Cache Side Channels: State of the Art and Research Opportunities

Yinqian Zhang
Department of Computer Science and Engineering
The Ohio Sate University
yinqian@cse.ohio-state.edu

ABSTRACT

Cache side channels are a type of attack vectors through which an adversary infers secret information of a running program by observing its use of CPU caches or other caching hardware. The study of cache side channels, particularly access-driven cache side channels, is gaining traction among security researchers in recent years. A large volume of papers on cache side-channel attacks or defenses is being published in both security and computer architecture conferences each year. However, due to the diversity of the research goals, methods, and perspectives, it becomes much harder for researchers new to this field to keep track of the frontiers of this research topic. As such, in this tutorial, we will provide a high-level overview of the studies of cache side channels to help other security researchers to comprehend the state of the art of this research area, and to identify research problems that have not been addressed by the community. We also hope to bridge the gaps between the security community and the computer architecture community on this specific research topic by summarizing research papers from both sides.

ACM Reference Format:
Yinqian Zhang. 2017. Cache Side Channels: State of the Art and Research Opportunities. In *Proceedings of CCS '17*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3133956.3136064

1 INTRODUCTION

Cache side channels are a type of attack vectors through which an adversary infers secret information of a running program by observing its use of CPU caches or other caching hardware. Attacks that exfiltrate sensitive data from CPU caches are called "side"-channel attacks, because the cache data cannot be read by the adversary directly, and the leakage is usually from indirect, "side" information. Prior studies have explored three types of cache side channels: time-driven, access-driven and trace-driven. They differ in their threat models. Time-driven attacks assume only the overall execution time of certain operation is observable by the adversary; trace-driven attacks assume the adversary is able to observe the power consumption traces of the execution; and access-driven attacks assume the adversary has logical access to a cache shared with the victim and infers the victim program's execution through its own use of the shared cache.

This tutorial focuses on access-driven cache side-channel attacks. These attacks have been studied in the past 10 years in multiple contexts, most noticeably in desktop computers [1, 2, 20, 21, 21, 23, 40, 42, 44, 49], cloud servers and virtual machines [6, 25, 37, 61, 62, 67, 68], mobile devices [17, 33, 65], browsers [7, 41], and SGX-enabled Intel processors [8, 22, 47], where CPU caches are shared between mutually-distrusting software components. The exploited caches include not only CPU caches (*e.g.*, L1 data caches and per-core L2 unified caches [21, 23, 40, 42, 44, 49], instruction caches [1, 2, 67], and inclusive LLCs [6, 20, 24, 25, 27, 33, 37, 41, 61, 62, 65, 68]), but also hardware components that temporarily store data for speeding up the computation, such as TLBs [23], and Branch Prediction Units [3, 4, 16, 31]. Attacks on different micro-architectures, such as x86 and ARM, also differ because many of the characteristics of the attacks are CPU specific. Even distinct processors from the same vendors, such as Intel, may render cache side-channel attacks different. Targets of these attacks include cryptographic systems (*e.g.*, cryptographic keys), randomness in computer systems (*e.g.*, address space layout), user privacy (*e.g.*, passwords, private activities), *etc.*

Solutions to cache side channels are typically categorized into three types. First, some solutions are built into the victim programs. They are typically called software-based solutions. Prominent approaches include software transformation [10, 11, 18, 39, 46] and vulnerability detection [15, 53, 59]. Second, some solutions requires system software modifications. Hence they are called system-based defenses. Some of these work proposed to partition the shared caches [29, 35, 45, 48, 70], to eliminate fine-grained `rdtsc` instructions [52], to enforce coarse-grained context switches [50], to enforce deterministic execution [5, 32], to inject noise into the side channels [69], or to detect side-channel attacks at runtime using performance counters [12, 63]. Third, some solutions require new hardware designs, thus are called hardware-based defenses. Most prominent ideas are dynamic cache partitioning [14, 30, 43, 55, 57], cache allocation randomization [28, 36, 56, 57], coarse-grained time keeping [38], oblivious memory traces [34], relaxed inclusion caches [26]. These hardware solutions are typically effective, but are limited in that the time window required to have major processor vendors to incorporate them in commercial hardware is very long.

