RowHammer and Beyond

Onur Mutlu

ETH Zürich and Carnegie Mellon University
onur.mutlu@inf.ethz.ch

Abstract. We will discuss the RowHammer problem in DRAM, which is a prime (and likely the first) example of how a circuit-level failure mechanism in Dynamic Random Access Memory (DRAM) can cause a practical and widespread system security vulnerability. RowHammer is the phenomenon that repeatedly accessing a row in a modern DRAM chip predictably causes errors in physically-adjacent rows. It is caused by a hardware failure mechanism called read disturb errors. Building on our initial fundamental work that appeared at ISCA 2014, Google Project Zero demonstrated that this hardware phenomenon can be exploited by user-level programs to gain kernel privileges. Many other recent works demonstrated other attacks exploiting RowHammer, including remote takeover of a server vulnerable to RowHammer. We will analyze the root causes of the problem and examine solution directions. We will also discuss what other problems may be lurking in DRAM and other types of memories, e.g., NAND flash and Phase Change Memory, which can potentially threaten the foundations of reliable and secure systems, as the memory technologies scale to higher densities.

1 Summary

As memory scales down to smaller technology nodes, new failure mechanisms emerge that threaten its correct operation [79, 80]. If such failures are not anticipated and corrected, they can not only degrade system reliability and availability but also, even more importantly, open up new security vulnerabilities: a malicious attacker can exploit the exposed failure mechanism to take over an entire system. As such, new failure mechanisms in memory can become practical and significant threats to system security.

In this keynote talk, based on our ISCA 2014 paper [55], we introduce the RowHammer problem in DRAM, which is a prime (and likely the first) example of a real circuit-level failure mechanism that causes a practical and widespread system security vulnerability. RowHammer, as it is now popularly referred to, is the phenomenon that repeatedly accessing a row in a modern DRAM chip causes bit flips in physically-adjacent rows at consistently predictable bit locations. It is caused by a hardware failure mechanism called *DRAM disturbance errors*, which is a manifestation of circuit-level cell-to-cell interference in a scaled memory technology. Specifically, when a DRAM row is opened (i.e., activated) and closed (i.e., precharged) repeatedly (i.e., *hammered*), enough times within a DRAM refresh interval, one or more bits in physically-adjacent DRAM rows can be flipped to the wrong value. Using an FPGA-based DRAM testing infrastructure [70, 42], we tested 129 DRAM modules manufactured by three major manufacturers in seven recent years (2008–2014) and found that 110 of them exhibited RowHammer errors, the earliest of which dates back to 2010. Our ISCA 2014 paper [55] provides a detailed and rigorous analysis of various characteristics of RowHammer, including its data pattern dependence, repeatability of errors, relationship with leaky cells, and various circuit-level causes of the phenomenon.

We demonstrate that a very simple user-level program [55, 3] can reliably and consistently induce RowHammer errors in commodity AMD and Intel systems using vulnerable DRAM modules. We released the source code of this program [3], which Google Project Zero later enhanced [4]. Using our user-level RowHammer program, we showed that both read and write accesses to memory can induce bit flips, all of which occur in rows other than the one that is being accessed. Since different DRAM rows are mapped to different software pages, our user-level program could reliably corrupt specific bits in pages belonging to other programs. As a result, RowHammer errors can be exploited by a malicious program to breach memory protection and compromise the system. In fact, we hypothesized, in our ISCA 2014 paper, that our user-level program, with some engineering effort, could be developed into a *disturbance attack* that injects errors into other programs, crashes the system, or hijacks control of the system.

RowHammer exposes a *security threat* since it leads to a serious breach of memory isolation: an access to one memory row (e.g., an OS page) predictably modifies the data stored in another row (e.g., another OS page). Malicious software, which we call *disturbance attacks* [55], or *RowHammer attacks*, can be written to take advantage of these disturbance errors to take over an entire system. Inspired by our ISCA 2014 paper's fundamental findings, researchers from Google Project Zero demonstrated in 2015 that RowHammer can be effectively exploited by user-level programs to gain kernel privileges on real systems [94, 95]. Tens of other works since then demonstrated other attacks exploiting RowHammer. These include remote takeover of a server vulnerable to RowHammer via JavaScript code execution [40], takeover of a victim virtual machine by another virtual machine running on the same system [92], takeover of a mobile device by a malicious user-level application that requires no permissions [103], takeover of a mobile system by triggering RowHammer using the WebGL interface on a mobile GPU [35], takeover of a remote system by triggering RowHammer through the Remote Direct Memory Access (RDMA) protocol [101, 67], and various other attacks (e.g., [108, 14, 39, 87, 13, 45, 86, 8, 102, 85]). Thus, RowHammer has widespread and profound real implications on system security, as it destroys memory isolation on top of which modern system security principles are built.

