The DRAM Latency PUF:

Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim Minesh Patel

Hasan Hassan Onur Mutlu

Systems@ETH Zürich

SAFARI

ETH Zürich

Carnegie Mellon

Executive Summary

- **Motivation**:
  - We can authenticate a system via **unique signatures** if we can evaluate a **Physical Unclonable Function (PUF)** on it
  - Signatures **(PUF response)** reflect inherent properties of a device
  - DRAM is a promising substrate for PUFs because it is **widely** used
- **Problem**: Current DRAM PUFs are 1) very slow, 2) require a DRAM reboot, or 3) require additional custom hardware
- **Goal**: To develop a novel and effective PUF for **existing** commodity DRAM devices with **low-latency evaluation time** and **low system interference** across **all operating temperatures**
- **DRAM Latency PUF:** Reduce DRAM access latency **below reliable values** and exploit the resulting error patterns as **unique identifiers**
- **Evaluation:**
  1. Experimentally characterize **223 real LPDDR4 DRAM devices**
  2. **DRAM latency PUF** (88.2 ms) achieves a speedup of **102x/860x** at 70°C/55°C over prior DRAM PUF evaluation mechanisms

The DRAM Latency PUF Outline

- Motivation
- Effective PUF Characteristics
- DRAM Latency PUF
  - DRAM Operation
  - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- Methodology
- Results
- Summary

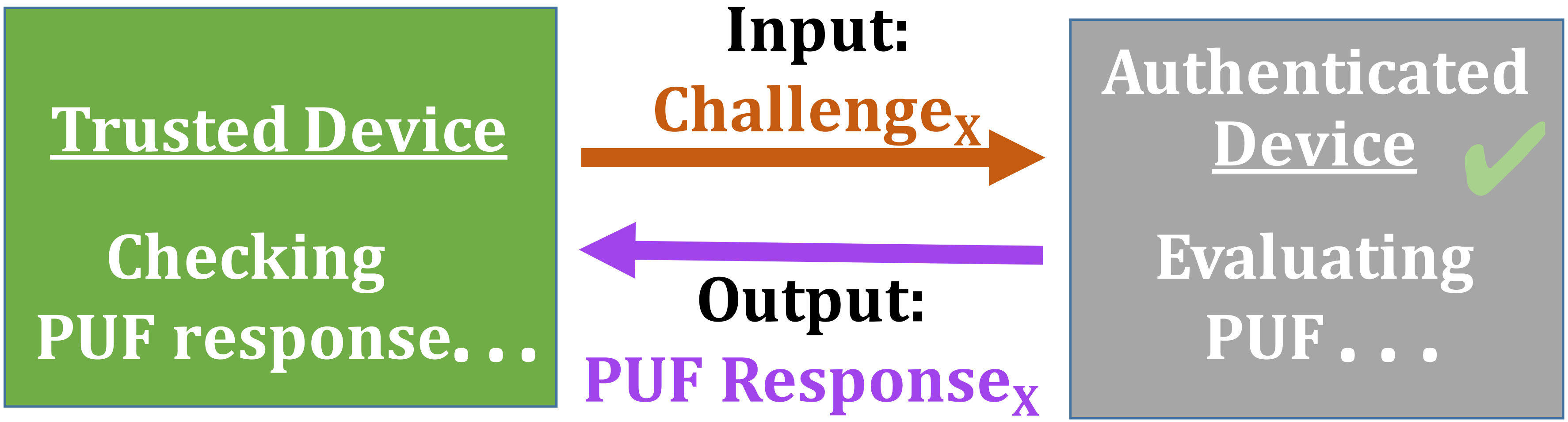
The DRAM Latency PUF Outline

- **Motivation**
- Effective PUF Characteristics
- DRAM Latency PUF
    - DRAM Operation
    - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
    - DRAM Cell Retention
    - Key Idea
    - Weaknesses
- Methodology
- Results
- Summary