2 PURPOSES OF THE TUTORIAL

The study of cache side channels, particularly access-driven cache side channels, is gaining traction among security researchers in recent years. A large volume of papers on cache side-channel attacks or defenses is being published in both security and computer architecture conferences each year. Although these papers

CCS '17, October 30-November 3, 2017, Dallas, TX, USA

ACM ISBN 978-1-4503-4946-8/17/10.
https://doi.org/10.1145/3133956.3136064

are all related to cache side channels, their research angles vary significantly: Among studies that develop new attacks, some explore new attack vectors (*i.e.*, new hardware resources that are shared and exploitable as side channels), some report new attack methods (*e.g.*, PRIME+ABORT [13], FLUSH+FLUSH [19]), some improve existing attacks under different micro-architectures (*e.g.*, ARM [17, 33, 65] vs. x86), some demonstrate attacks in new environments (*e.g.*, browsers [7, 41], Intel SGX [8, 22, 47]), and others exploit known methods and vectors against new targets (*e.g.*, userspace ASLR or kernel ASLR). Among the defense papers, the security community and the computer architecture community usually take different perspectives. The computer architecture community seeks for hardware solutions, or a combination of hardware and software solutions to address the issues. In contrast, the security community either enhances existing software systems to defeat side-channel attacks, by leveraging virtualization techniques or hardware features that are available on commodity processors, or transforms software applications to mitigate side-channel vulnerabilities. A recently emerging direction, however, is to conduct static or dynamic analysis on the software to detect vulnerabilities.

However, as the size of the literature grows, it becomes much harder to keep track of the frontiers of this research topic. Therefore, researchers who are new to the field will find it even more challenging to comprehend the state of the art of this research area, or to identify research problems that have not been addressed by the community. Therefore, in this tutorial, we will provide a high-level overview of the studies of cache side channels to help researchers to bootstrap their research.

The second purpose of this tutorial is to bridge the gap between the security community and the architecture community in the area of cache side channels. Cache side-channel research is an interdiscipline topic that is of interest in both communities. However, because the research angles from these two groups of researchers are different, cross-domain understanding and references are insufficient. Therefore, this tutorial also aims to serve as a bridge between the two communities and allow security researchers to learn more about the architecture side of the research.

Intended audience: The intended audience of this tutorial are security researchers with zero experience in cache side channels. It is expected that the audience have basic knowledge of computer architecture, operating systems and computer networking.

Duration: We expect the tutorial to last for 1.5 hours, with about 1 hour and 10 minutes for presentation and 20 minutes for questions.

3 SCOPE OF THE TUTORIAL

The tutorial will cover the following topics of cache side channels.

- *Background.* We will discuss the root causes of cache side channels, various attack vectors in computer micro-architectures, and basic methods to exploiting cache side channels for information extraction. We will try to make the description accessible to non-expert audience.
- *State of the arts.* We will guide the audience to review the literature of cache side-channel attacks and defenses. We will provide new perspectives to taxonomize related work in the field and discuss the state of the arts in each research category.
- *Research opportunities.* We will help the audience to identify key research problems in the field that have not been addressed, or inspire the audience to brainstorm opportunities to pursue innovation.

4 A SHORT BIO OF THE LECTURER

Dr. Yinqian Zhang is an assistant professor of the Department of Computer Science and Engineering at The Ohio State University. His research focuses on system security. His publication on side-channel attacks and defenses include those in the context of cloud computing [51, 60, 63, 64, 66–70], smartphones [58, 65], and Intel SGX [9, 54, 59]. Before joining OSU, Dr. Zhang receives his Ph.D. from University of North Carolina at Chapel Hill from Prof. Michael K. Reiter's research group.

Acknowledgement: This tutorial was supported in part by NSF 1566444.

