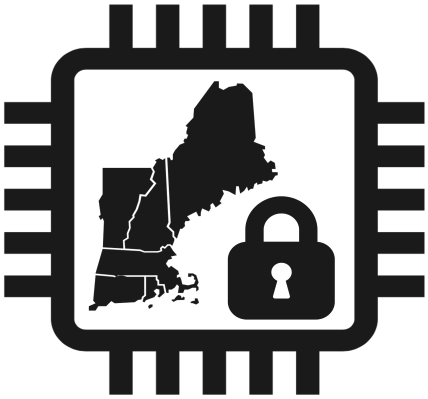


New England Hardware Security Day

5th Workshop, NEHWS Day
Massachusetts Institute of Technology, Cambridge, MA

Organization

General Chair

Fatemeh Ganji	Worcester Polytechnic Institute

Program Committee Chairs

Xiaolin Xu	Northeastern University
Dean Sullivan	University of New Hampshire

Program Committee

Orlando Arias	University of Massachusetts, Lowell
Prabuddha Chakraborty	University of Maine
Daniel Holcomb	University of Massachusetts, Amherst
Sandhya Koteshwara	IBM
Yukui Luo	University of Massachusetts, Dartmouth
Satwik Patnaik	University of Delaware
Patrick Schaumont	Worcester Polytechnic Institute
Berk Sunar	Worcester Polytechnic Institute
Shahin Tajik	Worcester Polytechnic Institute
David Kaeli	Northeastern University
Xiaolong Guo	Kansas State University
Yan Long	University of Virginia
Zirui Zhao	NVIDIA

Panel Chair

Daniel Holcomb	University of Massachusetts, Amherst
Seetal Potluri	University at Albany

Poster Chair

Yingjie Lao	Tufts University

Industry Chairs

Shahin Tajik	Worcester Polytechnic Institute
Jakub Szefer	Northwestern University

Outreach Chair

Jakub Szefer	Northwestern University

Publicity Chairs

Shahin Tajik	Worcester Polytechnic Institute
Dean Sullivan	University of New Hampshire

Web Chair

Kemal Derya	Worcester Polytechnic Institute

Preface

The 5th New England Hardware Security (NEHWS) Day was held at Massachusetts Institute of Technology, Cambridge, MA on April 18, 2025. The workshop is organized annually by the hardware security research community at these universities: Massachusetts Institute of Technology, Northeastern University, University of Massachusetts Amherst, University of New Hampshire, Worcester Polytechnic Institute, Tufts, and Northwestern.

NEHWS Day aims to initiate a discussion between industry and academia by outlining research priorities and opportunities in *hardware* security and verification according to the strategic visions of semiconductor companies and government agencies. NEHWS DAY 2025 was a one-day regional workshop with a poster session, panel, long and short talks, as well as three keynotes.

NEHWS Day 2025 received 55 submissions, including abstracts of 21 long and 7 short talks, and 27 posters. The program committee chose proposed talks based on fit for NEHWS, whether the talk was likely to be good, and whether it would stimulate discussion and interest among the audience of hardware security practitioners, academics, and students. With this regard, the NEHWS program committee selected 6 long and 5 short talks to be presented from the podium with slides, followed by a brief Q&A. Authors of talk abstracts not accepted for the presentation were invited to present their work in poster format. In total, 20 posters were displayed at NEHWS 2025.

These proceedings include the program, abstract of 11 accepted talks, and the list of accepted posters.

We are grateful to General Chair Fatemeh Ganji for her help and efforts. We are thankful to the workshop's sponsors: National Science Foundation (NSF), MIT's School of Engineering, MIT's Schwarzman College of Engineering, MIT's Department of Electrical Engineering and Computer Science, BAE Systems, Northeast Microelectronics Coalition (NEMC) Hub, DRAPER.

We would like to also extend our appreciation to Anantha P. Chandrakasan (dean of MIT's School of Engineering), Daniel Huttenlocher(dean of the MIT Schwarzman College of Computing), and Asu Ozdaglar (EECS Department Head at MIT) for their support.

Finally, we thank all the authors who submitted papers to NEHWS 2025, and program committee members who made this event a truly intellectually stimulating one.

March 2025
Xiaolin Xu
Dean Sullivan

NEHWS Day 2025

April 18th, 2025

Massachusetts Institute of Technology

Samberg Conference Center, Chang Building

9:00 - 9:15	Welcome Remarks		Session chair: Mengjia Yan
9:15 - 9:30	Opening remarks by Dean of the MIT Schwarzman College of Computing, Daniel Huttenlocher		Session chair: Mengjia Yan
	News from the NEHWS Community		
9:30 - 10:30	Invited Keynote by Todd Austin (University of Michigan)		
10:30 - 10:45	Break		
10:45 - 12:00	Submitted Talks-- Session I		Session chair: Xiaolin Xu
	Long Talk	FaultDetective: Explainable to a Fault, from the Design Layout to the Software Zhenyuan Liu, Dillibabu Shanmugam and Patrick Schaumont	
	Short Talk	BackMon: IC Backside Tamper Detection using On-Chip Impedance Monitoring Tahoura Mosavirik and Shahin Tajik	
	Long Talk	Who's in Control? Security Risks in Commercial ADAS Yingjie Cao, George Crane, Haoqi Shan and Dean Sullivan	
	Short Talk	Energy-Accuracy-Security Trade-offs in Resistive In-Memory Computing Architectures Saion Roy and Naresh Shanbhag	
	Long Talk	VGF: :Value-Guided Fuzzing -- Fuzzing Hardware as Hardware -- Ruochen Dai, Michael Lee, Patrick Hoey, Weimin Fu, Yier Jin, Xiaolong Guo, Shuo Wang, Dean Sullivan, Tuba Yavuz and Orlando Arias	
	Short Talk	USBSnoop - Revealing Device Activities via USB Congestions Davis Ranney, Yufei Wang, A. Adam Ding and Yunsi Fei	
12:00 - 1:30	Lunch and Poster Session		
1:30 - 2:30	Panel "Secure-by-Construction for Hardware Design: Mission Impossible or Walk in the Park?" - Shivam Basin (Nanyang Technical University) - Srini Devadas (Massachusetts Institute of Technology) - Sharad Malik (Princeton University) - Christof Paar (Max Planck Institute) - Dominic Rizzo (zeroRISC)		Session chair: Daniel Holcomb
2:30 - 3:30	Invited Keynote by Edward Suh (Nvidia)		Session chair: Dean Sullivan
3:30 - 3:40	Break		
3:40 - 4:45	Submitted Talks-- Session II		Session chair: Dean Sullivan
	Long Talk	Exploiting Exclusive System-Level Cache in Apple M-Series SoCs for Enhanced Cache Occupancy Attacks Tianhong Xu, Aidong Ding and Yunsi Fei	
	Short Talk	Uncertainty Estimation in Neural Network-enabled Side-channel Analysis and Links to Interpretability Seyedmohammad Nouraniboosjin and Fatemeh Ganji	
	Long Talk	Oreo: Protecting ASLR Against Microarchitectural Attacks Shixin Song, Joseph Zhang and Mengjia Yan	
	Short Talk	Cross-Layer EM Fault Injection Assessment Framework Hanqiu Wang, Ruochen Dai, Tuba Yavuz, Xiaolong Guo, Orlando Arias, Dean Sullivan, Siqi Dai, Honggang Yu, Michael Lee, Domenic Forte and Shuo Wang	
	Long Talk	IvLeague: Side Channel-resistant Secure Architectures Using Isolated Domains of Dynamic Integrity Trees Md Hafizul Islam Chowdhuryy and Fan Yao	
4:45 - 5:45	Invited Keynote by Makoto Nagata (Kobe University)		Session chair: Patrick Schaumont
5:45 - 6:00	Best Poster Award and Closing Remarks		Session chairs: Yingjie Lao
6:00 - 7:00	Networking reception		

