WISSEN
TECHNIK
LEIDENSCHAFT

Rowhammer and FPGAs

Seminar Presentation
Digital System Integration and Programming

Fabian Gruber

Outline

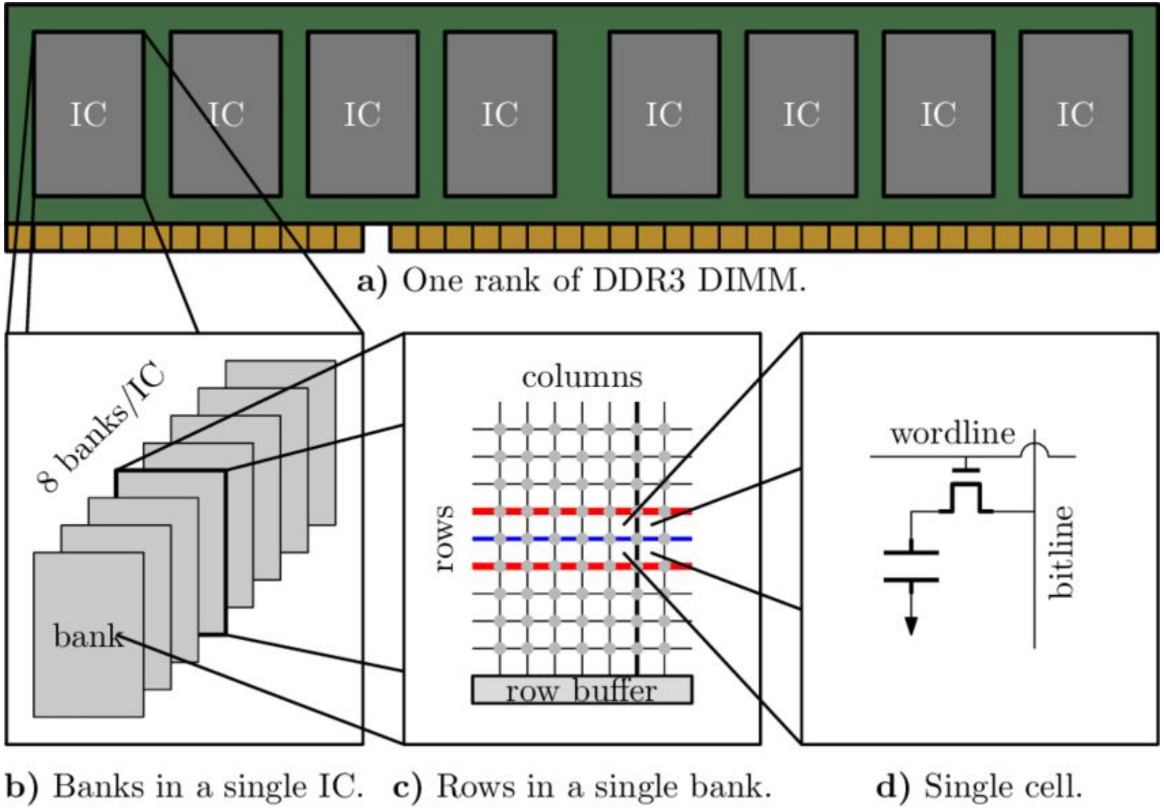
- What is Rowhammer?
- How can FPGAs be used to perform such attacks?
- What are existing mitigations?

Dynamic Random Access Memory (DRAM)

- DRAM consists of multiple ICs
- ICs consist of Banks
- Banks consist of Rows
- Rows consist of of cells
- Each cell has a transistor and capacitor

[3]

Rowhammer and FPGAs



a) One rank of DDR3 DIMM.

b) Banks in a single IC. c) Rows in a single bank. d) Single cell.

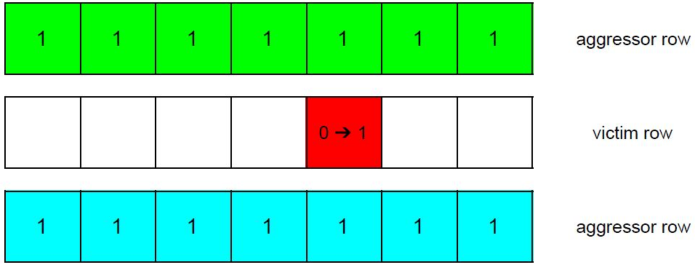
[3]

Dynamic Cell Refreshing

- Cells lose charge over time
- Refresh periodically
- Problem for Rowhammer (need to flip bits within interval)
- Otherwise progress gets reset

Rowhammer

- Hardware vulnerability
- Repeated access of rows leaks into neighboring rows/cells
- Bitflips in inaccessible memory