REFERENCES

[1] Onur Aciiçmez. 2007. Yet another MicroArchitectural Attack: exploiting I-Cache. In *ACM workshop on Computer security architecture.*
[2] Onur Aciiçmez, Billy Bob Brumley, and Philipp Grabher. 2010. New results on instruction cache attacks. In *12th international conference on Cryptographic hardware and embedded systems.*
[3] Onur Aciiçmez, Shay Gueron, and Jean-Pierre Seifert. 2007. New branch prediction vulnerabilities in openSSL and necessary software countermeasures. In *11th international conference on Cryptography and coding.*
[4] Onur Aciiçmez, Çetin Kaya Koç, and Jean-Pierre Seifert. 2007. Predicting secret keys via branch prediction. In *7th Cryptographers' track at the RSA conference on Topics in Cryptology.*
[5] Amittai Aviram, Sen Hu, Bryan Ford, and Ramakrishna Gummadi. 2010. Determinating timing channels in compute clouds. In *ACM Workshop on Cloud Computing Security.*
[6] Naomi Benger, Joop van de Pol, Nigel P. Smart, and Yuval Yarom. 2014. "Ooh Aah... Just a Little Bit": A small amount of side channel can go a long way. In *Cryptology ePrint Archive.*
[7] Erik Bosman, Kaveh Razavi, Herbert Bos, and Cristiano Giuffrida. 2016. Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector. In *IEEE Symposium on Security and Privacy.*
[8] Ferdinand Brasser, Urs Müller, Alexandra Dmitrienko, Kari Kostiainen, Srdjan Capkun, and Ahmad-Reza Sadeghi. 2017. Software Grand Exposure: SGX Cache Attacks Are Practical. In *11th USENIX Workshop on Offensive Technologies.*
[9] Sanchuan Chen, Xiaokuan Zhang, Michael Reiter, and Yinqian Zhang. 2017. Detecting Privileged Side-Channel Attacks in Shielded Execution with Déjà Vu. In *12th ACM Symposium on Information, Computer and Communications Security.*
[10] B. Coppens, I. Verbauwhede, K. De Bosschere, and B. De Sutter. 2009. Practical Mitigations for Timing-Based Side-Channel Attacks on Modern x86 Processors. In *30th IEEE Symposium on Security and Privacy.*
[11] S. Crane, A. Homescu, S. Brunthaler, P. Larsen, and M. Franz. 2015. Thwarting cache side-channel attacks through dynamic software diversity. In *ISOC Network and Distributed System Security Symposium.*
[12] John Demme, Matthew Maycock, Jared Schmitz, Adrian Tang, Adam Waksman, Simha Sethumadhavan, and Salvatore Stolfo. 2013. On the Feasibility of Online Malware Detection with Performance Counters. In *ACM Intl. Symp. on Computer Architecture.*
[13] Craig Disselkoen, David Kohlbrenner, Leo Porter, and Dean Tullsen. 2017. Prime+Abort: A Timer-Free High-Precision L3 Cache Attack using Intel TSX. In *26th USENIX Security Symposium.* USENIX Association.
[14] Leonid Domnitser, Aamer Jaleel, Jason Loew, Nael Abu-Ghazaleh, and Dmitry Ponomarev. 2012. Non-monopolizable caches: Low-complexity mitigation of cache side channel attacks. *ACM Trans. Archit. Code Optim.* 8, 4 (Jan. 2012).
[15] Goran Doychev, Dominik Feld, Boris Köpf, Laurent Mauborgne, and Jan Reineke. 2013. CacheAudit: A tool for the static analysis of cache side channels. In *22st USENIX Security Symposium.*
[16] Dmitry Evtyushkin, Dmitry Ponomarev, and Nael Abu-Ghazaleh. 2016. Jump over ASLR: Attacking branch predictors to bypass ASLR. In *49th Annual IEEE/ACM International Symposium on Microarchitecture.*
[17] Marc Green, Leandro Rodrigues-Lima, Andreas Zankl, Gorka Irazoqui, Johann Heyszl, and Thomas Eisenbarth. 2017. AutoLock: Why Cache Attacks on ARM Are Harder Than You Think. In *26th USENIX Security Symposium.*

[18] Daniel Gruss, Julian Lettner, Felix Schuster, Olya Ohrimenko, Istvan Haller, and Manuel Costa. 2017. Strong and Efficient Cache Side-Channel Protection using Hardware Transactional Memory. In *26th USENIX Security Symposium*.