NSF

BAE SYSTEMS

DRAPER

NEMC HUB
Northeast Microelectronics Coalition

zeroRISC

FaultDetective: Explainable to a Fault, from the Design Layout to the Software

Zhenyuan Liu, Dillibabu Shanmugam and Patrick Schaumont
Worcester Polytechnic Institute, Worcester, USA, {zliu12, dshanmugam, pschaumont}@wpi.edu

Motivation. Fault injection in secure embedded systems leads to information leakage and privilege escalation, and countless fault attacks have been demonstrated both in simulation and in practice. However, there is a significant gap between simulated fault attacks and physical fault attacks. Simulations use idealized fault models such as single-bit flips with uniform distribution. These ideal fault models may not hold in practice. On the other hand, practical experiments lack the white-box visibility necessary to determine the true nature of the fault, leading to probabilistic vulnerability assessments and unexplained results. In embedded software, this problem is further exacerbated by the layered abstractions between the hardware (where the fault originates) and the application software (where the fault effect is observed).

Main Contribution. We present FAULTDETECTIVE [1], a method to investigate the root-cause of fault injection from software fault detection, revealing it as the *endpoint* of a fault propagation chain originating in hardware. To analyze fault effects at the hardware level, a scan chain is employed with white-box simulation to propagate and observe hardware faults in embedded software. A hash-tree of the scan bits is used to visualize fault propagation across abstraction levels and supports the construction of the Dynamic State Transition Graphs (DSTGs). The scan chain state transitions are analyzed for both sane and weird machines. A digest is computed over scan bits per word and module, forming a hierarchical tree structure up to a processor-level digest. An architecture-level digest incorporates architecture-visible scan bits, while a micro-architecture digest includes all scan bits. The processor digest provides a compact, bit-sensitive representation of the processor state. This hierarchical structure enables comparisons among multiple scan chains to pinpoint the origin or outcome of faults by isolating differences in modules, words, or bits.

Realization in ASIC. An ASIC was developed to test the proposed method by implementing a redundant microcontroller design, where cores execute in lock-step, each with its own memory, processor, and peripherals (Figure 1, left). An on-chip network (blue connections) links the cores, enabling them to exchange checksum values computed in software for distributed fault detection. Any core can trigger a system-wide exception, ensuring fault detection across the entire system. Once a fault is injected, scan states are periodically scanned out to construct the DSTGs (Figure 1, right). These scan states reveal how faults propagate from hardware to software, bridging the gap between simulated and empirical fault models. The fault root-cause is identified as the earliest observable fault in the scan chain, while the fault trajectory maps the path from this initial hardware fault to its eventual detection in software. We observe the fault effects for several different stressors, including clock glitching and thermal laser stimulation, and explain the root-cause in several examples.

Acknowledgment. This research was supported in part by NSF Award 2219810.

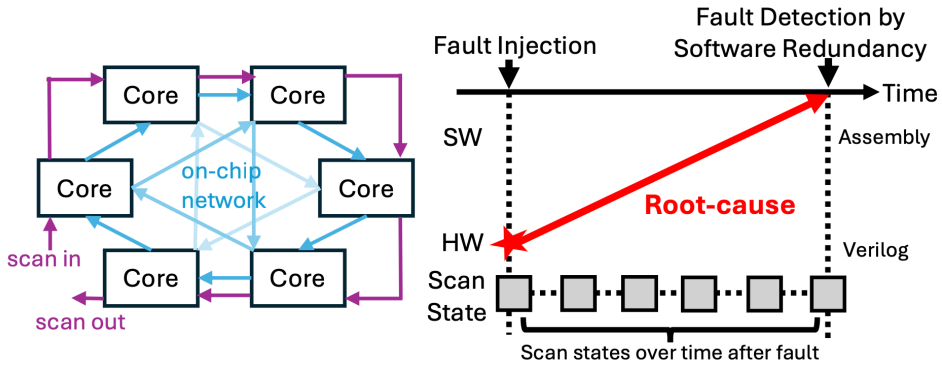


Figure 1: (Left) Scan chain in a multi-core system. (Right) Root-cause analysis for faults.

References

[1] Liu, Z., Shanmugam, D. and Schaumont, P. (2024) "FaultDetective: Explainable to a Fault, from the Design Layout to the Software", IACR Transactions on Cryptographic Hardware and Embedded Systems, 2024(4), pp. 610–632. doi:10.46586/tches.v2024.i4.610-632

BackMon: IC Backside Tamper Detection using On-Chip Impedance Monitoring

Tahoura Mosavirik[1] and Shahin Tajik[1]

[1] Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (WPI)

tmosavirik@wpi.edu, stajik@wpi.edu

The expansion of flip-chip technologies and a lack of backside protection make the integrated circuit (IC) vulnerable to certain classes of physical attacks mounted from the IC's backside. Laser-assisted probing, electromagnetic, and body-biasing injection attacks are examples of such attacks. Unfortunately, few countermeasures have been proposed in the literature, and none are available commercially. Those that do exist are not only expensive but also incompatible with current IC manufacturing processes. They also cannot be integrated into legacy systems, such as field programmable gate arrays (FPGAs), which are integral parts of many industrial and defense systems. In this work, we demonstrate how the impedance monitoring of the PCB and IC package's power distribution network (PDN) using on-chip circuit-based network analyzers can detect IC backside tampering. Our method is based on the fact that any attempt to expose the backside silicon substrate, such as the removal of the fan and heatsinks, leads to changes in the equivalent impedance of the package's PDN, and hence, scanning the package impedance will reveal if the package integrity has been violated. To validate our claims, we deploy an on-FPGA network analyzer on an AMD Zynq UltraScale+ MPSoC manufactured with 16 nm technology, which is part of a multi-PCB system. We conduct a series of experiments at different temperatures, leveraging the difference of means as the statistical metric, to demonstrate the effectiveness of our method in detecting tamper events required to expose the IC backside silicon.

