Retrospective: Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Onur Mutlu
ETH Zürich

***Abstract*—Our ISCA 2014 paper [1] provided the first scientific and detailed characterization, analysis, and real-system demonstration of what is now popularly known as the RowHammer phenomenon (or vulnerability) in modern commodity DRAM chips, which are used as main memory in almost all modern computing systems. It experimentally demonstrated that more than 80% of all DRAM modules we tested from the three major DRAM vendors were vulnerable to the RowHammer read disturbance phenomenon: one can predictably induce bitflips (i.e., data corruption) in real DRAM modules by repeatedly accessing a DRAM row and thus causing electrical disturbance to physically nearby rows. We showed that a simple unprivileged user-level program induced RowHammer bitflips in multiple real systems and suggested that a security attack can be built using this proof-of-concept to hijack control of the system or cause other harm. To solve the RowHammer problem, our paper examined seven different approaches (including a novel probabilistic approach that has very low cost), some of which influenced or were adopted in different industrial products.**

Many later works from various research communities examined RowHammer, building real security attacks, proposing new defenses, further analyzing the problem at various (e.g., device/circuit, architecture, and system) levels, and exploiting RowHammer for various purposes (e.g., to reverse-engineer DRAM chips). Industry has worked to mitigate the problem, changing both memory controllers and DRAM standards/chips. Two major DRAM vendors finally wrote papers on the topic in 2023, describing their current approaches to mitigate RowHammer. Research & development on RowHammer in both academia & industry continues to be very active and fascinating.

This short retrospective provides a brief analysis of our ISCA 2014 paper and its impact. We describe the circumstances that led to our paper, mention its influence on later works and products, describe the mindset change we believe it has helped enable in hardware security, and discuss our predictions for future.

I. Background and Circumstances

Our stumbling on the RowHammer problem and creation of its first scientific analysis happened as a result of a confluence of multiple factors. First, my group was working on DRAM technology scaling issues since late 2010. We were very interested in failure mechanisms that appear or worsen due to aggressive technology scaling. To study such issues (e.g., data retention errors [2]), we built an FPGA-based DRAM testing infrastructure [2] between 2011-2012, which we later open sourced as SoftMC [3, 4] and DRAM Bender [5, 6]. Second, around the same timeframe, we were investigating similar technology scaling issues in flash memory using real NAND flash chips [7, 8]. We knew read disturbance errors were significant in NAND flash memory [7–11] and were very interested in how prevalent they were in DRAM. Third, we were collaborating with Intel (e.g., [2]) to understand and solve DRAM technology scaling problems and build our DRAM infrastructure. Three of my students and I spent the summer of 2012 at Intel to work closely with our collaborators (two are co-authors): during this time, we finalized the calibration and stabilization of our infrastructure and had significant technical discussions and experimentation on DRAM scaling problems.

Although there was awareness of the RowHammer problem in industry in 2012 (see Footnote 1 in [1]), there was no comprehensive experimental analysis and detailed real-system demonstration of it. We believed it was critical to provide a rigorous scientific analysis using a wide variety of DRAM chips and scientifically establish major characteristics and prevalence of RowHammer. Hence, in the summer of 2012, we set out to use our DRAM testing infrastructure to analyze RowHammer. Our initial results showed how widespread the read disturbance problem was across the (at the time) recent DRAM chips we tested, so we studied the problem comprehensively and developed many solutions to it. The resulting paper was submitted to MICRO in May 2013 but was rejected. We strengthened the results, especially of the mitigation mechanisms and the number of tested chips, and made the analysis more comprehensive before it was accepted to ISCA 2014 (2 of the 6 reviewers still rejected it for interesting reasons).

II. Major Contribution and Influence

The major contribution of our paper is the exposure and detailed analysis of a fundamental hardware failure mechanism that breaks memory isolation in real systems and thus has huge implications on system reliability, security, and safety. Our paper is a comprehensive study of a major DRAM technology scaling problem, RowHammer, including its first scientific analysis, experimental characterization, real system demonstration, and solutions with their evaluation. To our knowledge, RowHammer is the first example of a hardware failure mechanism that creates a significant and widespread system security vulnerability [12–15], as our ISCA 2014 paper suggested.