Motivation

We want a way to ensure that a system's components are not **compromised**

- **Physical Unclonable Function (PUF):** a function we **evaluate** on a device to **generate** a **signature** **unique** to the device
- We refer to the unique signature as a **PUF response**
- Often used in a **Challenge-Response Protocol (CRP)**



Motivation

1. We want a **runtime-accessible** PUF
   - Should be evaluated **quickly** with **minimal** impact on concurrent applications
   - Can protect against **attacks that swap system components with malicious parts**

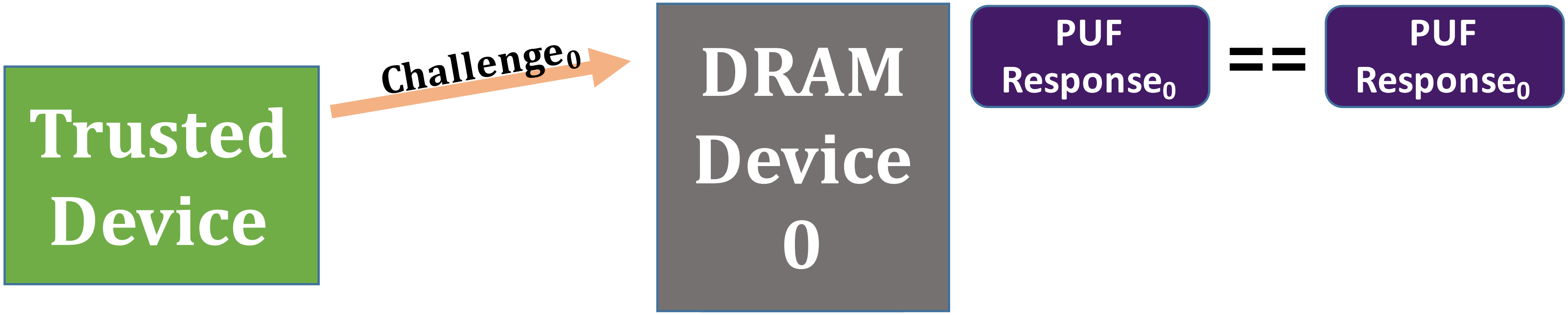
2. DRAM is a **promising substrate** for evaluating PUFs because it is **ubiquitous** in modern systems
   - Unfortunately, current DRAM PUFs are **slow** and get **exponentially slower** at lower temperatures

The DRAM Latency PUF Outline

- Motivation
- **Effective PUF Characteristics**
- DRAM Latency PUF
  - DRAM Operation
  - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- Methodology
- Results
- Summary

