Holistic Hardware Security Assessment Framework: A Microarchitectural Perspective

Tochukwu Idika Ismail Akturk
Electrical Engineering and Computer Science, *University of Missouri-Columbia*
tnifgf@mail.missouri.edu akturki@missouri.edu

***Abstract*—Our goal is to enable holistic hardware security evaluation from the microarchitectural point of view. To achieve this, we propose a framework that categorizes threat models based on the microarchitectural components being targeted, and provides a generic security metric that can be used to assess the vulnerability of components, as well as the system as a whole.**

***Index Terms*—hardware security, security metrics, threat models, security assessment**

I. Introduction

For decades, security aspect of systems has been overlooked behind the unceasing effort to improve the performance. As performance becomes restricted by the degree of energy-efficiency, the power management optimizations (e.g., dynamic voltage and frequency scaling) start to play a significant role in microarchitectural design and constraints. However, these optimizations may create side channels. Differential and leakage power analysis are the two types power monitoring side channel attack [1]. Differential power analysis (DPA) utilizes the correlation between input data and the dynamic power consumption of the microarchitectural components [2], whereas the leakage power analysis (LPA) employs the correlation between the input data and leakage power dissipation of the given component [3]. Evidently, hardware itself becomes a target for security attacks (especially by exploiting the side effect of energy/power optimizations in hardware) forcing system architects to consider the security as first-class design constraint, similar to performance and energy efficiency. While such realization is promising, the existing threat model descriptions are ad-hoc, and security metrics are ill-defined (or non-existing), making it challenging to have fair and consistent vulnerability analysis among different microarchitectural components, optimizations, and system architectures. To address this challenge, we propose a framework that provides systematic categorization of threat models, and describes a security metric formulation which can be used to evaluate vulnerability of microarchitectural components, as well as the system as a whole.

For many microarchitectural exploration and assessment, we are used to rely on well-defined set of applications, such SPEC Benchmarks [4]. These benchmarks allow architects to assess the impact of different microarchitectural components (and system as a whole) under various execution patterns on performance and/or energy-efficiency. Each benchmark can stress a particular microarchitectural component, and the evaluation can be reported in a well-defined, standardized way that allows fair comparison of alternatives. However, currently, we have no equivalent mechanism to perform hardware security assessment, neither at microarchitectural-level, nor at the system-level. The proposed framework would fill this gap. In this framework, there will be standardized definition of threat models (similar to benchmarks), and unified (i.e., universal) security metric that allows comparison of vulnerabilities of microarchitectural components (and whole system) to the respective threat models. In the following sections, we detail how threat models can be classified from a microarchitectural perspective, and how a security metric can be developed to be used as an individual (and global) assessment tool.

II. Classification of Threat Models

Each threat model should be described in a standardized way that would allow system architects to reproduce the attack on the system being evaluated. Similar to applications in benchmark suites, a particular threat model may target different microarchitectural component (e.g., cache, branch predictor, memory). So, an architect who is interested in assessing a particular component's security vulnerability should be able to identify which threat models are targeting that particular component.

In the proposed framework, each threat model can be categorized with respect to the microarchitectural component it targets. Fig. 1 illustrates such a categorization. As an example, if an architect is trying to assess the security vulnerability of the cache in his/her design, then (s)he has to experiment with threat models Y (a timing side channel attack), meltdown (side channel attack) and W (a power side channel attack). Without experimenting with all known attacks targeting the given component, it is impossible to make a fair and consistent assessment with respect to other alternative design choices for that particular microarchitectural component.

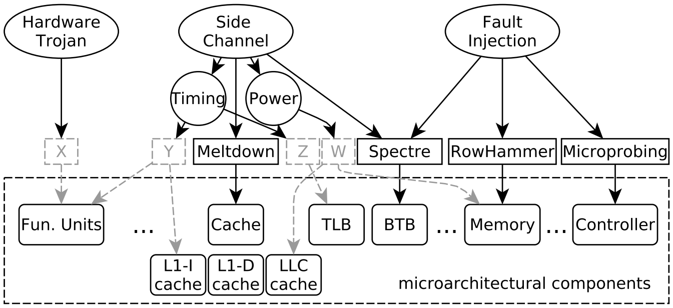

Fig. 1. Microarchitectural-based threat model categorization. X, Y, Z, W should be known threat models, for illustration purposes, here they are shown generically.

By using such a categorization given above, it is also possible to have holistic security assessment of a given system. In its simplest form, it would be an aggregation of evaluations performed for each microarchitectural component in the given system. To make a quantitative assessment and aggregation of separate assessments (for each component), we need a unified security metric that is representative for all possible components that can exist in any system and all threat models defined in the framework. In the following, we discuss how such unified security metric can be defined and used for holistic hardware security assessment.