We provide a wide variety of solutions, both *immediate* and *longer-term*, to RowHammer, starting from our ISCA 2014 paper [55]. A popular *immediate* solution we describe and analyze, is to increase the refresh rate of memory such that the probability of inducing a RowHammer error before DRAM cells get refreshed is reduced. Several major system manufacturers have adopted this solution and released security patches that increased DRAM refresh rates (e.g., [11, 43, 66, 34]) in memory controllers deployed in the field. While this solution is practical and effective in reducing the vulnerability, assuming the refresh rate is increased enough to avoid the vulnerability, it has the significant drawbacks of increasing energy/power consumption, reducing system performance, and degrading quality of service experienced by user programs. Our paper shows that the refresh rate needs to be increased by 7X if we want to eliminate *every single* RowHammer-induced error we saw in our tests of 129 DRAM modules. Since DRAM refresh is already a significant burden [69, 70, 31, 46, 47, 89, 49, 84, 33] on energy, performance, and QoS, increasing it by any significant amount would only exacerbate the problem. Yet, increased refresh rate is likely the most practical *immediate* solution to RowHammer that can protect vulnerable chips that are already deployed in the field.

After describing and analyzing six solutions to RowHammer, our ISCA 2014 paper shows that the long-term solution to RowHammer can actually be simple and low cost. We introduce a new idea, called *PARA (Probabilistic Adjacent Row Activation)*: when the memory controller closes a row (after it was activated), with a very low probability, it refreshes the adjacent rows. The probability value is a parameter determined by the system designer or provided programmatically, if needed, to trade off between performance overhead and vulnerability protection guarantees. We show that this solution is very effective: it eliminates the RowHammer vulnerability, providing much higher reliability guarantees than modern hard disks provide today, while requiring no storage cost and having negligible performance and energy overheads [55]. Variants of this solution are currently being adopted in DRAM chips and memory controllers [5, 6].

The RowHammer problem leads to a new mindset that has enabled a renewed interest in hardware security research: real memory chips are vulnerable, in a simple and widespread manner, and this causes real security problems. We believe the RowHammer problem will worsen over time since DRAM cells are getting closer to each other with technology scaling. Other similar vulnerabilities may also be lurking in DRAM and other types of memories, e.g., NAND flash memory or Phase Change Memory, that can potentially threaten the foundations of secure systems [80]. Our work advocates a principled system-memory co-design approach to memory reliability and security research that can enable us to better anticipate and prevent such vulnerabilities.

2 Significance, Impact and the Future

RowHammer has spurred significant amount of research and industry attention since its publication in 2014. Our ISCA 2014 paper [55] is the first to experimentally and scientifically demonstrate the RowHammer vulnerability, its characteristics, and its prevalence in real DRAM chips. RowHammer is a prime (and likely the first) example of a hardware failure mechanism that causes a practical and widespread system security vulnerability. Thus, the implications of RowHammer and our ISCA 2014 paper on systems security is tremendous, both in the short term and the long term: it is the first work we know of that shows that a real reliability problem in one of the ubiquitous general-purpose hardware components (DRAM chips) can cause practical and widespread system security vulnerabilities.

Since its publication in 2014, RowHammer has already had significant real-world impact on both industry and academia in at least four directions. These directions will continue to exert long-term impact for RowHammer, as memory cells continue to get closer to each other while the technology scaling of memory continues.

First, our work has inspired many researchers to exploit RowHammer to devise new attacks. As mentioned earlier, tens of papers were written in top security venues that demonstrate various practical attacks exploiting RowHammer (e.g., [108, 14, 39, 87, 13, 45, 8, 85, 40, 92, 103, 35]). These attacks started with Google Project Zero's first work in 2015 [94, 95] and they continue to this date, with the latest ones that we know of being published in Summer 2018 [86, 67, 101, 102]. We believe there is a lot more to come in this direction: as systems security researchers understand more about RowHammer, and as the RowHammer phenomenon continues to fundamentally affect memory chips due to technology scaling problems [80], researchers and practitioners will develop different types of attacks to exploit RowHammer in various contexts and in many more creative ways. Some recent reports suggest that new-generation DDR4 DRAM chips are vulnerable to RowHammer [58, 85, 8, 10], so the fundamental security research on RowHammer is likely to continue into the future.