Our work has had large influence on both industry & academia. Individual follow-on works are many to list here; we refer the reader to longer invited retrospectives we wrote [12–14]. We give major examples of influence, focusing on RowHammer's effect on the collective mindset of security research and major industry milestones related to RowHammer.

RowHammer Attacks & Mindset Shift in Hardware Security. Our demonstration that one can easily and predictably induce bitflips in commodity DRAM chips using a real user-level program enabled a major mindset shift in hardware security. It showed that general-purpose hardware is fallible in a very widespread manner and its problems are exploitable. Tens of works (see [13, 14]) built directly on our work to exploit RowHammer bitflips to develop many attacks that compromise system integrity and confidentiality, starting from the first RowHammer exploit by Google Project Zero in 2015 [16, 17] to recent works in 2022-2023 (e.g., [18, 19]). These attacks showed increasingly sophisticated ways by which an unprivileged attacker can exploit RowHammer bitflips to circumvent memory protection and gain complete control of a system (e.g., [16, 20–28]), gain access to confidential data (e.g., [18, 19, 29]), or maliciously destroy the safety and accuracy of a system, e.g., an otherwise accurate machine learning inference engine (e.g., [30, 31]). The mindset enabled by RowHammer bitflips caused a renewed interest in hardware security research, enticing many researchers to deeply understand hardware's inner workings and find new vulnerabilities. Thus, hardware security issues have become mainstream discussion in top security & architecture venues, some having sessions entitled RowHammer.

RowHammer Defenses. Tens of works proposed mitigations against RowHammer, some of which were inspired by the solutions we discussed in our ISCA 2014 paper. To date, the search for more efficient and low-cost RowHammer solutions continues. We refer the reader to our prior overview papers [13, 14, 32] and more recent works in 2023 (e.g., [33–35]).

RowHammer Analyses. Our paper initiated works at both architectural & circuit/device-levels to better understand RowHammer and reverse-engineer DRAM chips, to develop better models, defenses, and attacks (see [13, 14]). Our ISCA'20 work [36] revisited RowHammer, comprehensively analyzed of 1580 DRAM chips of three different types from at least two generations, showing that RowHammer has gotten much worse with technology scaling & existing solutions are not effective at future vulnerability levels.

Industry Reaction: Attacks, Analyses, and Mitigations. Folks developing industrial memory testing programs immediately included RowHammer tests, e.g., in memtest86 [37], citing our work. Industry needed to immediately protect RowHammer-vulnerable chips already in the field, so almost all system vendors increased refresh rates; a solution we examined in our paper and deemed costly for performance and energy, yet it was the only practical lever that could be used in the field. Apple publicly acknowledged our work in their security release [38] that announced higher refresh rates to mitigate RowHammer. Intel designed memory controllers that performed probabilistic activations (i.e., pTRR [39, 40]), similar to our PARA solution [1]. DRAM vendors modified the DRAM standard to introduce TRR (target row refresh) mechanisms [39] and claimed their new DDR4 chips to be RowHammer-free [39, 41]. This bold claim was later refuted by our TRRespass work [39] in 2020, which introduced the many-sided RowHammer attack to circumvent internal protection mechanisms added to the DRAM chips. Our later work, Uncovering TRR [41] showed that one can almost completely reverse-engineer and thus easily bypass RowHammer mitigations employed in all tested DRAM chips, i.e., RowHammer solutions in DRAM chips are broken. The analysis done by our two major works in 2020 [36, 39] caused the industry to reorganize the RowHammer task group at JEDEC, which produced two white papers on mitigating RowHammer [42, 43]. Nine years after our paper, in 2023, two major DRAM vendors, SK Hynix and Samsung, finally wrote papers [44, 45] on the RowHammer problem, describing their solutions. Several of these industry solutions build on the probabilistic & access-counter-based solution approaches our ISCA 2014 paper introduced.

Major Internet and cloud systems companies also took a deep interest in RowHammer as it can greatly impact their system security, dependability, and availability. Multiple works from Google, e.g., by Google Project Zero in 2015 [16, 17] and Half Double in 2021-2022 [46] directly built on our paper to demonstrate attacks in real systems. Researchers from Microsoft have developed deeper analyses of RowHammer [47], along with new RowHammer attacks [48] and defenses (e.g., [48–51]).