1	1	1	1	1	1	1	aggressor row
				0 → 1			victim row
1	1	1	1	1	1	1	aggressor row

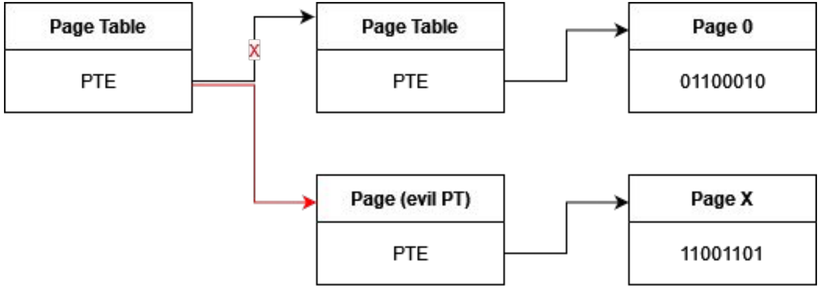
[1]

Rowhammer Attacks

- Target Page Tables (PTs) and Entries (PTEs)
  - Get arbitrary read and write access
- Fault attacks on RSA (Bellcore Attack)
  - Create faulty signature -> retrieve secret data

Privilege Escalation Attack

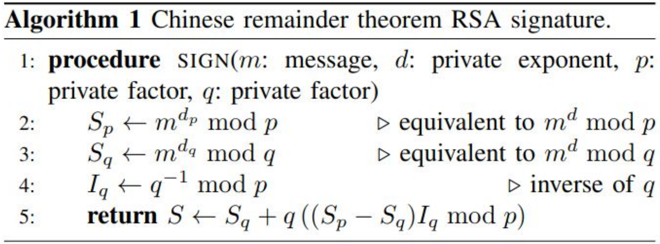
- Create attacker controlled PT
- Hammer PFN in attacker process
- Arbitrary memory access



[4]

Bellcore Attack

- Chinese Remainder Theorem (CRT) used in RSA to compute **m^d mod N** where **N = pq**
- Fault in **S_p** or **S_q** -> faulty signature **S'**
- Difference between **S** and **S'** with same m leaks **p** or **q**



Algorithm 1 Chinese remainder theorem RSA signature.

```
procedure SIGN($m$: message, $d$: private exponent, $p$: private factor, $q$: private factor)
    $S_p \leftarrow m^{d_p} \bmod p$                     ▷ equivalent to $m^d \bmod p$
    $S_q \leftarrow m^{d_q} \bmod q$                     ▷ equivalent to $m^d \bmod q$
    $I_q \leftarrow q^{-1} \bmod p$                         ▷ inverse of $q$
    return $S \leftarrow S_q + q\,((S_p - S_q) I_q \bmod p)$
```

[2]

Rowhammer Attacks from CPU

- Place victim data at location we can “hammer”
- Access “aggressor” rows
- Flush cache (access has to go to DRAM)
- Wait for biflips

Rowhammer Attacks from FPGA

- Same as before, but NO cache because of Direct Memory Access (DMA)
- Bitflip rates are a lot higher
- Not detectable by some mitigations (use cache)

Threat Model

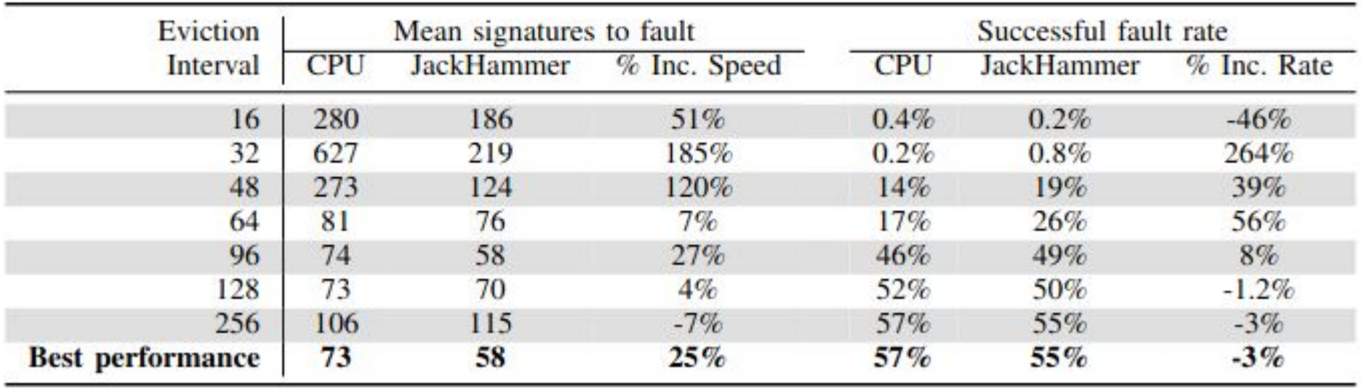
- Malicious 3rd party IPs
  - Untrusted vendors
  - Altered during data transfer
  - Contain hardware trojans