[19] Daniel Gruss, Clémentine Maurice, Klaus Wagner, and Stefan Mangard. 2016. *Flush+Flush: A Fast and Stealthy Cache Attack*. Springer International Publishing.

[20] Daniel Gruss, Raphael Spreitzer, and Stefan Mangard. 2015. Cache Template Attacks: Automating Attacks on Inclusive Last-level Caches. In *24th USENIX Conference on Security Symposium*. USENIX Association.

[21] David Gullasch, Endre Bangerter, and Stephan Krenn. 2011. Cache games – Bringing access-based cache attacks on AES to practice. In *32nd IEEE Symposium on Security and Privacy*.

[22] Marcus Hähnel, Weidong Cui, and Marcus Peinado. 2017. High-Resolution Side Channels for Untrusted Operating Systems. In *2017 USENIX Annual Technical Conference*. USENIX Association.

[23] Ralf Hund, Carsten Willems, and Thorsten Holz. 2013. Practical Timing Side Channel Attacks Against Kernel Space ASLR. In *34th IEEE Symposium on Security and Privacy*.

[24] Mehmet Sinan Inci, Berk Gulmezoglu, Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar. 2015. Seriously, get off my cloud! Cross-VM RSA Key Recovery in a Public Cloud. Cryptology ePrint Archive. (2015).

[25] Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar. 2015. S$A: A Shared Cache Attack that Works Across Cores and Defies VM Sandboxing—and its Application to AES. In *IEEE Symposium on Security and Privacy*.

[26] Mehmet Kayaalp et al. 2017. RIC: Relaxed Inclusion Caches for Mitigating LLC Side-Channel Attacks. In *Design Automation Conference*.

[27] Mehmet Kayaalp, Nael Abu-Ghazaleh, Dmitry Ponomarev, and Aamer Jaleel. 2016. A High-resolution Side-channel Attack on Last-level Cache. In *53rd Annual Design Automation Conference*.

[28] G. Keramidas, A. Antonopoulos, D.N. Serpanos, and S. Kaxiras. 2008. Non deterministic caches: a simple and effective defense against side channel attacks. *Design Automation for Embedded Systems* 12 (2008), 221–230. Issue 3.

[29] T. Kim, M. Peinado, and G. Mainar-Ruiz. 2012. STEALTHMEM: system-level protection against cache-based side channel attacks in the cloud. In *21st USENIX Security Symposium*.

[30] Jingfei Kong, Onur Aciicmez, Jean-Pierre Seifert, and Huiyang Zhou. 2013. Architecting Against Software Cache-Based Side-Channel Attacks. *IEEE Trans. Comput.* 62, 7 (July 2013).

[31] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. 2017. Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing. In *26th USENIX Security Symposium*. USENIX Association.

[32] Peng Li, Debin Gao, and Michael K. Reiter. 2014. StopWatch: A Cloud Architecture for Timing Channel Mitigation. *ACM Trans. Inf. Syst. Secur.* 17, 2 (Nov. 2014).

[33] Moritz Lipp, Daniel Gruss, Raphael Spreitzer, Clémentine Maurice, and Stefan Mangard. 2016. ARMageddon: Cache Attacks on Mobile Devices. In *USENIX Security Symposium*.

[34] Chang Liu, Austin Harris, Martin Maas, Michael Hicks, Mohit Tiwari, and Elaine Shi. 2015. GhostRider: A Hardware-Software System for Memory Trace Oblivious Computation. In *20th International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM.

