RETROSPECTIVE: New Cache Designs for Thwarting Software Cache-Based Side Channel Attacks

Ruby B. Lee[1]
[1]*Princeton University*
[1]rblee@princeton.edu

Zhenghong Wang[2]
[2]*Apple*
[2]zhenghong_wang@apple.com

I. Context

We did the work on this paper around 2005, when the problem of side-channel attacks on the cache was much less understood than it is today in 2023. We were working in the broader context of information leakage due to processor and cache architectures [1], where we identified attacks on hardware features like Simultaneous Multi-Threading (SMT), speculative execution and caches. This was one aspect of our research under a DARPA and NSF CyberTrust grant on "SecureCore for Trustworthy Commodity Computing and Communications."

When some practical software cache-based side-channel attacks were published at that time, e.g., Bernstein [O5][1] and Percival [O6] attacks, there was great concern in the security-sensitive communities. These software attacks are easy to launch, and can leak the secret or private key of cryptographic algorithms, defeating any cryptographic protections provided by the system. Some attacks could even be done remotely. Since these attacks apply to any computer with a cache, they affect billions of computers in the world. So, we focused our information leakage studies on cache side-channel attacks.

The goal of this paper was to introduce the problem of software cache-based side-channel attacks to the computer architecture community, and to show that they could be defeated by designing more secure cache architectures. The defenses that had been proposed at that time were mostly software solutions that had devastating performance impact, or proposals for better crypto algorithms. The only hardware defense proposed was static cache partitioning [O18] which had severe performance degradation.

This was the first paper to show that secure caches could be designed to defeat these software cache side-channel attacks, without sacrificing performance. Since caches are perhaps the most important performance feature in modern computers, our goal was always to provide better security without compromising performance. A theme of the paper was to show that some hardware mechanisms designed initially for performance, can also be used to improve security. We believe that an important (but realizable) goal for computer architects moving forward is to improve both security and performance. This is like a grand challenge, since in general, one has to sacrifice performance for security, or vice versa.

[1][Ox] refers to Reference [x] in the original paper

II. Security-aware cache designs

We proposed two major hardware defense strategies for conflict-based cache side-channel attacks: Partitioning and Randomization. Partitioning separates the cache into different parts, used by the victim and the attacker, so that they cannot interfere with each other. Unfortunately, this prevents sharing the cache, causing cache fragmentation, and significant performance degradation. Randomization allows cache sharing but randomizes the interference due to sharing, so that the attacker gets no useful information about the victim's cache usage. In conflict-based cache side-channel attacks, interference results in cache misses, which enable an attacker to observe victim cache usage and thus leak information when the usage is secret-dependent.

While Partitioning is an obvious defense, this paper is the first to propose a randomized cache, the Random Permutation cache (RPcache). RPcache randomizes the memory-to-cache mapping, so that even if the attacker observes a cache miss on a cacheline shared with the victim, he gets no information about the memory address actually used by the victim. RPcache uses the conventional set-associative cache, but dynamically produces a random permutation of the memory address to cache set mappings. When a cache miss occurs, instead of placing the new line in the set selected for replacement, another set is randomly chosen for the incoming cache line. The memory-to-cache-set mappings are swapped for these two cache sets. This swapping does incur some invalidations of cache lines, which can impact performance. However, our studies showed that there was negligible performance impact compared to the baseline set-associative cache. We also modeled the cache side-channel as a communications channel, and used information theory to help understand why the RPcache randomization provides the receiver (attacker) with no information.

This paper also showed that the Partitioning defense strategy can be improved with the Partition locked cache, PLcache. PLcache performs dynamic, fine-grained partitioning of the cache, resulting in minimum performance degradation, compared to static partitioning methods. Used properly, the security-sensitive cache lines (e.g., the AES

tables in AES encryption) are fetched and locked into the cache before any cryptographic operations are performed. This results in constant-time encryption or decryption, which leak no information about the secret key. Furthermore, the best performance is achieved, since all AES table accesses are cache hits. Partitioning is by cache lines and not by cache sets or ways, thus achieving better performance with fine-grained partitioning when needed. The cache lines must be unlocked when encryption or decryption is completed, so as not to hog up the cache resources. PLcache is an example of how a hardware feature, allowing cache lines to be locked in the cache (to prevent replacement) to improve performance, can also be used to improve security (by preventing an attacker from interfering with the victim).

III. Follow-on work

An important follow-on work to RPcache is Newcache, which provides the same security protection but even better performance [2].