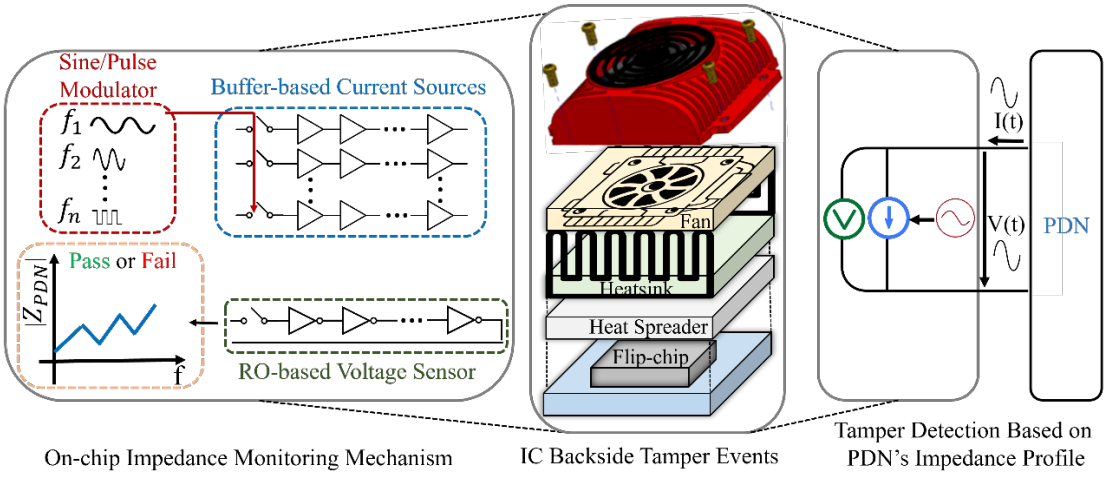


Figure 1: Overview of the proposed IC backside tamper detection methodology.

Who's In Control: Security Risks in Commercial ADAS

Yingjie Cao[1], George Crane[2], Haoqi Shan[3] and Dean Sullivan[2]

[1] The Hong Kong Polytechnic University, Yingjie.cao@polyu.edu.hk
[2] University of New Hamsphire, george.crane@unh.edu
[3] Certik, haoqi.shan@certik.com

1 Introduction

The rapid advancement of autonomous driving technology has significantly propelled the integration of Tier-1 Advanced Driving Assistance Systems (ADAS) into contemporary vehicles. Presently, however, a considerable number of older vehicles lacking modern electronic/electrical (E/E) architecture requisite for ADAS continue to operate on the road. This has catalyzed the development of aftermarket ADAS solutions to retrofit these older vehicles to reap the benefits of autonomous driving. These aftermarket ADAS devices are often designed without consideration of regulations and independently installed in vehicles without guidance for secure integration. Given their capability to capture environmental data, their access to the vehicle's CAN bus, and, critically, their connectivity to external networks, we expected to find a rich body of existing work evaluating their security. Our analysis, however, revealed that a relevant body of existing work was largely absent.

2 Approach

To address these challenges we present an approach, ReverseGear, which automatically reverse engineers car control components to identify their relationship with remote access interfaces that are then crafted into adversarial inputs in aftermarket ADAS systems. We deployed ReverseGear against 16 aftermarket ADAS devices, identifying a total of 35 vulnerabilities, of which 6 present critical security risks that could potentially result in unauthorized vehicle control or the leakage of user privacy information, including video and geographic data. To evaluate the capabilities of ReverseGear beyond aftermarket ADAS devices, we also evaluated a total of 9 Tier-1 ADAS from different manufacturers. Among the 9 Tier-1 manufacturers evaluated, we found 5 vulnerabilities. Importantly, ReverseGear could not find a method to exploit them without first gaining permissions outside of our threat model. To summarize, this work makes three significant contributions:

- **Comprehensive vulnerability analysis.** We conduct the first security analysis of aftermarket ADAS devices. In doing so, we discover a range of new security vulnerabilities. We develop and demonstrate practical system and semantic exploits for these vulnerabilities to highlight their severity.
- **Revealing and Quantifying the Security Difference between Tier-1 and Aftermarket ADAS**. We compare the security level between 9 tier-1 ADAS and 16 aftermarket ADAS. Our results show that Tier-1 ADAS, designed under strict guidelines and regulations, are more resilient.
- **Attack case-study.** We construct 6 different exploit chains, including 2 for Tier-1 ADAS, and 4 for aftermarket ADAS. We do this to demonstrate the real-world impact of these vulnerabilities.

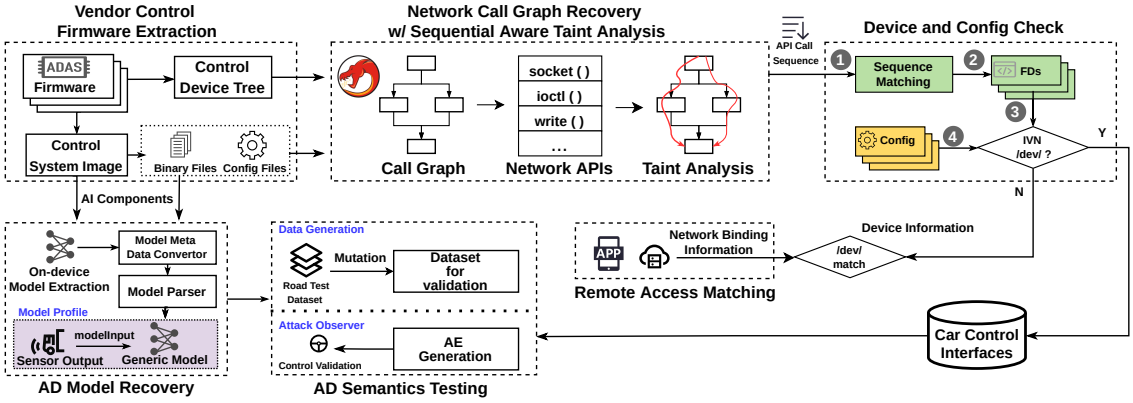


Figure 1: Overview of the ReverseGear.

Energy-Accuracy-Security Trade-offs in Resistive In-Memory Computing Architectures

Saion K. Roy[1,2], and Naresh R. Shanbhag[2]

[1]Northeastern University, Boston, USA, sai.roy@northeastern.edu
[2]University of Illinois at Urbana-Champaign, Urbana, USA, shanbhag@illinois.edu

This is the first work to experimentally characterize the fundamental energy-accuracy-security trade-off in embedded non-volatile memory (eNVM)-based in-memory computing (IMC) architectures. When deployed on Edge devices, resistive IMCs enable on-chip storage of DNN model parameters, reducing the latency and energy costs associated with off-chip data movement. However, storing DNN parameters on-chip expands the attack surface, posing security threats such as model extraction attacks (MEAs). Attackers with access to a victim device can exploit controllable inputs and corresponding observable outputs, creating a strong attack scenario for mounting MEAs. These attacks allow the attacker to retrieve DNN model parameters and reconstruct DNN architectures from various signatures. Since it is possible to infer the training data from the retrieved model, MEAs can lead to the leakage of sensitive private data that may have been used for training the models. Considering the Edge application of resistive IMCs, we ask the question: Does the low compute accuracy of resistive IMCs make them secure?