III. Summary and Future Outlook

Since 2012-2014, RowHammer vulnerability has become much worse due to technology scaling: without mitigation, one can now induce RowHammer bitflips with orders of magnitude smaller number of activations (e.g., ∼10K) and cause much higher rates of errors in cutting-edge DRAM chips [36, 41]. Sophisticated attacks are continuously developed to circumvent the mitigations employed in real DRAM chips. Fortunately, we have also come a long way in further understanding and better mitigating the RowHammer vulnerability. The industry is now (hopefully) fully aware of the importance of the problem and of avoiding bitlips. Unfortunately, an efficient and completely-secure solution is not found yet. The solution space poses a rich area of tradeoffs in terms of security, performance, power/energy, cost/complexity. All solutions forego some desirable properties in favor of others. As such, a critical direction for the future is to find solutions superior to what we have today. We believe system-DRAM cooperation [14, 52] will be important to enabling complete solutions. We also believe it is critical to deeply understand the properties of RowHammer under many different conditions so that we can develop effective solutions that work under all circumstances. Unfortunately, we do not yet fully understand many facets of RowHammer (see [14, 53–55]).

DRAM technology scaling will continue to create problems that will exacerbate the bitflips and the resulting robustness (i.e., safety/security/reliability) problems. Our ISCA 2023 paper on RowPress [55] provides the first scientific and detailed characterization, analysis, and real-system demonstration of yet another read disturbance mechanism in DRAM. What other fascinating problems will we see and can we completely solve them efficiently? Will we ever be free of bitflips at the system and application levels?

References

[1] Y. Kim *et al.*, “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors,” in *ISCA*, 2014.

[2] J. Liu *et al.*, “An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms,” *ISCA*, 2013.

[3] H. Hassan *et al.*, “SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies,” in *HPCA*, 2017.

[4] SoftMC Source Code, https://github.com/CMU-SAFARI/SoftMC.

[5] A. Olgun *et al.*, “DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips,” *TCAD*, 2023.

[6] “DRAM Bender,” https://github.com/CMU-SAFARI/DRAM-Bender.

[7] Y. Cai *et al.*, “Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis,” in *DATE*, 2012.

[8] Y. Cai *et al.*, “Error Analysis and Retention-Aware Error Management for NAND Flash Memory,” *ITJ*, 2013.

[9] Y. Cai *et al.*, “Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation,” in *ICCD*, 2013.

[10] Y. Cai *et al.*, “Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery,” in *DSN*, 2015.

[11] Y. Cai *et al.*, “Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid-State Drives,” *Proc. IEEE*, 2017.

[12] O. Mutlu, “The RowHammer Problem and Other Issues we may Face as Memory Becomes Denser,” *DATE*, 2017.

[13] O. Mutlu and J. Kim, “RowHammer: A Retrospective,” *IEEE TCAD Special Issue on Top Picks in Hardware and Embedded Security*, 2019.

[14] O. Mutlu *et al.*, “Fundamentally Understanding and Solving RowHammer,” in *ASP-DAC*, 2023.

[15] T. Dullien, “Security, Moore’s Law, and the Anomaly of Cheap Complexity,” in *CCDCOE*, 2018, https://www.youtube.com/watch?v=q98foLaAfX8.

[16] M. Seaborn and T. 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.

[17] M. Seaborn and T. Dullien, “Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges,” *Black Hat*, 2015.

[18] A. S. Rakin *et al.*, “DeepSteal: Advanced Model Extractions Leveraging Efficient Weight Stealing in Memories,” in *S&P*, 2022.

[19] K. Mus *et al.*, “Jolt: Recovering TLS Signing Keys via Rowhammer Faults,” in *S&P*, 2023.

[20] D. Gruss *et al.*, “Rowhammer.js: A Remote Software-Induced Fault Attack in Javascript,” in *DIMVA*, 2016.

[21] V. van der Veen *et al.*, “Drammer: Deterministic Rowhammer Attacks on Mobile Platforms,” in *CCS*, 2016.

[22] Y. Xiao *et al.*, “One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation,” in *USENIX Security*, 2016.

[23] K. Razavi *et al.*, “Flip Feng Shui: Hammering a Needle in the Software Stack,” *USENIX Security*, 2016.

[24] A. Tatar *et al.*, “Throwhammer: Rowhammer Attacks Over the Network and Defenses,” in *USENIX ATC*, 2018.