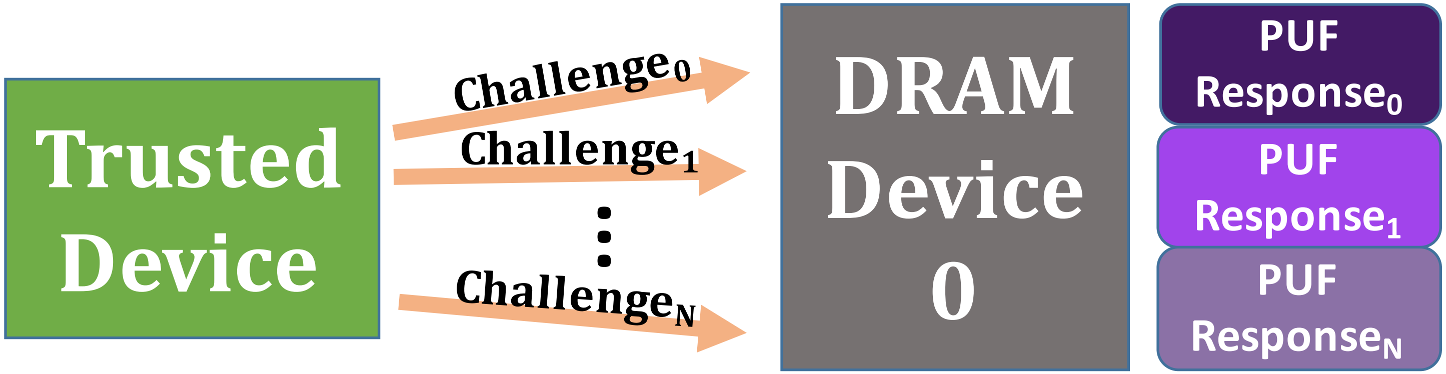
Effective PUF Characteristics

1. Repeatability



Effective PUF Characteristics

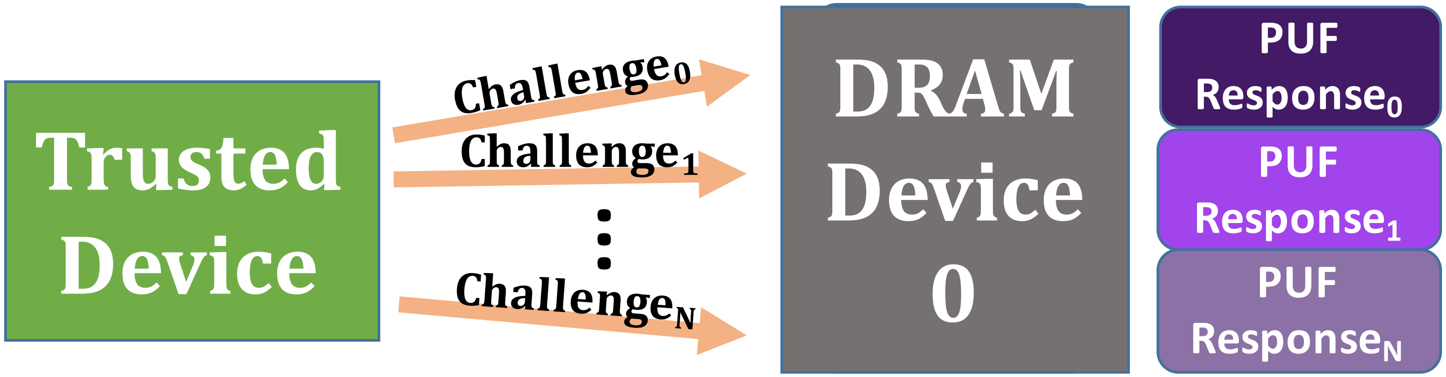
1. Repeatability
2. Diffuseness



Effective PUF Characteristics

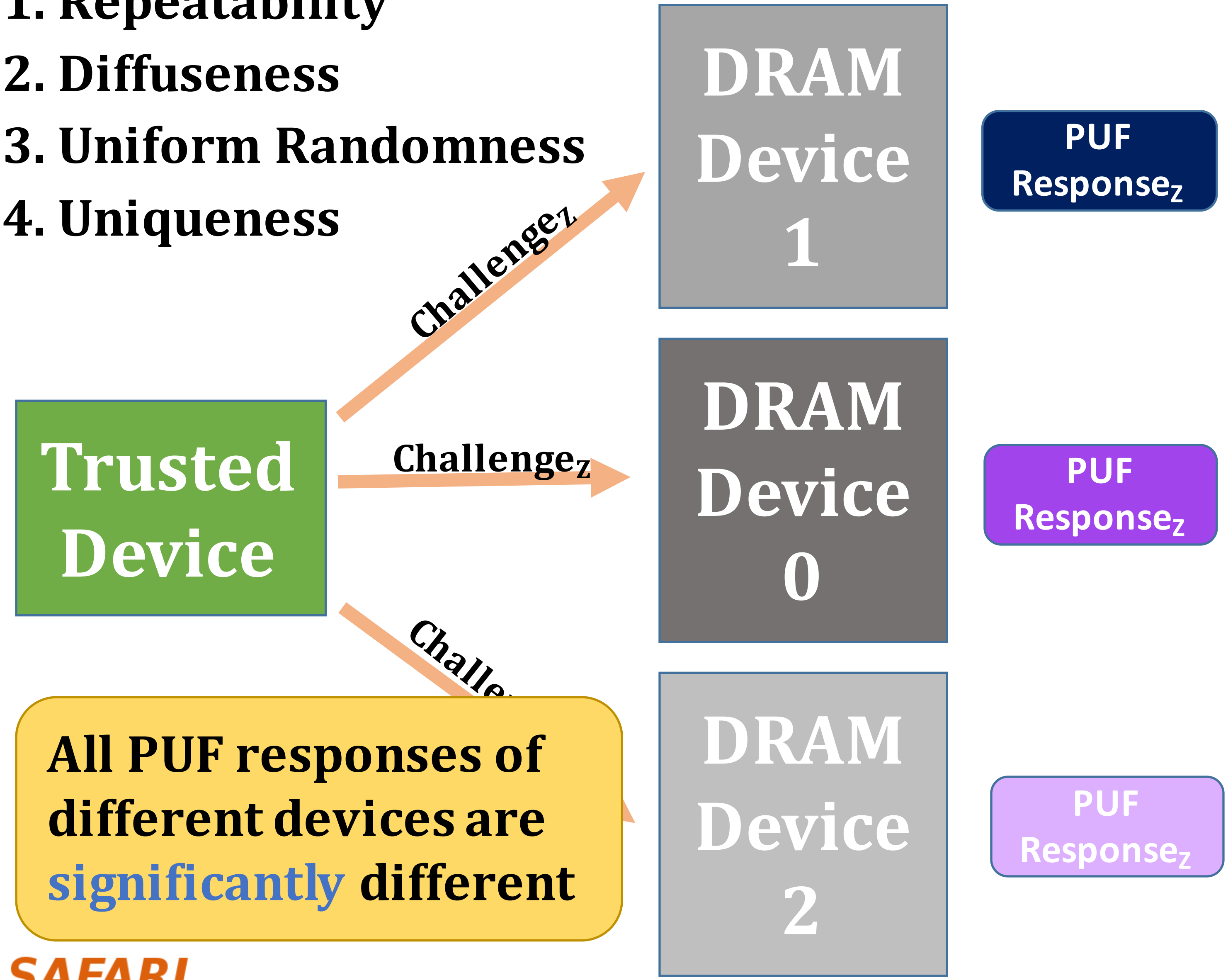