We hypothesize that the presence of analog noise in eNVM-based IMCs might provide resilience against MEAs. This hypothesis is invalidated by our proposed resistive IMC-specific MEAs developed using circuit-aware stochastic nonlinear functions (Fig. 1(a)). We demonstrate the efficacy of these attacks in extracting the model parameters of the last layer of a ResNet-20 network from the bitcell array of an MRAM-based IMC prototype in a 22nm process. Employing the proposed MEAs, the attacker obtains a CIFAR-10 accuracy within 0.1% of the fixed-point digital baseline (Fig. 1(b)). Our findings reveal that the security vulnerability of resistive IMCs is shown to be a function of compute accuracy (Fig. 1(c)), and we quantify the trade-off between energy-accuracy-security of resistive IMCs. This work opens up opportunities for AI chip designers and security experts to design resistive IMCs that are secure, energy-efficient, and accurate.

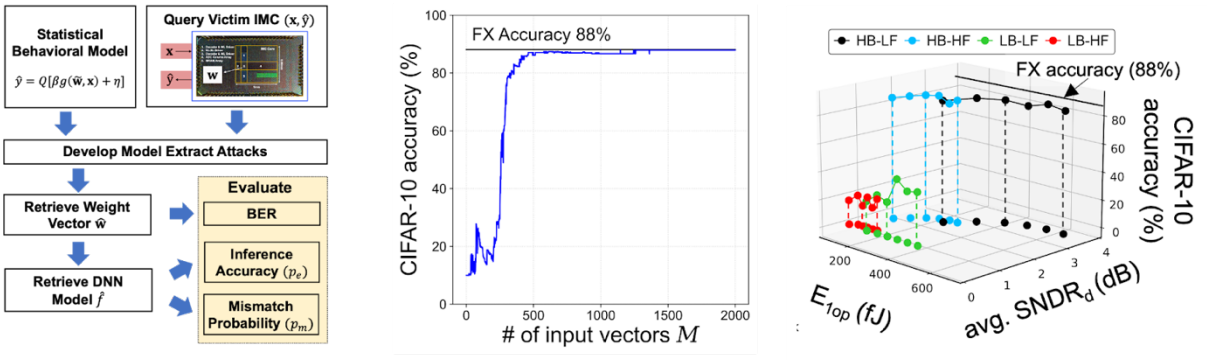


Figure 1: Key contributions of our work including (a) statistical framework to construct MEAs for resistive IMCs, (b) resistive IMC specific MEA achieving high inference accuracy within 0.1% of the fixed point baseline, and (c) energy-accuracy-security trade-off for resistive IMCs.

References

[1] Roy, S.K., Shanbhag, N. R., 2024. *On the Security Vulnerabilities of MRAM-based In-Memory Computing Architectures against Model Extraction Attacks*. ICCAD.

[2] Roy, S.K. and Shanbhag, N.R., 2024, December. *The Energy-Accuracy-Security Trade-off in Resistive In-memory Architectures.* In 2024 IEEE International Electron Devices Meeting (IEDM) (pp. 1-4). IEEE.

[3] Roy, S.K., Ou, H.M., Ahmed, M.G., Deaville, P., Zhang, B., Verma, N., Hanumolu, P.K. and Shanbhag, N.R., 2024. *Compute SNDR-boosted 22-nm MRAM-based in-memory computing macro using statistical error compensation.* IEEE Journal of Solid-State Circuits.

VGF: Value-Guided Fuzzing

Fuzzing Hardware as Hardware

Ruochen Dai[1], Michael Lee[1], Patrick Hoey[2], Weimin Fu[3], Yier Jin[4], Xiaolong Guo[3], Shuo Wang[1], Dean Sullivan[5], Tuba Yavuz[1], Orlando Arias[2]

[1] University of Florida, Gainesville, USA, ruochendai@ufl.edu, michael.lee@student.uml.edu, shuo.wang@ece.ufl.edu, tuba@ece.ufl.edu
[2] University of Massachusetts Lowell, USA, patrick_hoey1@student.uml.edu, orlando_arias@uml.edu
[3] Kansas State University, USA, weiminf@ksu.edu, guoxiaolong@ksu.edu
[4] University of Science and Technology, China
[5] University of New Hampshire, Durham, USA, dean.sullivan@unh.edu

1 Abstract

As the complexity of logic designs increase, new avenues for testing digital hardware becomes necessary. Fuzz Testing (fuzzing) has recently received attention as a potential candidate for input vector generation on hardware designs. Using this technique, a fuzzer is used to generate an input to a logic design. Using a simulation engine, the logic design is given the generated stimulus and some metric of feedback is given to the fuzzer to aid in the input mutation. However, much like software fuzzing, hardware fuzzers use code coverage as a metric to find new possible paths. As we show in this work, this coverage metric is insufficiently generic on hardware designs where designers have taken a more direct approach at expressing a particular microarchitecture of the desired hardware.

In this work we introduce a new coverage metric which employs not code coverage, but state coverage internal to a design. By observing changes in signals within the circuit under testing, we can explore the state space of the design and provide feedback to a fuzzing engine for input generation. Our approach, Value-Guided Fuzzing (VGF), provides a generic metric of coverage which can be applied to *any* design regardless of its implementation. We introduce our state-based VGF metric as well as a sample implementation which can be used with *any* VPI, DPI, VHPI, or FLI compliant simulator, making it completely HDL agnostic. We demonstrate the generality of VGF and show how our implementation is capable of finding bugs considerably faster than previous approaches.

2 Results

Since TheHuzz [2] and HyPFuzz [3] are not publicly available, As such, we are only able to compare our approach to the publicly available HWFuzz [4]. In terms of the number of rounds required to trigger an assertion, VGF achieves a maximum (RS232-T600) and an average speedup of 5760 and 758, respectively, compared to HWFuzz. Regarding execution time, VGF attains a maximum (on RS232-T600) and an average acceleration of 66.7 and 10.5, respectively, in comparison with HWFuzz. Additionally, VGF is able to successfully fuzz and trigger assertions in the design `async_fifo` within the 24-hour timeout window while HWFuz cannot, due to the design's use of multiple clock domains. This substantial enhancement underscores the significance of considering the relevance of signals to a design's control-flow and data-flow with respect to the property signal being tested for effective hardware fuzzing.