Newcache was awarded a very selective DHS and AFRL grant for technology-transfer to industry [3][4]. This resulted in a chip implementation to verify the feasibility of the novel Newcache design, its size, latency and power consumption, all of which were found to be comparable to a conventional 8-way set-associative cache of the same size (32 kilobytes). We also verified the performance of Newcache under many different scenarios, and its security against conflict-based cache side-channel attacks. Newcache can be a replacement for set-associative caches, achieving both the security and the performance of a fully associative cache with random replacement, at a lower cost.

Since RPcache and Newcache, there has been much follow-on work by the research communities in both security and computer architecture on randomized caches, especially a decade later, after the publication of our paper that side-channel attacks on the last level cache (LLC) are practical [5]. This includes different ways to achieve the randomized memory-to-cache mapping. Some notable examples are the encrypted address caches [6-7] like Ceaser, Ceaser-S, ScatterCache, and different ways to produce a pseudo fully-associative cache for large caches. These all attempt memory-to-cache address randomization, like RPcache and Newcache, but for large Last Level Caches.

A very interesting type of randomized cache, called the Random Fill cache [8], randomizes *what* is brought into the cache and *when*, rather than *where* it is placed in the cache, which is what has been proposed by RPcache and Newcache, and the encrypted-address caches above.

Much work has also focused on the issues with randomization and how they can be overcome. Indeed, the issue of randomization in caches is a very active area of research today.

Another area that has blossomed are newer classes of cache side-channel attacks, in addition to the conflict-based attacks considered in this paper. These include cache side-channel attacks based on reuse (cache hits) [8] rather than conflicts or contention (cache misses). They also include a large number of speculative execution attacks that use cache side-channels to leak illegally accessed information, starting with the Spectre and Meltdown attacks in 2018.

More defenses, both hardware and software, have also been proposed, especially for the cache-based speculative execution attacks [9].

The research area of designing more secure caches to defeat cache-based side-channel attacks is still very active today, more than 15 years after publication of our paper. More attacks and defenses are continuing to be proposed.

We are happy to have helped bring understanding to this serious problem space and hopefully inspired computer architects to design more secure and high-performing caches.

Authors

Ruby Lee is the Forrest G. Hamrick Professor emeritus of Computer and Electrical Engineering at Princeton University. Prior to Princeton, she was Chief Architect at Hewlett Packard in processor architecture, then multimedia architecture, then security architecture. She is a Fellow of IEEE, ACM and the American Academy of Arts and Sciences. Although graduating to emeritus last year, Ruby is still active in research in Computer Architecture, Cyber Security and Deep Learning.

Zhenghong Wang did his PhD at Princeton University, in information leakage mechanisms and defenses in modern processor and cache architectures. After Princeton, he joined nVidia in 2008, working on GPU architectures. In the following years, his interests broadened beyond processor architecture design and he joined Apple in 2015 where he is a system architect, working on a variety of Mac systems.

References

[1] Z. Wang and R. B. Lee, "Covert and Side Channels Due to Processor Architecture," 22nd Annual Computer Security Applications Conference (ACSAC), 2006.

[2] Z. Wang and R. B. Lee, "A novel cache architecture with enhanced performance and security," IEEE/ACM International Symposium on Microarchitecture, 2008.

[3] F. Liu, H. Wu, K. Mai and R. B. Lee, "Newcache: Secure Cache Architecture Thwarting Cache Side-Channel Attacks," IEEE Micro, vol. 36, no. 5, pp. 8-16, 2016.

[4] Lee, R.B., "Using Moving Target Defense for Secure Hardware Design", DHS and AFRL. Final Report FA8750-12-0295, 2016. Approved for public release. www.dtic.mil

[5] F. Liu, Y. Yarom, Q. Ge, G. Heiser and R. B. Lee, "Last-Level Cache Side-Channel Attacks are Practical," IEEE Symposium on Security and Privacy, 2015.

[6] M. K. Qureshi, "CEASER: Mitigating Conflict-Based Cache Attacks via Encrypted-Address and Remapping," IEEE/ACM International Symposium on Microarchitecture, 2018.

[7] M. Werner, T. Unterluggauer, L. Giner, M. Schwarz, D. Gruss and S. Mangard, "ScatterCache: Thwarting cache attacks via cache set randomization", USENIX Security Symposium (Security), 2019.

[8] F. Liu and R. B. Lee, "Random Fill Cache Architecture," IEEE/ACM International Symposium on Microarchitecture, 2014.

[9] G. Hu, Z. He, and R. B. Lee, "SoK: Hardware Defenses Against Speculative Execution Attacks", IEEE International Symposium on Secure and Private Execution Environment Design (SEED), 2021.