1. Repeatability
2. Diffuseness
3. Uniform Randomness

Cannot use multiple challenge-response pairs to guess another



Effective PUF Characteristics

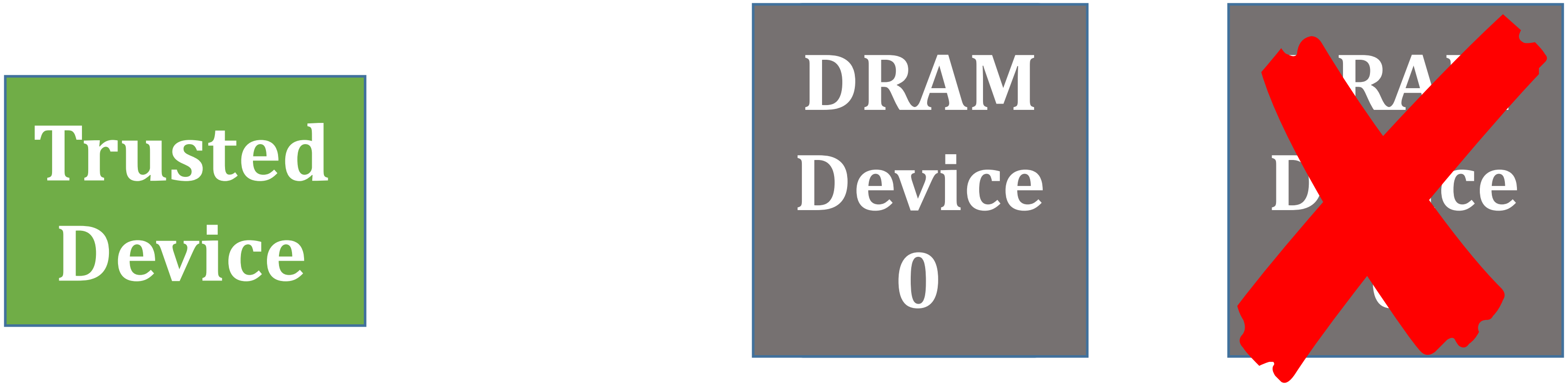
1. Repeatability
2. Diffuseness
3. Uniform Randomness
4. Uniqueness