Second, due to its prevalence in real DRAM chips, as demonstrated in our ISCA 2014 paper, RowHammer has become a popular phenomenon [105, 1, 2, 41, 58, 95, 83, 9, 37], which, in turn, has helped make hardware security even more "mainstream" in popular media and the broader security community. It showed that hardware reliability problems can be very serious security threats that have to be defended against. A well-read article from the Wired magazine, all about RowHammer, is entitled "Forget Software – Now Hackers are Exploiting Physics!" [38], indicating the shift of mindset towards very low-level hardware security vulnerabilities in the popular mainstream security community. Many other popular articles in press have been written about RowHammer, many of which pointing to the our ISCA 2014 work [55] as the first demonstration and scientific analysis of the RowHammer problem. Showing that hardware reliability problems can be serious security threats and pulling them to the popular discussion space, and thus influencing the mainstream discourse, creates a very long term impact for the RowHammer problem.

Third, our work inspired many solution and mitigation techniques for RowHammer from both researchers and industry practitioners. *Apple* publicly mentioned, in their critical security release for RowHammer, that they increased the memory refresh rates due to the "original research by Yoongu Kim et al. (2014)" [11]. Memtest86 program was updated, including a RowHammer test, acknowledging our ISCA 2014 paper [83]. Many academic works developed solutions to

RowHammer, working from our original research (e.g., [12, 50, 39, 96, 15, 44, 97, 36, 104, 65]). Multiple industrial solutions (e.g., [5, 6]) were inspired by our new solution to RowHammer, Probabilistic Adjacent Row Activation (PARA). We believe such solutions will continue to be generated in both academia and industry, extending RowHammer's impact into the very long term.

Fourth, and perhaps most importantly, RowHammer enabled a shift of mindset among mainstream security researchers: general-purpose hardware is fallible (in a very widespread manner) and its problems are actually exploitable. This shift of mindset enabled many systems security researchers to examine hardware in more depth and understand its inner workings and vulnerabilities better. We believe it is no coincidence that two of the groups that concurrently discovered the Meltdown [68] and Spectre [56] vulnerabilities (Google Project Zero and TU Graz InfoSec) have heavily worked on RowHammer attacks before. We believe this shift in mindset, enabled in good part by the existence and prevalence of RowHammer, will continue to be very be important for discovering and solving other potential vulnerabilities that may appear as a result of both technology scaling and hardware design.

3 Other Potential Vulnerabilities

We believe that, as memory technologies scale to higher densities, other problems may start appearing (or may already be going unnoticed) that can potentially threaten the foundations of secure systems. There have been recent large-scale field studies a well as small-scale controlled studies of real memory errors on real devices and systems, showing that both DRAM and NAND flash memory technologies are becoming less reliable [82, 78, 98–100, 77, 93, 28, 27, 74, 73, 17, 25, 79, 84, 80]. As detailed experimental analyses of real DRAM and NAND flash chips show, both technologies are becoming much more vulnerable to cell-to-cell interference effects [82, 55, 26, 22, 20, 17, 21, 81, 72, 23, 28, 27, 79, 80], data retention is becoming significantly more difficult in both technologies [69, 47, 70, 49, 89, 31, 46, 75, 25, 18, 71, 17, 21, 19, 81, 48, 28, 27, 74, 73, 82, 79], and error variation within and across chips is increasingly prominent [70, 63, 30, 29, 17, 21, 64, 51–53]. Emerging memory technologies [79, 76], such as Phase-Change Memory [59, 111, 88, 90, 106, 91, 61, 60, 110, 109], STT-MRAM [32, 57], and RRAM/ReRAM/memristors [107] are likely to exhibit similar and perhaps even more exacerbated reliability issues. We believe, if not carefully accounted for and corrected, these reliability problems may surface as security problems as well, as in the case of RowHammer, especially if the technology is employed as part of the main memory system that is directly exposed to user-level programs. We believe future work examining these vulnerabilities, among others, is promising for both fixing the vulnerabilities and enabling the effective scaling of memory technology.

