Detecting Microarchitectural Side-Channel Attacks via Hardware Security Checkers

Alessandro Palumbo*

CentraleSupélec, Inria, Univ Rennes, CNRS, IRISA

Abstract

Microarchitectural Side-Channel Attacks represent significant challenges to the security and reliability of modern microprocessor-based systems. This manuscript introduces an approach to enhance hardware security. A programmable Hardware Security Checker is proposed to detect Microarchitectural Side-Channel Attacks by employing hash functions and Machine Learning algorithms to analyze runtime features, such as performance counters, enabling the real-time detection of attack patterns.

Introduction & Motivation

Microarchitectural Side-Channel Attacks (MSCAs) consist of malicious software capable of inferring sensitive information by analyzing microprocessor features that are seemingly unrelated to program execution. Attackers exploit timing information, power consumption, thermal footprints, or electromagnetic emanations of computing systems to extract sensitive data [1]. This extracted information can then be used to infer critical details about the microprocessor's operations. Notable examples of MSCAs include *Spectre* [2] and *Meltdown* [3]. Building upon these foundational attacks, numerous other MSCAs have been developed [1]. This manuscript addresses the growing need for robust hardware-based security mechanisms to protect microprocessor-based systems against MSCAs: integrating Hardware Security Checkers (HSCs) into microprocessor-based systems, looking at their features to tackle such attacks.

The HSCs monitor the runtime behavior of the microprocessor and identify attacks through the analysis of its features and the detection of specific "attack signatures." Unlike software-based defenses, which can struggle to detect circuit-level vulnerabilities or be bypassed by attackers, HSCs operate directly at the hardware level, making them uniquely positioned to identify and mitigate these threats. Key features of the proposed approaches include:

- **Programmability and Flexibility:** HSCs can be reconfigured and reprogrammed after deployment to address the detection of even new attacks.
- **High Accuracy and Low Overhead:** The HSCs achieve 100% detection accuracy with no false alarm rates and no impact on the microprocessor performance.
- **Transparency and Efficiency:** The HSCs operate seamlessly with the microprocessor.

*Corresponding author: `alessandro.palumbo@inria.fr`

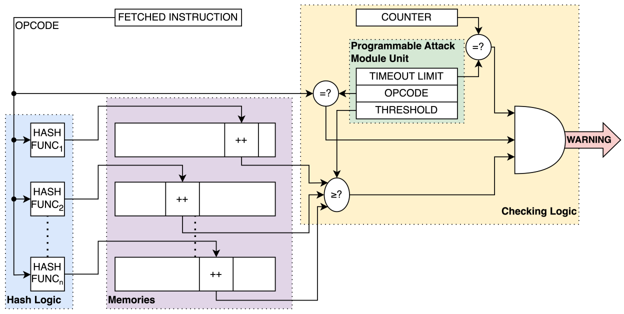

Figure 1: *The CMS-based HSC structure [4]*

The Hardware Security Checker

The HSC shown in Figure 1 observes the microprocessor's fetching activity, specifically focusing on instruction patterns and their frequencies. Leveraging a Count-Min Sketch (CMS) probabilistic data structure, the module estimates the occurrence frequencies of the hash of the instructions associated with specific attack models. These attack models, programmed by the user, are defined in terms of recognizable patterns and thresholds that must be met to classify an activity as malicious. The system operates in a time-window-based mode, during which it tracks all fetched instructions and identifies suspicious activity at the end of each window. Figure 2 shows the flow for detecting the attacks.

A complementary design flow for creating a HSC builds upon a machine-learning-based methodology to enhance attack detection capabilities. This approach starts with extensive system-level simulations of the microprocessor under protection executed while simulating known attacks. This initial phase aims to generate a comprehensive database of features for training the Machine Learning (ML) model integrated into the HSC. These features include a wide range of microprocessor behaviors and runtime characteristics, such as:

- #data cache writebacks;

Table 1: *Best rates and the resource usage of proposed solutions*

Ref.	**Acc.**	**FP**	**FN**	#LUT	#FF	#BRAM	#LUTRAM
[4]	100%	0%	0%	19063 (3.98%)	11552 (6.13%)	17	4536 (0.53%)
[5]	99.60%	0.40%	0%	45,200 (6.75%)	6,100 (NA)	0	0