References

[1] Michal Zalewski. Google/afl: American fuzzy lop – a security-oriented fuzzer.
[2] Rahul Kande, Addison Crump, Garrett Persyn, Patrick Jauernig, Ahmad-Reza Sadeghi, Aakash Tyagi, and Jeyvijayan Rajendran. TheHuzz: Instruction fuzzing of processors using Golden-Reference models for finding Software-Exploitable vulnerabilities. In 31st USENIX Security Symposium (USENIX Security 22), 2022
[3] Chen Chen, Rahul Kande, Nathan Nguyen, Flemming Andersen, Aakash Tyagi, Ahmad-Reza Sadeghi, and Jyavijavan Rajendran. HyPFuzz: Formal-Assisted processor fuzzing. In 32nd USENIX Security Symposium (USENIX Security 23), 2023
[4] Timothy Trippel, Kang G Shin, Alex Chernyakhovsky, Garret Kelly, Dominic Rizzo, and Matthew Hicks. Fuzzing hardware like software. In 31st USENIX Security Symposium (USENIX Security 22), 2022

USBSnoop: Revealing Device Activities via USB Congestions

Davis Ranney[1], Yufei Wang[1], A. Adam Ding[1], and Dr. Yunsi Fei[1]

[1] Northeastern University, {ranney.d, wang.yufei1, a.ding, y.fei}@northeastern.edu

1 Abstract

The USB protocol has become a ubiquitous standard for connecting peripherals to computers, making its security a critical concern. A recent research study demonstrated the potential to exploit weaknesses in well-established protocols, such as PCIe, and created a side-channel for leaking sensitive information by leveraging congestion within shared interfaces. Drawing inspiration from that, this project introduces an innovative approach to USB side-channel attacks via congestion. We evaluated the susceptibility of USB devices and hubs to remote profiling and side-channel attacks, identified potential weaknesses within the USB standard, and highlighted the critical need for heightened security and privacy in USB technology. Our findings discover vulnerabilities within the USB standard, which are difficult to effectively mitigate and underscore the need for enhanced security measures to protect user privacy in an era increasingly dependent on USB-connected devices.

2 Extended Abstract

The Universal Serial Bus (USB) protocol is now an aptly named nearly universal standard that supports a plethora of peripheral devices to interact with computers. Initially designed to simplify external devices' connection to a host with a standard connector and protocol, USB has evolved significantly in functionality and speed. USB operates as a packet-based communication system, utilizing the concept of pipes and communication layers, which logically resembles IP-based networking and Peripheral Component Interconnect Express (PCIe). A critical aspect of USB technology is the integration of hubs, which serve as instrumental components in expanding the connectivity of a single computer or controller to multiple devices. Hubs not only enhance the versatility of USB but also add layers of complexity to the data traffic management.

This research explores a critical yet under-examined aspect of USB security and privacy, demonstrating how USB hubs can be exploited to recover sensitive user data by monitoring patterns of congestion among devices connected to USB hubs. Unlike traditional eavesdropping attacks, this method does not require physical access to the device but operates within common configurations used by regular users without needing administrator privileges. Using these hub-level side-channels, an attacker can recover a victim user's internet browsing history, keystrokes, and possibly even more sensitive data transmitted across USB.

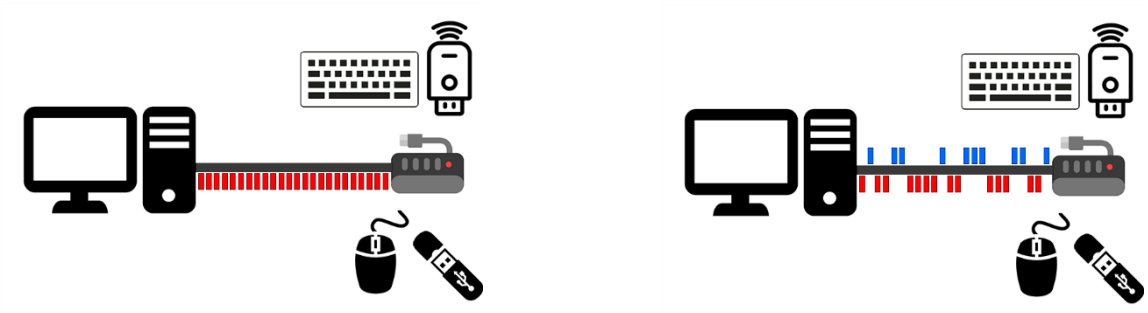


Figure 1: Spy devices (red data) utilize the full bandwidth of a USB connection to a hub, then as the victim devices (blue data) share the bandwidth of the hub, they cause measurable congestion to the spy devices.

Exploiting Exclusive System-Level Cache in Apple M-Series SoCs for Enhanced Cache Occupancy Attacks

Tianhong Xu[1], Aidong Adam Ding[2] and Yunsi Fei[1]

[1] Northeastern University, xu.tianh@northeastern.edu, a.ding@northeastern.edu, y.fei@northeastern.edu

Abstract

Cache attacks [4] exploit microarchitectural features of modern processors, such as timing differences between cache hits and misses, to leak sensitive information. Cache occupancy attacks [1] are a nuanced variant that targets the overall state of cache occupancy to infer sensitive information, breaching privacy and confidentiality. The popular access-driven cache attacks focus on specific cache sets or lines to infer memory address-related secrete.

ARM's big.LITTLE architecture exemplifies heterogeneous computing systems by combining high-performance "big" cores with energy-efficient "LITTLE" cores. ARM introduced the System Level Cache (SLC), an exclusive last-level cache situated between the higher-level caches (L1 and L2) and main memory [2]. Apple's M-series System-on-Chips (SoCs) build upon ARM's heterogeneous designs, featuring multiple CPU clusters with local caches, an integrated GPU, and an SLC shared among clusters and the GPU. While this architecture offers significant performance benefits, it also introduces new security challenges due to the shared SLC between clusters and GPU.

In this talk, we present a novel suite of cache occupancy attacks targeting the SLC of Apple M-series SoCs—the first to exploit an exclusive last-level cache, where an adversary can monitor GPU and other CPU cluster activities from their own CPU cluster. By reverse- engineering the SLC's sharing mechanism and structure in the Apple M1, we obtain critical insights that enable these attacks. We demonstrate the effectiveness of the SLC occupancy side-channel through three attacks:

(1) Website Fingerprinting Attack: We perform a website fingerprinting attack, showing that our method achieves high precision across a wider range of scenarios, including cross-browser setups, where previous cache occupancy channels fail [1].

(2) Cross-Origin Pixel Stealing Attack: We conduct a finer- grained cross-origin pixel stealing attack by exploiting the SLC occupancy channel. We leverage the data-dependent behavior of GPU rendering and compression, and accurately retrieve the screen display pixel-by-pixel, violating confidentiality and privacy. Unlike previous attacks that relied on measuring the rendering time, our approach uses the SLC occupancy side-channel to extract pixel-level information. It is effective even in the presence of constant rendering-time implementations and Apple's recent security fixes addressing related vulnerabilities (CVE-2023-38599 [3]).