[35] Fangfei Liu, Qian Ge, Yuval Yarom, Frank Mckeen, Carlos Rozas, Gernot Heiser, and Ruby B. Lee. 2016. CATalyst: Defeating Last-Level Cache Side Channel Attacks in Cloud Computing. In *22nd IEEE Symposium on High Performance Computer Architecture*.

[36] Fangfei Liu and Ruby B. Lee. 2014. Random Fill Cache Architecture. In *47th IEEE/ACM Symposium on Microarchitecture*.

[37] Fangfei Liu, Yuval Yarom, Qian Ge, Gernot Heiser, and Ruby B. Lee. 2015. Last-Level Cache Side-Channel Attacks are Practical. In *IEEE Symposium on Security and Privacy*.

[38] Robert Martin, John Demme, and Simha Sethumadhavan. 2012. TimeWarp: rethinking timekeeping and performance monitoring mechanisms to mitigate side-channel attacks. In *39th Annual International Symposium on Computer Architecture*.

[39] David Molnar, Matt Piotrowski, David Schultz, and David Wagner. 2005. The program counter security model: automatic detection and removal of control-flow side channel attacks. In *8th international conference on Information Security and Cryptology*.

[40] Michael Neve and Jean-Pierre Seifert. 2007. Advances on access-driven cache attacks on AES. In *13th international conference on Selected areas in cryptography*.

[41] Yossef Oren, Vasileios P. Kemerlis, Simha Sethumadhavan, and Angelos D. Keromytis. 2015. The Spy in the Sandbox: Practical Cache Attacks in JavaScript and Their Implications. In *22nd ACM SIGSAC Conference on Computer and Communications Security*.

[42] Dag Arne Osvik, Adi Shamir, and Eran Tromer. 2006. Cache attacks and countermeasures: the case of AES. In *6th Cryptographers' track at the RSA conference on Topics in Cryptology*.

[43] D. Page. 2005. Partitioned Cache Architecture as a Side-Channel Defence Mechanism. (2005). http://eprint.iacr.org/2005/280

[44] Colin Percival. 2005. Cache missing for fun and profit. In *2005 BSDCan*.

[45] Himanshu Raj, Ripal Nathuji, Abhishek Singh, and Paul England. 2009. Resource Management for Isolation Enhanced Cloud Services. In *ACM Cloud Computing Security Workshop*.

[46] Ashay Rane, Calvin Lin, and Mohit Tiwari. 2015. Raccoon: Closing Digital Side-Channels through Obfuscated Execution. In *24th USENIX Security Symposium*. USENIX Association.

[47] Michael Schwarz, Samuel Weiser, Daniel Gruss, Clémentine Maurice, and Stefan Mangard. 2017. *Malware Guard Extension: Using SGX to Conceal Cache Attacks*. Springer International Publishing.

[48] Jicheng Shi, Xiang Song, Haibo Chen, and Binyu Zang. 2011. Limiting cache-based side-channel in multi-tenant cloud using dynamic page coloring. In *41st International Conference on Dependable Systems and Networks Workshops*.

[49] Eran Tromer, Dag Arne Osvik, and Adi Shamir. 2010. Efficient Cache Attacks on AES, and Countermeasures. *J. Cryptol.* 23, 2 (Jan. 2010), 37–71.

[50] Venkatanathan Varadarajan, Thomas Ristenpart, and Michael Swift. 2014. Scheduler-based Defenses against Cross-VM Side-channels. In *23th USENIX Security Symposium*.

[51] Venkatanathan Varadarajan, Yinqian Zhang, Thomas Ristenpart, and Michael Swift. 2015. A Placement Vulnerability Study in Multi-Tenant Public Clouds. In *USENIX Security Symposium*.

[52] Bhanu C. Vattikonda, Sambit Das, and Hovav Shacham. 2011. Eliminating fine grained timers in Xen. In *3rd ACM Workshop on Cloud Computing Security*.