JackHammer

- FPGA implementation of Rowhammer
- Heterogeneous FPGA-CPU platform
- Fault attack on SSL RSA implementation running on CPU
- Rowhammer attack from FPGA
- Bypass cache for more performance

[2]

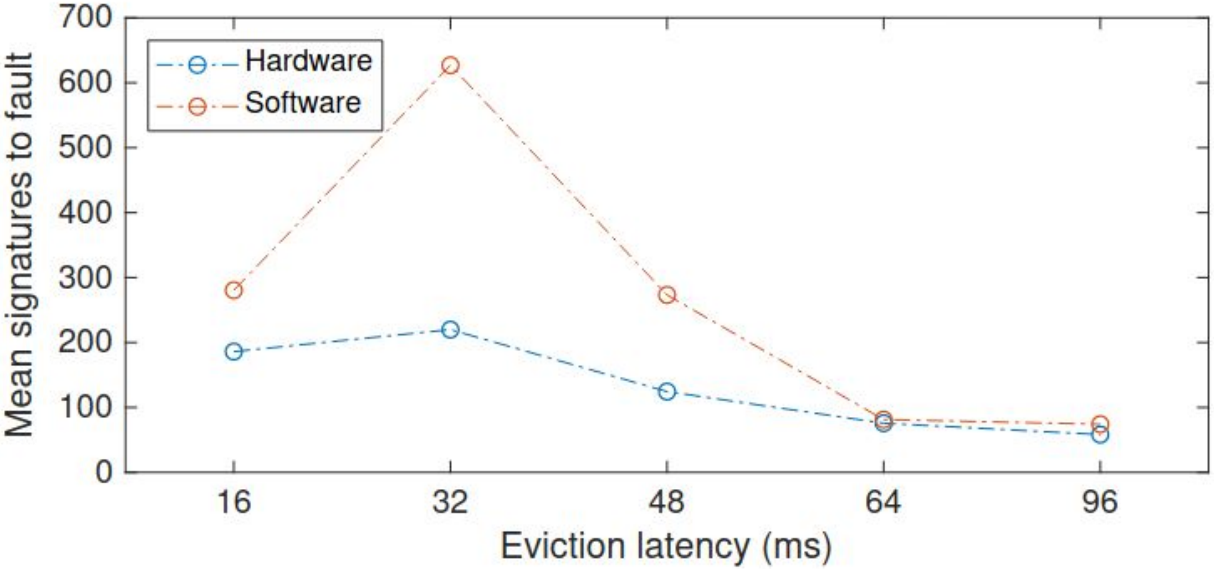
Performance CPU vs JackHammer



Eviction Interval	Mean signatures to fault			Successful fault rate		
	CPU	JackHammer	% Inc. Speed	CPU	JackHammer	% Inc. Rate
16	280	186	51%	0.4%	0.2%	-46%
32	627	219	185%	0.2%	0.8%	264%
48	273	124	120%	14%	19%	39%
64	81	76	7%	17%	26%	56%
96	74	58	27%	46%	49%	8%
128	73	70	4%	52%	50%	-1.2%
256	106	115	-7%	57%	55%	-3%
Best performance	**73**	**58**	**25%**	**57%**	**55%**	**-3%**

[2]

Rowhammer and FPGAs



[2]

Countermeasures

- Hardware Performance Counters [2]
- Increased DRAM row refresh rate [1]
- Verify RSA signatures [5]
- Hash-based PTE protection
- Monitoring of DMA transfers [5]

Hardware Performance Counters

- Detect “hammering” by looking at HPC
- Count access to DRAM rows
- After some threshold -> refresh

Increasing DRAM Refresh Rate

- Cells get refresh more often
- Leaked charge gets reset to valid bit
- Makes attack more difficult

Hash-based PTE Protection

- Compute hash of every PTE
- Update on valid change
- Verify periodically
- Potentially restore corrupted PTE

Monitoring of DMA Transactions

- Implemented in FPGA
- If rows in the same bank accessed more than threshold

  -> Rowhammer

Bibliography

[1] Kim et al. “**Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors**”. 2014. https://users.ece.cmu.edu/~yoonguk/papers/kim-isca14.pdf

[2] Weissman et al. “**JackHammer: Efficient Rowhammer on Heterogeneous FPGA-CPU Platforms**”. https://arxiv.org/abs/1912.11523

[3] Poddebniak et al. “**Attacking Deterministic Signature Schemes Using Fault Attacks**”. https://www.researchgate.net/publication/326276966_Attacking_Deterministic_Signature_Schemes_Using_Fault_Attacks

Bibliography

[4] Project Zero. “**Exploiting the DRAM rowhammer bug to gain kernel privileges**”. https://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html

[5] Elnaggar et al. “**Detection of Rowhammer Attacks in SoCs with FPGAs**”. https://ieeexplore.ieee.org/document/9131554