(3) Screen Display Snooping Attack: Notably, we introduce a novel attack that does not rely on website-processing tools. By monitoring the GPU's rendering processes via the SLC occupancy channel, we can extract any information displayed on the screen. This significantly expands the attack surface beyond web pages, allowing the adversary to compromise any on-screen information, posing a substantial new threat to system security.

References

1. Anatoly Shusterman, Ayush Agarwal, Sioli O'Connell, Daniel Genkin, Yossi Oren, and Yuval Yarom. 2021. {Prime+ Probe} 1,{JavaScript} 0: Overcoming Browser- based {Side-Channel} Defenses. In 30th USENIX Security Symposium (USENIX Security 21). 2863–2880.
2. ArmLimited.2022.ArmCoreLinkCI-700CoherentInterconnectTechnicalReference Manual. Arm Limited. https://www.arm.com/corelink-ci700-manual
3. NIST. 2023. CVE-2023-38599. https://nvd.nist.gov/vuln/detail/CVE-2023-38599.
4. Mehmet Kayaalp, Nael Abu-Ghazaleh, Dmitry Ponomarev, and Aamer Jaleel. 2016. A high-resolution side-channel attack on last-level cache. In Proceedings of the 53rd Annual Design Automation Conference. 1–6.

Uncertainty Estimation in Neural Network-enabled Side-channel Analysis and Links to Explainability

Seyedmohammad Nouraniboosjin, Fatemeh Ganji
Worcester Polytechnic Institute,
{snouraniboosjin,fganji}@wpi.edu

1 Introduction

Assessing the security of implementations against Side-Channel Analysis (SCA) is a challenging task [1]. Since its introduction in the 1990s, research has focused on analyzing physical leakage, such as power consumption, timing, and electromagnetic emissions. Profiled SCA, which estimates leakage models using an open sample, represents the worst-case attack scenario, relying on knowledge of the device's leakage distribution. However, optimal attack strategies depend on accurately modeling this distribution, which is highly complex, especially in the presence of countermeasures like masking. In cases where the probability density function (PDF) of the leakage is unknown, deep neural networks (NNs) have emerged as sub-optimal but effective alternatives.

Despite the success of NN-based SCA, a critical challenge remains—understanding predictive uncertainty. Unlike traditional machine learning, where accuracy is a standard performance metric, SCA relies on key rank, guessing entropy, and success rate [2]. However, NNs are often overconfident in their predictions, leading to incorrect key guesses with high probabilities. This uncertainty stems from multiple factors, including measurement errors, randomness in physical quantities, and variability in NN training. Understanding whether this uncertainty arises from inherent data characteristics or can be mitigated through better training is crucial. Additionally, if data uncertainty dominates, identifying specific trace features responsible for misclassification becomes essential.

2 Approach

We introduce a novel approach to uncertainty estimation in deep learning-based SCA by leveraging Rényi entropy as a measure of uncertainty in key recovery. Unlike traditional entropy measures, Rényi entropy provides a more generalized framework for capturing different types of uncertainty, allowing for a more refined analysis of model confidence. By applying this metric, we quantify the level of uncertainty in neural network predictions and assess how it influences key recovery effectiveness. We also decompose the uncertainty to epistemic and aleatoric entropies, corresponding to the uncertainty in the model weights and data (see an illustrative example in Figure 1) [3]. Since direct PDF estimation is infeasible in high-dimensional data, we employ matrix-based Rényi's α-entropy. We also propose using α-divergence to approximate the leakage distribution, overcoming the limitations of Kullback-Leibler divergence. We also analyze whether uncertainty arises from data or model-related factors. For example, we demonstrate how desynchronization in traces increases model uncertainty, linking it to attack performance. This helps evaluators determine if uncertainty is reducible by feeding more traces or if it fundamentally limits attack success. Using SHapley Additive exPlanations (SHAP), we further identify which time instances in a trace contribute most to uncertainty. Since leakage traces exhibit dependencies across time, conventional sensitivity analysis is ineffective. By integrating SHAP with uncertainty estimation, we pinpoint the most critical time features affecting model uncertainty. Finally, we establish a strong relationship between uncertainty and standard SCA evaluation metrics such as Guessing Entropy (GE) and Success Rate (SR). Higher predictive uncertainty often corresponds to harder-to-break key classes, providing an additional evaluation metric for attack performance.

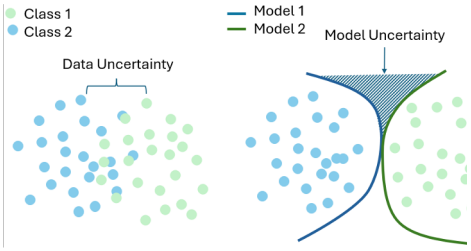


Figure 1: Uncertainty decomposition in a binary classification task

References

[1] Chari, S., Rao, J.R. and Rohatgi, P., 2002, August. Template attacks. In International workshop on cryptographic hardware and embedded systems (pp. 13-28). Berlin, Heidelberg: Springer Berlin Heidelberg.

[2] Standaert, F.X., Malkin, T.G. and Yung, M., 2009. A unified framework for the analysis of side-channel key recovery attacks. In Advances in Cryptology-EUROCRYPT 2009: 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 443-461). Springer Berlin Heidelberg.

[3] Gal, Y., 2016. Uncertainty in deep learning.

Oreo: Protecting ASLR Against Microarchitectural Attacks

Shixin Song[1], Joseph Zhang[1] and Mengjia Yan[1]

[1] Massachusetts Institute of Technology, shixins@mit.edu, jzha@mit.edu, mengjiay@mit.edu

1 Research Problem

Address Space Layout Randomization (ASLR) is one of the most prominently deployed mitigations against memory corruption attacks. ASLR randomly shuffles program virtual addresses to prevent attackers from knowing the location of program contents in memory.

However, in the current virtual-physical memory interface (Figure 1(a)), the ASLR secret is spread through page tables and other microarchitectural structures (caches, BTB, TLB, etc.), thereby leaked by various side-channel attacks. For example, Binoculars [1] leaks the ASLR secret by triggering the victim fetching instructions using secret-dependent virtual addresses, and the processor will walk through page tables to translate these addresses. During this process, the secret bits are used to index into page tables and modulate caches, so the attacker can leak the secret offset by monitoring cache states.

2 Proposed Solution

Motivated by this, we present *Oreo*, a software-hardware co-design mitigation that strengthens ASLR against these attacks. The core of *Oreo* is a new memory interface, as shown in Figure 1(b), where *Oreo* introduces a new layer of memory, called the masked memory, sitting between the virtual and physical memory. A masked address is constructed from a randomized virtual address by redacting the ASLR secret. All the hardware structures that used to use virtual addresses as input now switch to using secret independent masked addresses. The new memory interface ensures the ASLR secret stays microarchitectural oblivious, blocking all the side channels.