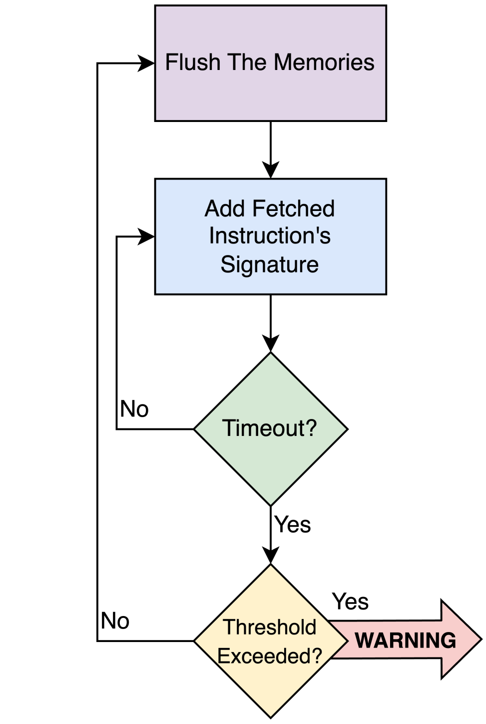

Figure 2: *The workflow of the CMS-based HSC [4]*

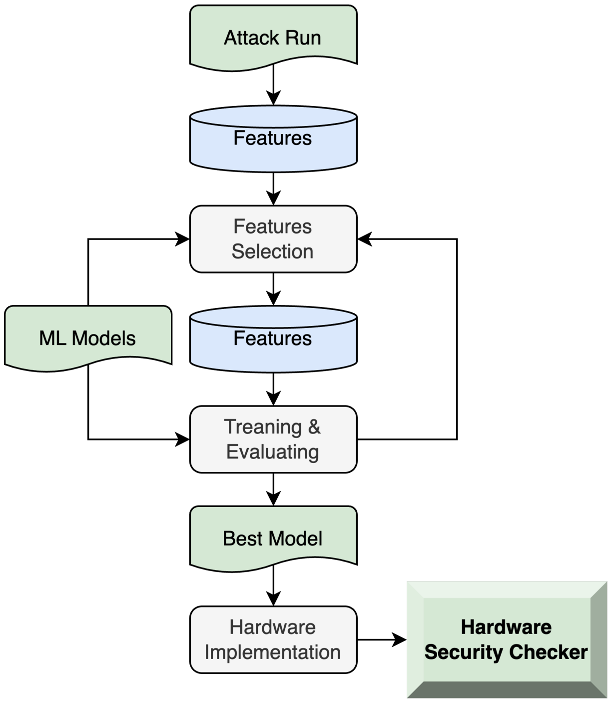

Figure 3: *The workflow of the ML-based HSC [5]*

- #data cache hits and misses;
- #branch mispredictions;
- ...

Later on, the optimal subset of features for training the ML model will be detected. This selection is guided by the dual objectives of maximizing anomaly detection accuracy and minimizing false alarms while considering the impact of the number of selected features on the HSC's implementation overhead.

The final phase involves a high-level synthesis procedure that translates the trained ML model into hardware-ready code for deployment. The whole workflow is reported in Figure 3

A key advantage of both HSCs is their ability to detect also new attacks. In the first approach, the HSC can be easily reprogrammed with updated attack models as new threats are discovered. Similarly, the second approach allows the design flow to be restarted to incorporate the new attack scenario. So, the HSCs maintain their relevance and effectiveness against emerging vulnerabilities.

This manuscript presented innovative approaches to enhancing the security of modern microprocessor-based systems by addressing Microarchitectural Side-Channel Attacks. The feasibility of detecting and mitigating these threats in real-time with no performance and minimal resource overhead is achieved through programmable Hardware Security Checkers and side-channel analysis methodologies. Table 1 reports the attack detection rates and the resources used, indicating the percentage overhead with respect to the unprotected microprocessor not implementing any Hardware Security Checker.

References

[1] Jie Yuan et al. "A Survey of of Side-Channel Attacks and Mitigation for Processor Interconnects". In: *Applied Sciences* 14.15 (2024), p. 6699.

[2] P. Kocher et al. "Spectre Attacks: Exploiting Speculative Execution". In: *40th IEEE Symposium on Security and Privacy (S&P'19)*. 2019.

[3] M. Lipp et al. "Meltdown: Reading Kernel Memory from User Space". In: *27th USENIX Security Symposium (USENIX Security 18)*. 2018.

[4] K. Arıkan et al. "Processor Security: Detecting Microarchitectural Attacks via Count-Min Sketches". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 30.7 (2022), pp. 938–951. DOI: 10.1109/TVLSI.2022.3171810.

[5] Mattia Iamundo. "A machine learning-based security architecture to detect microarchitectural side-channel attacks in microprocessors". In: (2021).