Acknowledgments

This short paper and the associated keynote talk are heavily based on two previous papers we have written on RowHammer, one that first scientifically introduced and analyzed the phenomenon in ISCA 2014 [55] and the other that provides an analysis and future outlook on RowHammer [80]. They are a result of the research done together with many students and collaborators over the course of the past 7-8 years. In particular, three PhD theses have shaped the understanding that led to this work. These are Yoongu Kim's thesis entitled "Architectural Techniques to Enhance DRAM Scaling" [54], Yu Cai's thesis entitled "NAND Flash Memory: Characterization, Analysis, Modeling and Mechanisms" [24] and his continued follow-on work after his thesis, summarized in [28, 27], and Donghyuk Lee's thesis entitled "Reducing DRAM Latency at Low Cost by Exploiting Heterogeneity" [62]. We also acknowledge various funding agencies (NSF, SRC, ISTC, CyLab) and industrial partners (AliBaba, AMD, Google, Facebook, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware) who have supported the presented and other related work in my group generously over the years. The first version of this talk was delivered at a CMU CyLab Partners Conference in September 2015. Another version of the talk was delivered as part of an Invited Session at DAC 2016, with a collaborative accompanying paper entitled "Who Is the Major Threat to Tomorrows Security? You, the Hardware Designer" [16]. The most recent version is the invited talk given at the Top Picks in Hardware and Embedded Security workshop, co-located with ICCAD 2018 [7], where RowHammer was selected as a Top Pick among hardware and embedded security papers published between 2012-2017. I would like to also thank Christina Giannoula for her help in preparing this manuscript.

References

1. RowHammer Discussion Group. https://groups.google.com/forum/#!forum /rowhammer-discuss.
2. RowHammer on Twitter. https://twitter.com/search?q=rowhammer.
3. Rowhammer: Source Code for Testing the Row Hammer Error Mechanism in DRAM Devices. https://github.com/CMU-SAFARI/rowhammer.
4. Test DRAM for Bit Flips Caused by the RowHammer Problem. https://github.com/google/rowhammer-test.
5. ThinkPad X210 BIOS Debugging. https://github.com/tadfisher/x210-bios.
6. Tweet about RowHammer Mitigation on x210. https://twitter.com/isislovecruft/status/1021939922754723841.
7. Top Picks in Hardware and Embedded Security - Workshop Collocated with ICCAD 2018. https://wp.nyu.edu/toppicksinhardwaresecurity/, 2017.
8. Misiker Tadesse Aga, Zelalem Birhanu Aweke, and Todd Austin. When Good Protections go Bad: Exploiting anti-DoS Measures to Accelerate Rowhammer Attacks. In *HOST*, 2017.
9. Barbara Aichinger. The Known Failure Mechanism in DDR3 Memory referred to as Row Hammer.

http://ddrdetective.com/files/6414/1036/5710/The_Known_Failure_Mechanism_in_DDR3_memory_referred_to_as_Row_Hammer.pdf, September 2014.