[53] Shuai Wang, Pei Wang, Xiao Liu, Danfeng Zhang, and Dinghao Wu. 2017. CacheD: Identifying Cache-Based Timing Channels in Production Software. In *26th USENIX Security Symposium*. USENIX Association.

[54] Wenhao Wang, Guoxing Chen, Xiaorui Pan, Yinqian Zhang, XiaoFeng Wang, Vincent Bindschaedler, Haixu Tang, and Carl A. Gunter. 2017. Leaky Cauldron on the Dark Land: Understanding Memory Side-Channel Hazards in SGX. In *ACM Conference on Computer and Communications Security*.

[55] Zhenghong Wang and Ruby B. Lee. 2006. Covert and Side Channels Due to Processor Architecture. In *22nd Annual Computer Security Applications Conference*.

[56] Zhenghong Wang and Ruby B. Lee. 2007. New cache designs for thwarting software cache-based side channel attacks. In *34th annual international symposium on Computer architecture*.

[57] Zhenghong Wang and Ruby B. Lee. 2008. A novel cache architecture with enhanced performance and security. In *41st annual IEEE/ACM International Symposium on Microarchitecture*.

[58] Qiuyu Xiao, Michael K. Reiter, and Yinqian Zhang. 2015. Mitigating storage side channels using statistical privacy mechanisms. In *22nd ACM Conference on Computer and Communications Security*.

[59] Yuan Xiao, Mengyuan Li, Sanchuan Chen, and Yinqian Zhang. 2017. Stacco: Differentially Analyzing Side-Channel Traces for Detecting SSL/TLS Vulnerabilities in Secure Enclaves. In *ACM Conference on Computer and Communications Security*.

[60] Yuan Xiao, Xiaokuan Zhang, Yinqian Zhang, and Radu Teodorescu. 2016. One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation. In *USENIX Security Symposium*.

[61] Yuval Yarom and Naomi Benger. 2014. Recovering OpenSSL ECDSA Nonces Using the FLUSH+RELOAD Cache Side-channel Attack. In *Cryptology ePrint Archive*.

[62] Yuval Yarom and Katrina Falkner. 2014. FLUSH+RELOAD: A high resolution, low noise, L3 cache side-channel attack. In *23rd USENIX Security Symposium*. USENIX Association.

[63] Tianwei Zhang, Yinqian Zhang, and Ruby B. Lee. 2016. CloudRadar: A Real-Time Side-Channel Attack Detection System in Clouds. In *19th International Symposium on Research in Attacks, Intrusions and Defenses*.

[64] Tianwei Zhang, Yinqian Zhang, and Ruby B. Lee. 2017. DoS Attacks on Your Memory in Cloud. In *12th ACM on Asia Conference on Computer and Communications Security*. ACM, 253–265.

[65] Xiaokuan Zhang, Yuan Xiao, and Yinqian Zhang. 2016. Return-Oriented Flush-Reload Side Channels on ARM and Their Implications for Android Devices. In *ACM Conference on Computer and Communications Security*.

[66] Yinqian Zhang, Ari Juels, Alina Oprea, and Michael K. Reiter. 2011. HomeAlone: Co-residency Detection in the Cloud via Side-Channel Analysis. In *IEEE Symposium on Security and Privacy*.

[67] Yinqian Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2012. Cross-VM Side Channels and Their Use to Extract Private Keys. In *ACM Conference on Computer and Communications Security*.

[68] Yinqian Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2014. Cross-Tenant Side-Channel attacks in PaaS clouds. In *ACM Conference on Computer and Communications Security*.

[69] Yinqian Zhang and Michael K. Reiter. 2013. Düppel: retrofitting commodity operating systems to mitigate cache side channels in the cloud. In *ACM Conference on Computer and Communications Security*. 827–838.

[70] Ziqiao Zhou, Michael K. Reiter, and Yinqian Zhang. 2016. A Software Approach to Defeating Side Channels in Last-Level Caches. In *23rd ACM Conference on Computer and Communications Security*.