We prototyped and evaluated our design on Linux using the hardware simulator gem5 [2]. We show that our design introduces negligible performance overhead running the SPEC2017 IntRate benchmark [3] and the LEBench benchmark [4].

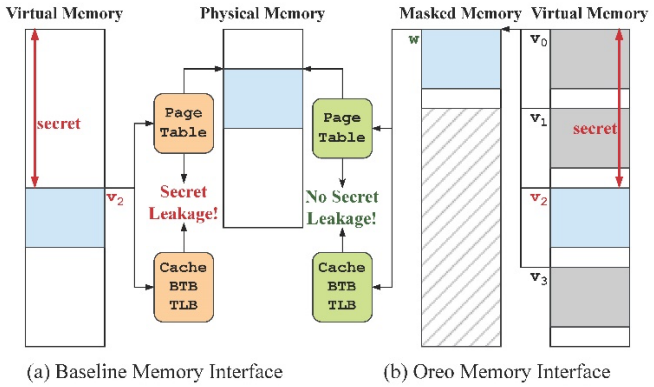


Figure 1: Overview of *Oreo*'s new memory interface.

References

[1] Zhao, Zirui Neil et al. (2022). "Binoculars: Contention-Based Side-Channel Attacks Exploiting the Page Walker". In: *31st USENIX Security Symposium (USENIX Security 22)*, pp. 699–716.

[2] Binkert, Nathan L. et al. (2011). "The gem5 simulator". In: *SIGARCH Comput. Archit. News* 39.2, pp. 1–7. DOI: 10.1145/2024716.2024718. URL: https://doi.org/10.1145/2024716.2024718.

[3] Bucek, James, Klaus-Dieter Lange, and Jóakim v. Kistowski (2018). "SPEC CPU2017: Next-generation compute benchmark". In: *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering*, pp. 41–42.

[4] Zirui Zhao (2022). *LEBench-Sim: A Benchmark Suite for Large-Eddy Simulation*. https://github.com/zzrcxb/LEBench-Sim.

Cross Layer EMFI Assessment Framework

Hanqiu Wang[1], Ruochen Dai[1], Tuba Yavuz[1], Xiaolong Guo[2], Orlando Arias[3], Dean Sullivan[4], Michael Lee[1], Honggang Yu[1], Siqi Dai[1], Domenic Forte[1], Shuo Wang[1]

[1] University of Florida, wanghanqiu@ufl.edu, ruochendai@ufl.edu, tuba@ece.ufl.edu, Michael.lee@ufl.edu, honggang.yu@ufl.edu, dais@ufl.edu, dforte@ece.ufl.edu, shuo.wang@ece.ufl.edu

[2] Kansas State University, guoxiaolong@ksu.edu

[3] University of Massachusetts Lowell, orlando_arias@uml.edu

[4] University of New Hampshire, dean.sullivan@unh.edu

1 Introduction

Electromagnetic Fault Injection (EMFI) poses a significant threat to integrated circuits by inducing register-level faults through transient voltage glitches. Existing modeling and simulation efforts[1] have largely remained confined to narrow abstraction layers, limiting their scalability and practical relevance in modern VLSI systems. In this talk, we will present a cross-layer EMFI assessment framework that bridges the gap between transistor-level behavior and system-level impact.

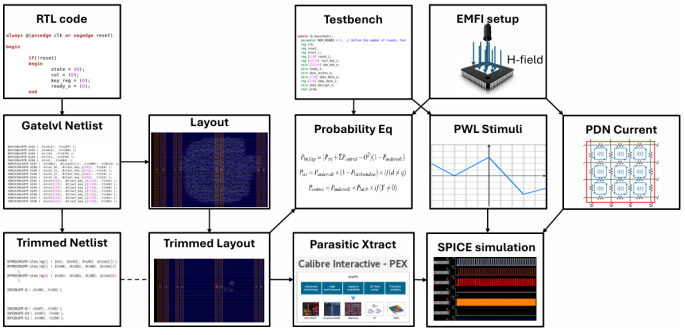
Our framework, shown in Figure 1, begins with gate-level netlists and physical layouts, trims the design to focus on vulnerable register paths, extracts parasitics, and performs SPICE simulations to accurately model voltage fluctuations under EMFI. To address scalability challenges, we propose a set of probability equations that estimate register bitflip likelihoods under various EMFI conditions. We extend the analysis to conditional bitflips affecting multiple registers simultaneously.

This framework is the first to quantitatively and scalably assess EMFI susceptibility across IC design layers—from RTL and gate level down to parasitic-aware SPICE simulations—making it a practical pre-silicon security evaluation tool. Future work includes integrating this work with hardware fuzzing[2] and Bayesian system-level fault analysis[3] to form a holistic hardware security assessment pipeline.

2 Evaluation

We did evaluation on 5 different benchmarks including cryptographic cores and Input/Output protocols. 5000+ iterations of SPICE simulation is run on each benchmark and the discrepancy between the SPICE simulation results and equation calculation results falls under 10%. The discrepancy is mainly caused by two or more control signals are undervolted simultaneously.

Figure 1: Overview of the proposed EMFI assessment framework.



References

[1] Dumont, M., Lisart, M. and Maurine, P., 2020. Modeling and simulating electromagnetic fault injection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 40(4), pp.680-693.

[2] Trippel, T., Shin, K.G., Chernyakhovsky, A., Kelly, G., Rizzo, D. and Hicks, M., 2022. Fuzzing hardware like software. In 31st USENIX Security Symposium (USENIX Security 22) (pp. 3237-3254).

[3] Vallero, A., Savino, A., Politano, G., Di Carlo, S., Chatzidimitriou, A., Tselonis, S., Kaliorakis, M., Gizopoulos, D., Riera, M., Canal, R. and González, A., 2016, November. Cross-layer system reliability assessment framework for hardware faults. In 2016 IEEE International Test Conference (ITC) (pp. 1-10). IEEE.

IvLeague: Side Channel-resistant Secure Architectures Using Isolated Domains of Dynamic Integrity Trees

Md Hafizul Islam Chowdhuryy and Fan Yao

University of Central Florida, hafizul.islam@ucf.edu, fan.yao@ucf.edu

1 Abstract

Trusted computing has become crucial due to concerns about trust in remote computing (i.e., cloud environments). Modern approaches use secure architectures that considers the processor as the root of trust, with hardware-based encryption and integrity checks to protect data. Industry solutions (i.e., Intel SGX, AMD SEV) create trusted execution environments (TEEs) to safeguard against threats to both hardware and privileged software. While off-chip data protection is essential, securing on-chip data has become equally critical as microarchitectural attacks, such as timing channels, continue to evolve. These attacks can extract sensitive information by manipulating states within on-chip resources, creating new attack vectors in microarchitecture security domain. Although secure processors typically do not include side channels in their threat models, recent findings reveal that microarchitecture security cannot be isolated from other security concerns. Notably, the integrity verification (IV) tree, a global structure for detecting off-chip tampering, can introduce shared-memory side channels due to implicit sharing of IV nodes across security domains, such as enclaves. This unintended metadata sharing creates new shared-memory side channels, even when traditional data-sharing attacks like Flush+Reload are mitigated. Unfortunately, existing defenses such as randomization and partitioning cannot adequately mitigate this metadata-based leakage. Unlike traditional side channels that exploit shared hardware resources, this new vulnerability arises from shared metadata, which highlights the need for more holistic security solutions, as existing defenses like randomization and partitioning are insufficient to address these vulnerabilities.