All PUF responses of different devices are significantly different

Effective PUF Characteristics

1. Repeatability
2. Diffuseness
3. Uniform Randomness
4. Uniqueness
5. Unclonability



Effective PUF Characteristics

1. Repeatability
2. Diffuseness
3. Uniform Randomness
4
5

More analysis
of the effective PUF characteristics
in the paper

Effective PUF Characteristics

Runtime-accessible PUFs must have

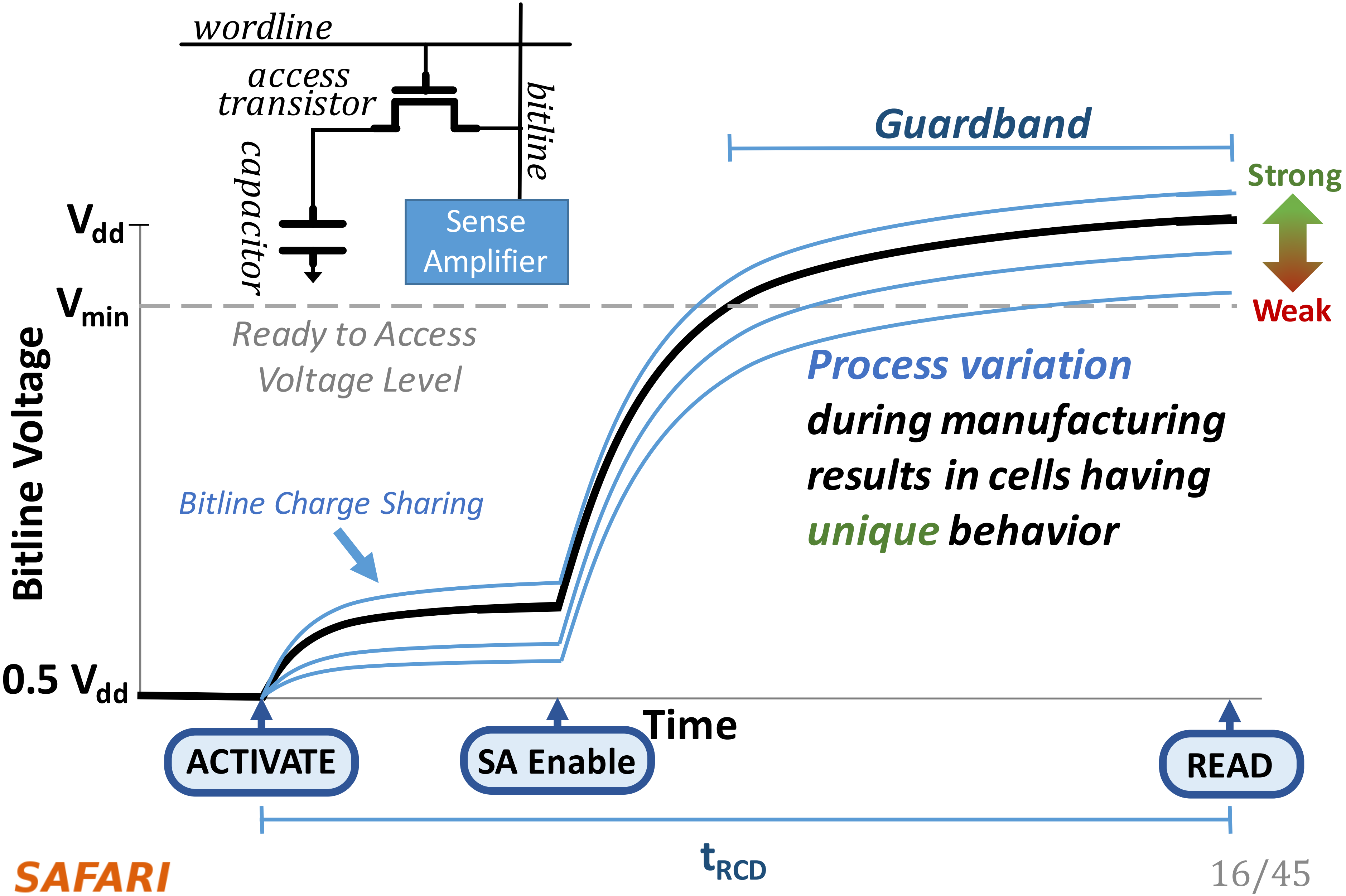
1. **Low Latency**
   - Each device can **quickly** generate a PUF response

