Abstract

Over the years, microprocessor research has largely been focused on improving performance and efficiency. These performance gains have been achieved through a number of sophisticated microarchitectural features such as cache memories, speculative and out-of-order execution, branch prediction, data forwarding, etc. Along with these, features like multi-programming and multi-threading were incorporated to achieve a higher throughput and efficient use of hardware resources. Since these features allow multiple processes to share hardware simultaneously, strong isolation guarantees like protection rings and page table access control are provided by the operating system (OS). These guarantees are unfortunately not enforced at the microarchitectural level. Shared resources have become a source of information leakage that could undermine the isolation provided. A plethora of side-channel attacks, presented in the literature, have shown that an adversary can leak secret information by examining the changes made by a victim's execution to the state of shared microarchitectural components. Thus, the very features in the processor that were incorporated to boost performance and throughput have now become a security liability.

The class of side-channel attacks that exploit vulnerabilities in modern processors are termed as *microarchitectural attacks*. In the last decade, the world of microarchitectural attacks has seen profound growth, with novel and unique vulnerabilities being discovered by exploiting various hardware resources and optimization features. While the traditional microarchitectural attacks were mostly focused on exploiting the inherent timing side channels emanating from cache memories and branch predictors, recent advances in this field have expanded the horizon by discovering vulnerabilities in a number of other microarchitectural components. Recent state-of-the-art attacks like Spectre, Meltdown, Spoiler, Zombieload, Rowhammer, etc. have uncovered new vulnerabilities that can undermine the security guarantees of both hardware and software. Most of these attacks are not easily prevented and the vulnerabilities are hard to fix as they require considerable hardware modifications and possibly performance degradation. Moreover, commercially available secure enclave solutions, such as Intel SGX, failed to impress the security community. These recent developments have led to several security-critical questions and open challenges.

In this work, we explore and investigate different security vulnerabilities in modern processors with a penchant for exploiting those vulnerabilities in the context of real-world applications. We investigate known as well as new side channels and also show that side channel exploration can be fully automated. In particular, we show that timing side channel leakages are hard to mitigate and can be found in several avenues in modern processors. First, we exploit the page frame cache (PFC), an important component of the memory allocation subsystem in modern Linux-based Operating Systems (OS), to control the physical memory allocation of the victim process. We leverage this to perform a pin-pointed Rowhammer-based fault attack on OpenSSL AES T-tables and successfully extract the secret key. Additionally, we propose Deep Learning-based countermeasures to detect Rowhammer attacks in modern processors. Secondly, we discover a novel timing channel in the Return Address Stack (RAS), a hardware-based stack in processors used for predicting return addresses, to perform covert channels and leak secret keys from ECC scalar multiplication. Further, we utilize another artefact from the OS, the deadline scheduler, to perform synchronous control of the attacker and victim processes and eventually demonstrate a lattice-based key recovery attack on the OpenSSL ECDSA signature scheme. Thirdly, we highlight that the store-to-load forwarding optimization feature present in all modern-day processors provides an inherent timing channel that can be leveraged to perform side-channel attacks. We show exploits using both correctly forwarded and wrongly forwarded load instructions by craftily polluting the store buffer with attacker-controlled memory locations. Fourth, we show that recent advances in secured cache designs are still vulnerable to sophisticated cache attack strategies. We provide a mathematical framework to determine different parameters to gauge the resilience of randomized-partitioned cache designs. Furthermore, we show that recently proposed pseudo-fully-associative caches, which are touted to be robust against traditional contention-based cache attacks, are vulnerable to a separate class of cache attacks called cache occupancy attacks. Finally, we propose an automated framework, motivated by Genetic Algorithm approaches, to discover novel transient leakage paths in new-generation Intel processors.

For most of the computing history, “security as an afterthought” has been the principle for all the major hardware manufacturers. However, the recent vulnerability discoveries from various research groups, including the works that culminated in this thesis, serve as strong proponents of robust and more serious consideration of security aspects in modern computer architectures.

Keywords: Microarchitectural Attacks, Side Channel, Timing Channel.