In our IvLeague paper published in MICRO'24, we investigate further to understand the root of the metadata-sharing vulnerabilities. IvLeague further proposes architecture support for side channel-resistant isolated integrity trees among dynamic domains in secure processors. At a high level, IvLeague splits the global integrity tree into many small statically-addressed subtrees (called TreeLings). Metadata sharing is prevented among TreeLings by keeping their roots on-chip. IvLeague enables efficient runtime scaling of memory coverage (upto entire system memory) by assigning and detaching TreeLings to each individual domain. We propose several optimizations to the IvLeague framework. The first optimization, IvLeague-Invert, shortens the path of integrity verification from leaf to root by directly mapping data pages to high-level intermediate nodes and gradually introducing nodes from lower levels (i.e., intra-TreeLing extension) only when all nodes in certain top levels are occupied. The second optimization, IvLeague-Pro, targets pages with high access frequency, or hotpages, which can significantly impact system performance. IvLeague-Pro allocates a dedicated sub-region within each IvLeague for hotpages, along with a lightweight hotpage tracker integrated into the memory controller. When a page is flagged as a hotpage, IvLeague-Pro relocates it closer to the root, minimizing access time. We demonstrate that IvLeague-based schemes effectively prevent side channel leakage originating from shared integrity trees, all while incurring minimal overheads

Evaluation of IvLeague schemes: We implement and evaluate IvLeague in the gem5 simulator. We compare performance of IvLeague on against state-of-the-art secure architecture design (susceptible to metadata-sharing attack) [Figure 1]. Based on our evaluations, IvLeague with its optimizations consistently demonstrates performance speedup up to 19% (14% on average) over Baseline, while preventing the metadata-based leakage. In addition, we exhaustively performed analysis on NFL to determine its effectiveness. We observe NFL has near perfect utilization (i.e., deallocated page tracking) at above 99.97% across all workloads. We further perform scalability analysis of IvLeague compared to state partitioning scheme and demonstrate that static partitioning is not able to support diverse memory footprint workloads, which IvLeague can support properly.

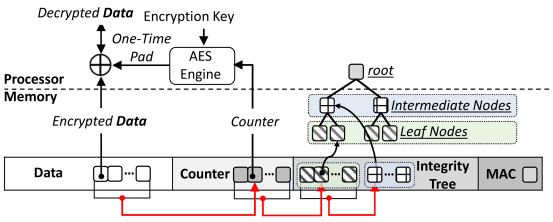


Figure 1: Secure architecture mechanisms.

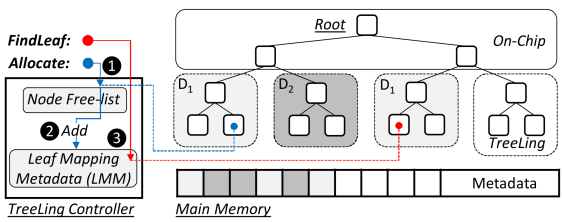


Figure 2: High-level operation of IvLeague.

List of Accepted Posters

Author(s)	Title
Shahriar Hadayeghparast, Xiang Li, Aleksa Deric, Daniel Holcomb	On-die Differential Sensing for Monitoring and Analysis of Dynamic Computing Environments
Kaiyuan Zhang, Antian Wang, Keshab Parhi, Yingjie Lao	Hardware Acceleration for Fully Homomorphic Encryption Scheme Switching from CKKS to FHEW
Fatemeh Khojasteh Dana, Saleh Khalaj Monfared, Shahin Tajik	Logical Maneuvers: Detecting and Mitigating Adversarial Hardware Faults in Space
Theodoros Trochatos, Christopher Kang, Frederic T. Chong, Jakub Szefer	Side-channel Threats in Fault-Tolerant Quantum Computers
Yizhuo Tan, Hrvoje Kukina, Jakub Szefer	Securing HHL Quantum Algorithm against Quantum Computer Attacks
Suraj Mandal, Debapriya Basu Roy	Winograd for NTT: A Case Study on Higher-Radix and Low-Latency Implementation of NTT for Post Quantum Cryptography on FPGA
Mashrafi Alam Kajol, Sandeep Sunkavilli, Qiaoyan Yu	An On-Chip Sensor Placement Strategy for Voltage-Drop Attack Mitigation
Tom Slooff, Anthony Etim, Jiaqi Yu, Francesco Regazzoni, Jakub Szefer	Fault Injection on Reinforcement Learning
Saleh Monfared, Maryam Saadat Safa, Shahin Tajik	ChipletPing: On-die Digital Impedance Sensing for Chiplet and Interposer Verification
Zirui Fu, Marco Donato	Reverse Cross Entropy Optimizations for Efficient Adversarial Detection and Defense
Dillibabu Shanmugam, Zhenyuan Liu, Patrick Schaumont	CAPRI6: A Solution for Fault Root Cause Detection
Andrew Malnicof, Zhenyuan Liu, Patrick Schaumont	LeakFlow: Power Side-Channel Leakage Simulation and Assessment Tool
Chuanqi Xu, Jakub Szefer	Security Attacks Abusing Pulse-level Quantum Circuits
Muhammad Faheemur Rahman, Wayne Burleson	Integrated Security Mechanisms for Weight Protection in Memristive Crossbar Arrays
Tanzim Mahfuz, Prabuddha Chakraborty	X-DFS: Explainable Artificial Intelligence Guided Design-for-Security Solution Space Exploration
Evan Apinis, Zhenyuan Liu, Patrick Schaumont	LooseWire: Identifying Root Causes of Side-channel Leakage in an Open-Source FPGA
Dev M. Mehta, Maryam Saadat Safa, Alessandra Savio Serpes, Seyedmohammad Nouraniboosjin, Shahin Tajik, Fatemeh Ganji	AI-enabled, Automated, and Efficient Electromagnetic Side-Channel Acquisition
Spencer Harding, Keegan Kuhn, Scott West, Dev M. Mehta, Shahin Tajik, Fatemeh Ganji	Automation of Photon Emission Analysis Pipeline for Hardware Security
Yashaswini Makaram, Yunsi Fei, David Kaeli	Chosen Cyphertext Attack on Barret Reduction in CRYSTALS-Kyber
Yufei Wang	Cross CPU-GPU Rowhammer Attack