2. **Low System Interference**
   - PUF evaluation **minimally affects performance** of concurrently-running applications

The DRAM Latency PUF Outline

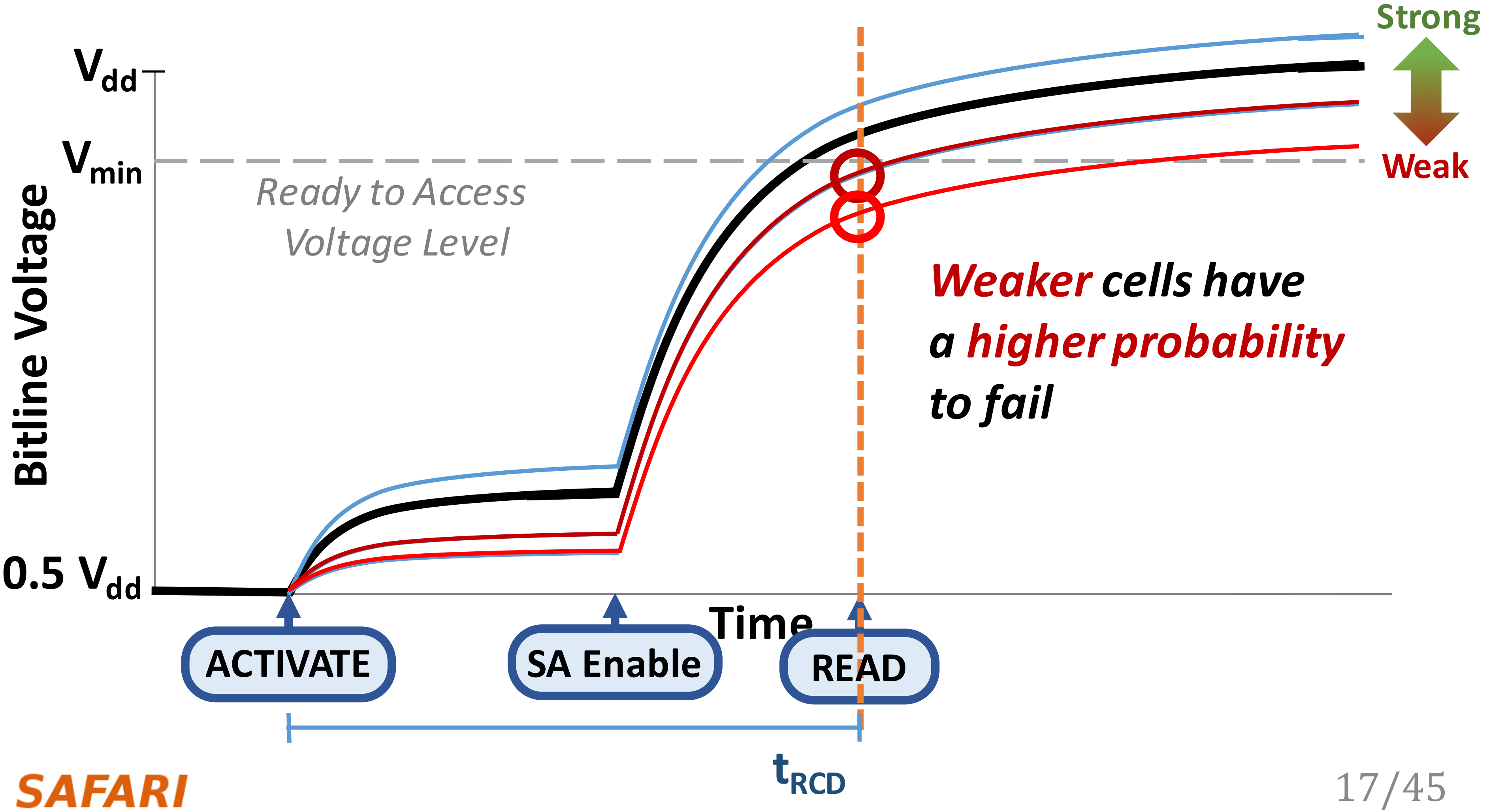
- Motivation
- Effective PUF Characteristics
- **DRAM Latency PUF**
  - **DRAM Operation**
  - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- Methodology
- Results
- Summary