III. Security Metric for Holistic Assessment

The previous studies proposed separate (and mostly incompatible) metrics for different attacks targeting different components [5], [6], hence, there is no comprehensive way of combining the results to have holistic hardware security assessment of an entire system. To address that, we propose a quantitative metric that abstracts software and network layers of the system and focuses the vulnerability of microarchitectural components. The proposed metric is generic, but universal, and called Vulnerability Factor (VF). It is similar in essence to side-channel vulnerability factor (SVF) [7] that quantifies patterns observed by the attackers and measures the correlation to the victims actual execution patterns. Cache side channel vulnerability metric (CSV) [8] is a narrow but a more pragmatic version of SVF. Despite the similarities, proposed VF it is not specific to a particular attack model (as opposed to [7] and [8]). In general, VF describes a metric that allows to measure the correlation between an operation performed on a particular microarchitectural component and the observation made by a specific threat model targeting the given component. Each component in the system has a VF under a given threat model. So, the overall VF for a component would become the aggregation of separate VFs under different threat models. Assume a set of components ($ComponentSet$) and a set of threats ($ThreatSet$):

$$VF_{comp_i} = \sum_{t=1}^{N} VF_{i,t} \times w_t$$

where VF_{comp_i} represents the aggregated vulnerability factor of *i'th component* ($comp_i \in ComponentSet$), $VF_{i,t}$ represents the vulnerability factor of *component i* under the *threat model t* (*threat model t* $\in ThreatSet$ and $1 \leq t \leq N$, where N is the number of threat models known), and w_t represents a weight of the given *threat model t*. The weights exist to provide flexibility to an architect to set relative importance for the threats – if all threats are equally important for an architect, then the weights will be same. When reporting the VF_{comp_i}, it is expected to report associated weights as well (default is 1).

The assessment of VF_{comp_i} of different systems would allow an architect to evaluate how various design and implementation choices of the given component could impact its overall vulnerability.

Similarly, the VF of the whole system can be measured as the aggregation of VFs of each component exist in the system:

$$HVF_x = \sum_{i=1}^{M} VF_{comp_i} \times w_i$$

where HVF_x represents the holistic vulnerability factor of a given system (e.g., *System X*), VF_{comp_i} represents the aggregated vulnerability factor of *i'th component* ($comp_i \in CompoonetSet_X$ and $1 \leq i \leq M$, where M is the number of components in the system), and w_i represents a weight of the given *component i*. The weights have similar semantic as mentioned above.

Note that the actual derivation of VF may vary as component and/or threat model changes. Here, we do not advocate a single VF derivation, but we provide a generic metric whose specific derivation should be determined by the community. Once a particular VF for a component under a threat model is determined, it can be standardized as part of the proposed framework which would allow the holistic security assessment of a system from the microarchitectural perspective.

IV. Challenges and Future Work

There are two fundamental challenges for quantitative holistic hardware security assessment. The first challenge is the fact that security is a moving target, meaning that vulnerability assessments and quantification are limited by the threat models that we know today (i.e., there is no way to know all possible threats beforehand that may exist). As a result, a system deemed secure considering threat models that we know today, could prove to be totally insecure tomorrow when a new threat model is identified. When a new threat model is identified, it should be added into the proposed framework (similar to adding a new application into a benchmark suite). That would translate into a new version of the framework. As a result, an architect should also indicate which version of the framework (s)he used when reporting the vulnerability assessment of a system. We plan to maintain a timestamped database that keeps track of the known threat models that allows to distinguish different versions of the framework.

The second challenge is the determination of vulnerability factor (VF) for each microarchitectural component and a threat model. Conceptually, VF is a metric that allows to measure the correlation between an operation performed on a particular microarchitectural component and the observation made by a specific threat model targeting the given component. However, a specific implementation may be needed to calculate the correlation on each component and a threat model. Identification of specific VFs is left as a future work.

References

[1] K. Baddam and M. Zwolinski, "Evaluation of dynamic voltage and frequency scaling as a differential power analysis countermeasure," in *Proceeding of the International Conference on VLSI Design (VLSID)*, pp. 854–862, Jan 2007.

[2] N. D. P. Avirneni and A. K. Somani, "Countering power analysis attacks using reliable and aggressive designs," *IEEE Transactions on Computers*, vol. 63, pp. 1408–1420, June 2014.

[3] M. Alioto, L. Giancane, G. Scotti, and A. Trifiletti, "Leakage Power Analysis Attacks: A Novel Class of Attacks to Nanometer Cryptographic Circuits," *IEEE Transactions on Circuits and Systems I*, vol. 57, pp. 355–367, Feb. 2010.

[4] J. L. Henning, "Spec cpu2006 benchmark descriptions," *SIGARCH Computer Architure News*, vol. 34, September 2006.

[5] M. Rostami, F. Koushanfar, and R. Karri, "A Primer on Hardware Security: Models, Methods, and Metrics," *Proceedings of the IEEE*, vol. 102, pp. 1283–1295, Aug. 2014.

[6] M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 819–823, Nov. 2013.

[7] J. Demme, R. Martin, A. Waksman, and S. Sethumadhavan, "Side-channel vulnerability factor: A metric for measuring information leakage," in *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, pp. 106–117, June 2012.

[8] T. Zhang, F. Liu, S. Chen, and R. B. Lee, "Side Channel Vulnerability Metrics: The Promise and the Pitfalls," in *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, pp. 1–8, ACM, 2013.