[25] M. Lipp *et al.*, “Nethammer: Inducing Rowhammer Faults Through Network Requests,” arXiv:1805.04956, 2018.

[26] L. Cojocar *et al.*, “Exploiting Correcting Codes: On the Effectiveness of ECC Memory Against Rowhammer Attacks,” in *S&P*, 2019.

[27] F. de Ridder *et al.*, “SMASH: Synchronized Many-Sided Rowhammer Attacks from JavaScript,” in *USENIX Security*, 2021.

[28] P. Jattke *et al.*, “Blacksmith: Scalable Rowhammering in the Frequency Domain,” in *S&P*, 2022.

[29] A. Kwong *et al.*, “RAMBleed: Reading Bits in Memory Without Accessing Them,” in *S&P*, 2020.

[30] S. Hong *et al.*, “Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks,” in *SS*, 2019.

[31] F. Yao *et al.*, “Deephammer: Depleting the Intelligence of Deep Neural Networks Through Targeted Chain of Bit Flips,” in *USENIX Security*, 2020.

[32] A. G. Yağlıkcı *et al.*, “BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows,” in *HPCA*, 2021.

[33] M. Marazzi *et al.*, “ProTRR: Principled yet Optimal In-DRAM Target Row Refresh,” in *S&P*, 2023.

[34] M. Wi *et al.*, “SHADOW: Preventing Row Hammer in DRAM with Intra-Subarray Row Shuffling,” in *HPCA*. IEEE, 2023.

[35] J. Juffinger *et al.*, “CSI: Rowhammer–Cryptographic Security and Integrity against Rowhammer (to appear),” in *S&P*, 2023.

[36] J. S. Kim *et al.*, “Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques,” in *ISCA*, 2020.

[37] PassMark Software, “MemTest86: The Original Industry Standard Memory Diagnostic Utility,” http://www.memtest86.com/troubleshooting.htm, 2015.

[38] Apple Inc., “About the Security Content of Mac EFI Security Update 2015-001,” https://support.apple.com/en-us/HT204934, 2015.

[39] P. Frigo *et al.*, “TRRespass: Exploiting the Many Sides of Target Row Refresh,” in *S&P*, 2020.

[40] M. Kaczmarski, “Thoughts on Intel Xeon E5-2600 v2 Product Family Performance Optimisation – Component Selection Guidelines,” http://infobazy.gda.pl/2014/pliki/prezentacje/d2s2e4-Kaczmarski-Optymalna.pdf, page 13, 2014.

[41] H. Hassan *et al.*, “Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications,” in *MICRO*, 2021.

[42] JEDEC, *JEP300-1: Near-Term DRAM Level RowHammer Mitigation*, 2021.

[43] JEDEC, *JEP301-1: System Level RowHammer Mitigation*, 2021.

[44] W. Kim, “A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement,” in *ISSCC*, 2023.

[45] S. Hong *et al.*, “DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm,” arXiv:2302.03591, 2023.

[46] A. Kogler *et al.*, “Half-Double: Hammering From the Next Row Over,” in *USENIX Security*, 2022.

[47] L. Cojocar *et al.*, “Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers,” in *S&P*, 2020.

[48] K. Loughlin *et al.*, “MOESI-Prime: Preventing Coherence-Induced Hammering in Commodity Workloads,” in *ISCA*, 2022.

[49] T. Bennett *et al.*, “Panopticon: A Complete In-DRAM Rowhammer Mitigation,” in *DRAMSec*, 2021.

[50] K. Loughlin *et al.*, “Stop! Hammer Time: Rethinking Our Approach to Rowhammer Mitigations,” in *HotOS*, 2021.

[51] S. Saroiu and A. Wolman, “How to Configure Row-Sampling-Based Rowhammer Defenses,” *DRAMSec*, 2022.

[52] O. Mutlu, “Memory Scaling: A Systems Architecture Perspective,” in *IMW*, 2013.

[53] L. Orosa, “A Deeper Look into RowHammer’s Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses,” in *MICRO*, 2021.

[54] A. G. Yağlıkcı *et al.*, “Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices,” in *DSN*, 2022.

[55] H. Luo *et al.*, “RowPress: Amplifying Read Disturbance in Modern DRAM Chips,” in *ISCA*, 2023.