DRAM Accesses and Failures



***Process variation* during manufacturing results in cells having *unique* behavior**

DRAM Accesses and Failures



***Weaker** cells have a **higher probability** to fail*

SAFARI

The DRAM Latency PUF Outline

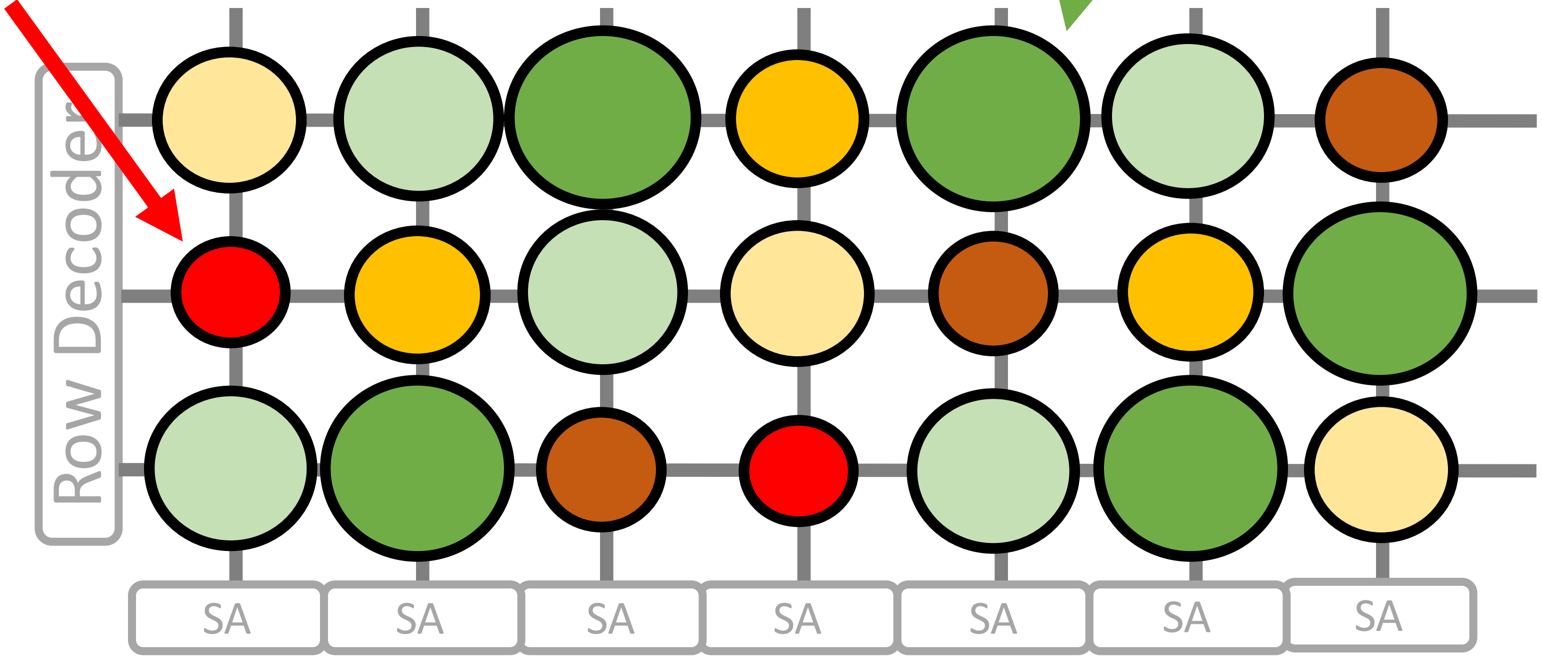
- Motivation
- Effective PUF Characteristics
- **DRAM Latency PUF**
  - DRAM Operation
  - **Key Idea**
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- Methodology
- Results
- Summary