10. Barbara Aichinger. DDR Memory Errors Caused by Row Hammer. In *HPEC*, 2015.
11. Apple Inc. About the security content of Mac EFI Security Update 2015-001. https://support.apple.com/en-us/HT204934, June 2015.
12. Zelalem Birhanu Aweke et al. Anvil: Software-based protection against next-generation rowhammer attacks. In *ASPLOS*, 2016.
13. Sarani Bhattacharya and Debdeep Mukhopadhyay. Curious Case of RowHammer: Flipping Secret Exponent Bits using Timing Analysis. In *CHES*, 2016.
14. E. Bosman et al. Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector. *S&P*, 2016.
15. Ferdinand Brasser, Lucas Davi, David Gens, Christopher Liebchen, and Ahmad-Reza Sadeghi. Can't Touch This: Practical and Generic Software-only Defenses Against RowHammer Attacks. *USENIX Sec.*, 2017.
16. W. Burleson et al. Who Is the Major Threat to Tomorrow's Security? You, the Hardware Designer. *DAC*, 2016.
17. Y. Cai et al. Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis. In *DATE*, 2012.
18. Y. Cai et al. Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime. In *ICCD*, 2012.
19. Y. Cai et al. Error Analysis and Retention-Aware Error Management for NAND Flash Memory. *ITJ*, 2013.
20. Y. Cai et al. Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation. In *ICCD*, 2013.
21. Y. Cai et al. Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling. In *DATE*, 2013.
22. Y. Cai et al. Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories. In *SIGMETRICS*, 2014.
23. Y. Cai et al. Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques. In *HPCA*, 2017.
24. Yu Cai. *NAND flash memory: Characterization, Analysis, Modeling and Mechanisms*. PhD thesis, Carnegie Mellon University, 2012.
25. Yu Cai et al. Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery. In *HPCA*, 2015.
26. Yu Cai et al. Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery. In *DSN*, 2015.
27. Yu Cai, Saugata Ghose, Erich F Haratsch, Yixin Luo, and Onur Mutlu. Error Characterization, Mitigation, and Recovery in Flash-memory-based Solid-state Drives. *Proceedings of the IEEE*, 2017.
28. Yu Cai, Saugata Ghose, Erich F Haratsch, Yixin Luo, and Onur Mutlu. Errors in Flash-Memory-Based Solid-State Drives: Analysis, Mitigation, and Recovery. *arXiv preprint arXiv:1711.11427*, 2017.
29. Karthik Chandrasekar et al. Exploiting Expendable Process-margins in DRAMs for Run-time Performance Optimization. In *DATE*, 2014.
30. K. Chang et al. Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization. *SIGMETRICS*, 2016.
31. Kevin Chang et al. Improving DRAM Performance by Parallelizing Refreshes with Accesses. In *HPCA*, 2014.
32. E. Chen et al. Advances and Future Prospects of Spin-Transfer Torque Random Access Memory. *IEEE Transactions on Magnetics*, 2010.

33. Anup Das et al. VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency. In *DAC*, 2018.
34. Troy Fridley and Omar Santos. Mitigations Available for the DRAM Row Hammer Vulnerability. http://blogs.cisco.com/security/mitigations-available-for-the-dram-row-hammer-vulnerability, March 2015.
35. P. Frigo et al. Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU. *IEEE S&P*, 2018.
36. Hector Gomez, Andres Amaya, and Elkim Roa. DRAM Row-hammer Attack Reduction using Dummy Cells. In *NORCAS*, 2016.
37. Dan Goodin. Once thought safe, DDR4 memory shown to be vulnerable to Rowhammer. https://arstechnica.com/information-technology/2016/03/once-thought-safe-ddr4-memory-shown-to-be-vulnerable-to-rowhammer/, 2016.
38. Andy Greenberg. Forget Software – Now Hackers are Exploiting Physics. https://www.wired.com/2016/08/new-form-hacking-breaks-ideas-computers-work/, 2016.
39. D. Gruss et al. Another Flip in the Wall of Rowhammer Defenses. *IEEE S&P*, 2018.
40. Daniel Gruss et al. Rowhammer.js: A remote software-induced fault attack in javascript. *CoRR*, abs/1507.06955, 2015.
41. Robin Harris. Flipping DRAM bits - maliciously. http://www.zdnet.com/article/flipping-dram-bits-maliciously/, December 2014.
42. Hasan Hassan et al. SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies. In *HPCA*, 2017.
43. Hewlett-Packard Enterprise. HP Moonshot Component Pack Version 2015.05.0. http://h17007.www1.hp.com/us/en/enterprise/servers/products/moonshot/component-pack/index.aspx, 2015.
44. Gorka Irazoqui, Thomas Eisenbarth, and Berk Sunar. MASCAT: Stopping Microarchitectural Attacks Before Execution. *IACR Cryptology ePrint Archive*, 2016.
45. Yeongjin Jang, Jaehyuk Lee, Sangho Lee, and Taesoo Kim. SGX-Bomb: Locking Down the Processor via Rowhammer Attack. In *SysTEX*, 2017.
46. Uksong Kang et al. Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling. In *The Memory Forum*, 2014.
47. Samira Khan et al. The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study. *SIGMETRICS*, 2014.
48. Samira Khan et al. A Case for Memory Content-Based Detection and Mitigation of Data-Dependent Failures in DRAM. *CAL*, 2016.
49. Samira Khan et al. PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM. In *DSN*, 2016.
50. Dae-Hyun Kim et al. Architectural Support for Mitigating Row Hammering in DRAM Memories. *IEEE CAL*, 2015.
51. Jeremie S Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu. Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines. In *ICCD*, 2018.
52. Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu. The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices. In *HPCA*, 2018.

53. Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu. D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput. In *HPCA*, 2019.
54. Yoongu Kim. *Architectural Techniques to Enhance DRAM Scaling.* PhD thesis, Carnegie Mellon University, 2015.
55. Yoongu Kim et al. Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors. In *ISCA*, 2014.
56. Paul Kocher, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre Attacks: Exploiting Speculative Execution. *S&P*, 2018.
57. E. Kultursay et al. Evaluating STT-RAM as an energy-efficient main memory alternative. In *ISPASS*, 2013.
58. Mark Lanteigne. How Rowhammer Could Be Used to Exploit Weaknesses in Computer Hardware. http://www.thirdio.com/rowhammer.pdf, March 2016.
59. B. C. Lee et al. Architecting Phase Change Memory as a Scalable DRAM Alternative. In *ISCA*, 2009.
60. B. C. Lee et al. Phase Change Memory Architecture and the Quest for Scalability. *CACM*, 2010.
61. Benjamin C. Lee et al. Phase Change Technology and the Future of Main Memory. *IEEE Micro*, 2010.
62. D. Lee. Reducing DRAM Latency by Exploiting Heterogeneity. *ArXiV*, 2016.
63. D. Lee et al. Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case. In *HPCA*, 2015.
64. Donghyuk Lee, Samira Khan, Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Gennady Pekhimenko, Vivek Seshadri, and Onur Mutlu. Design-induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms. *POMACS*, 2017.
65. Eojin Lee, Sukhan Lee, G Edward Suh, and Jung Ho Ahn. TWiCe: Time Window Counter Based Row Refresh to Prevent Row-Hammering. *CAL*, 2018.
66. Lenovo. Row Hammer Privilege Escalation. https://support.lenovo.com/us/en/product_security/row_hammer, March 2015.
67. M. Lipp et al. Nethammer: Inducing Rowhammer Faults through Network Requests. *arxiv.org*, 2018.
68. Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, et al. Meltdown: Reading Kernel Memory from User Space. In *USENIX Security*, 2018.
69. J. Liu et al. RAIDR: Retention-aware intelligent DRAM refresh. *ISCA*, 2012.
70. J. Liu et al. An experimental study of data retention behavior in modern DRAM devices: Implications for retention time profiling mechanisms. *ISCA*, 2013.
71. Y. Luo et al. WARM: Improving NAND Flash Memory Lifetime with Write-hotness Aware Retention Management. *MSST*, 2015.
72. Yixin Luo et al. Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory. *JSAC*, 2016.
73. Yixin Luo, Saugata Ghose, Yu Cai, Erich F Haratsch, and Onur Mutlu. HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness. In *HPCA*, 2018.
74. Yixin Luo, Saugata Ghose, Yu Cai, Erich F Haratsch, and Onur Mutlu. Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation. *POMACS*, 2018.

