board is mounted inside the PC. *Setting C:* we run two benchmarks on the PC, and the developing board is mounted outside the PC. Therefore, we collect the ROs sensor data and form a dataset with 9 testing cases under the three test settings. Based on different combinations of the three settings, we assemble four testing cases. For each testing case, the dataset will be divided into training, validation, and test datasets with the portion of $60\%, 20\%$, and 20%. Then, we apply our neural network to this dataset to check the inference accuracy, the result of which is shown in the Table I.

As we can see from the table I, the inference accuracy is around 90% for most of the test cases, which indicates

that our proposed solution shows great potential for security monitoring. We could also notice that, when including more test cases, the accuracy will decrease, which could be further solved by improving the model and training with more datasets.

TABLE I: Inference accuracy for different test combination

Test Case	Setting B	Setting B + Setting C	Setting A	Setting A + Setting B + Setting C
Accuracy (%)	94	94	92	88

B. Case Study Two: Code-Level Behaviors Check

This case study presents the results from code-level experiments, elucidating how the proposed framework detects system-level anomalies. Our experiments, conducted on an Artix-7 FPGA development board (Arty-100), aim to verify the universality of our methods. The experimental environment for this case study is nearly identical to that of the system-level experiments. The RO-based PDN sensor is again employed for power data measurement.

The primary distinction between the VC707 and Arty-100t platforms lies in their data collection interfaces. For the Arty-100t, we utilize the Logic Pro 16 logic analyzer [9] to directly capture data from the I/O pins. The results indicate that the voltage fluctuations inherent in this data collection method do not affect the detection accuracy. Additionally, we employ the MicroBlaze [10] IP microprocessor as our target device, executing standard embedded benchmarks (BEEBs [11]) on it to emulate diverse device behaviors. This setup allows the attachment of third-party IPs and peripherals to the system bus.

In our experiments, all 18 benchmarks were input into the neural network model for classification. The ability to differentiate these benchmarks demonstrates that our framework can identify specific software codes based on power fluctuation analysis during hardware execution. Table II details the accuracy for each benchmark. The average classification accuracy across 18 benchmarks exceeds 75%, with individual benchmark accuracies surpassing 65%. Notably, the accuracy diminishes as more benchmarks are included. However, the 18-classification accuracy is deemed satisfactory, and there's potential for improvement through the deployment of additional sensors and the collection of more extensive data per benchmark.

TABLE II: Inference accuracy for different Beebs benchmarks

Beebs Benchmark	Accuracy
strstr	71%
libcompress	67%
cnt	84%
sglib-arraybinsearch	70%
nettle-md5	68%
newlib-sqrt	91%
qsort	66%
duff	99%
dijkstra	70%
newlib-exp	69%
sqrt	65%
frac	68%
huffbench	69%
newlib-mod	99%
crc	80%
fir	75%
nettle-des	69%
newlib-log	80%

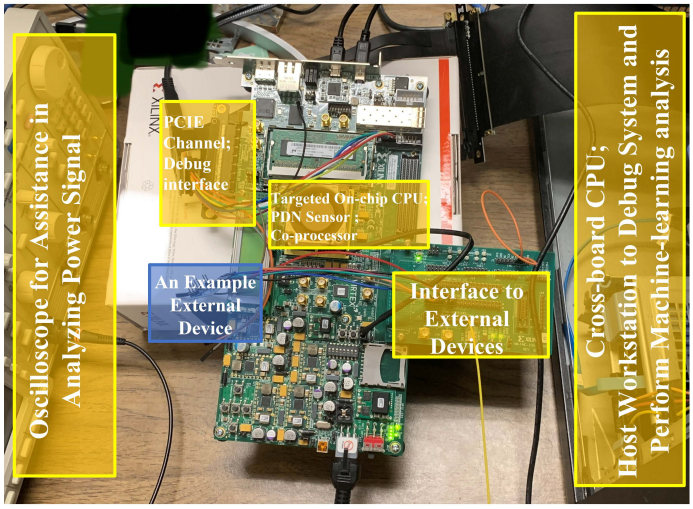

Fig. 4: The experimental test platform.

VI. Conclusion

In conclusion, our study introduces a novel hardware-based security framework for cloud computing, focusing on PDN activity in devices like CPUs and GPUs. This approach marks a significant shift from traditional software solutions, offering enhanced protection against hardware-level threats and setting a foundation for more secure cloud infrastructures.

Acknowledgments

Portions of this work were supported by the National Science Foundation (CCF-2019310, First Award Program of ARISE in EPSCoR 2148878).

References

[1] S. Jana and V. Shmatikov, "Abusing file processing in malware detectors for fun and profit," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 80–94.

[2] K. Basu, P. Krishnamurthy, F. Khorrami, and R. Karri, "A theoretical study of hardware performance counters-based malware detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 512–525, 2019.

[3] J. Krautter, D. Gnad, and M. Tahoori, "Cpamap: On the complexity of secure fpga virtualization, multi-tenancy, and physical design," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 121–146, 2020.

[4] Y. Luo, C. Gongye, Y. Fei, and X. Xu, "Deepstrike: Remotely-guided fault injection attacks on dnn accelerator in cloud-fpga," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 295–300.

[5] H. Zhu, H. Shan, D. Sullivan, X. Guo, Y. Jin, and X. Zhang, "Pdnpulse: sensing pcb anomaly with the intrinsic power delivery network," *IEEE Transactions on Information Forensics and Security*, 2023.

[6] H. Zhu, X. Guo, Y. Jin, and X. Zhang, "Powerscout: Security-oriented power delivery network modeling for side-channel vulnerability analysis," *IEEE Transactions on Emerging Topics in Computing*, 2023.

[7] K. Chen, O. Arias, X. Guo, Q. Deng, and Y. Jin, "Ip-tag: Tag-based runtime 3pip hardware trojan detection in soc platforms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 1, pp. 68–81, 2022.

[8] K. Chen, O. Arias, Q. Deng, D. Oliveira, X. Guo, and Y. Jin, "Finedift: Fine-grained dynamic information flow tracking for data-flow integrity using coprocessor," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 559–573, 2022.

[9] Logic Pro 16. https://www.saleae.com/.

[10] MicroBlaze. https://www.xilinx.com/products/design-tools/microblaze.html.

[11] J. Pallister, S. Hollis, and J. Bennett, "Beebs: Open benchmarks for energy measurements on embedded platforms," *arXiv preprint arXiv:1308.5174*, 2013.