75. J. Mandelman et al. Challenges and future directions for the scaling of dynamic random-access memory (DRAM). *IBM Journal of Research and Development*, 46, 2002.
76. J. Meza et al. A Case for Efficient Hardware-Software Cooperative Management of Storage and Memory. In *WEED*, 2013.
77. J. Meza et al. A Large-Scale Study of Flash Memory Errors in the Field. In *SIGMETRICS*, 2015.
78. J. Meza et al. Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field. *DSN*, 2015.
79. O. Mutlu. Memory Scaling: A Systems Architecture Perspective. *IMW*, 2013.
80. O. Mutlu. The RowHammer Problem and Other Issues we may Face as Memory Becomes Denser. *DATE*, 2017.
81. Onur Mutlu. Error Analysis and Management for MLC NAND Flash Memory. In *Flash Memory Summit*, 2014.
82. Onur Mutlu and Lavanya Subramanian. Research problems and opportunities in memory systems. *SUPERFRI*, 2014.
83. PassMark Software. MemTest86: The Original Industry Standard Memory Diagnostic Utility. http://www.memtest86.com/troubleshooting.htm, 2015.
84. Minesh Patel, Jeremie S Kim, and Onur Mutlu. The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions. *ISCA*, 2017.
85. Peter Pessl, Daniel Gruss, Clémentine Maurice, Michael Schwarz, and Stefan Mangard. DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks. In *USENIX Security*, 2016.
86. Damian Poddebniak, Juraj Somorovsky, Sebastian Schinzel, Manfred Lochter, and Paul Rösler. Attacking Deterministic Signature Schemes using Fault Attacks. In *EuroS&P*, 2018.
87. Rui Qiao and Mark Seaborn. A New Approach for Rowhammer Attacks. In *HOST*, 2016.
88. M. K. Qureshi et al. Scalable high performance main memory system using phase-change memory technology. In *ISCA*, 2009.
89. M. K. Qureshi et al. AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems. In *DSN*, 2015.
90. Moinuddin K. Qureshi et al. Enhancing Lifetime and Security of Phase Change Memories via Start-Gap Wear Leveling. In *MICRO*, 2009.
91. S. Raoux et al. Phase-change Random Access Memory: A Scalable Technology. *IBM Journal of Research and Development*, 2008.
92. K. Razavi et al. Flip Feng Shui: Hammering a Needle in the Software Stack. *USENIX Security*, 2016.
93. Bianca Schroeder et al. Flash Reliability in Production: The Expected and the Unexpected. In *USENIX FAST*, 2016.
94. Mark Seaborn and Thomas Dullien. Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges. http://googleprojectzero.blogspot.com.tr/2015/03/exploiting-dram-rowhammer-bug-to-gain.html, 2015.
95. Mark Seaborn and Thomas Dullien. Exploiting the DRAM rowhammer bug to gain kernel privileges. *BlackHat*, 2016.
96. Seyed Mohammad Seyedzadeh, Alex K Jones, and Rami Melhem. Counter-based Tree Structure for Row Hammering Mitigation in DRAM. *CAL*, 2017.
97. Mungyu Son, Hyunsun Park, Junwhan Ahn, and Sungjoo Yoo. Making DRAM Stronger Against Row Hammering. In *DAC*, 2017.

98. Vilas Sridharan, Nathan DeBardeleben, Sean Blanchard, Kurt B. Ferreira, Jon Stearley, John Shalf, and Sudhanva Gurumurthi. Memory Errors in Modern Systems: The Good, The Bad, and The Ugly. In *ASPLOS*, 2015.
99. Vilas Sridharan and Dean Liberty. A Study of DRAM Failures in the Field. In *SC*, 2012.
100. Vilas Sridharan, Jon Stearley, Nathan DeBardeleben, Sean Blanchard, and Sudhanva Gurumurthi. Feng Shui of Supercomputer Memory: Positional Effects in DRAM and SRAM Faults. In *SC*, 2013.
101. A. Tatar et al. Throwhammer: Rowhammer Attacks over the Network and Defenses. *USENIX ATC*, 2018.
102. Andrei Tatar, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi. Defeating Software Mitigations Against Rowhammer: A Surgical Precision Hammer. In *RAID*, 2018.
103. V. van der Veen et al. Drammer: Deterministic Rowhammer Attacks on Mobile Platforms. *CCS*, 2016.
104. Victor van der Veen, Martina Lindorfer, Yanick Fratantonio, Harikrishnan Padmanabha Pillai, Giovanni Vigna, Christopher Kruegel, Herbert Bos, and Kaveh Razavi. GuardION: Practical Mitigation of DMA-Based Rowhammer Attacks on ARM. In *DIMVA*, 2018.
105. Wikipedia. Row hammer. https://en.wikipedia.org/wiki/Row_hammer.
106. H-S. P. Wong et al. Phase Change Memory. *Proceedings of the IEEE*, 2010.
107. H-S. P. Wong et al. Metal-Oxide RRAM. In *Proceedings of the IEEE*, 2012.
108. Y. Xiao et al. One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation. *USENIX Sec.*, 2016.
109. H. Yoon et al. Row Buffer Locality Aware Caching Policies for Hybrid Memories. In *ICCD*, 2012.
110. HanBin Yoon et al. Efficient Data Mapping and Buffering Techniques for Multi-Level Cell Phase-Change Memories. *TACO*, 2014.
111. Ping Zhou et al. A Durable and Energy Efficient Main Memory using Phase Change Memory Technology. In *ISCA*, 2009.