DRAM Latency PUF Key Idea

- A cell's latency failure probability is inherently related to **random process variation** from manufacturing
- We can provide **repeatable and unique device signatures** using latency error patterns

High % chance to fail with reduced t_{RCD}

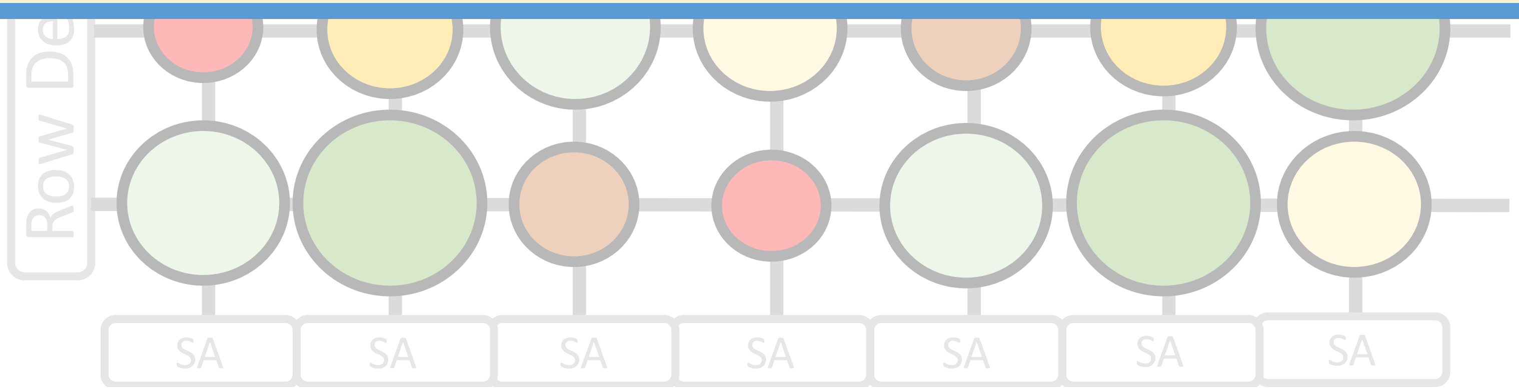
Low % chance to fail with reduced t_{RCD}



DRAM Latency PUF Key Idea

- A cell's latency failure probability is inherently related to **random process variation** from manufacturing
- We can provide **repeatable and unique device**

The **key idea** is to compose a PUF response using the DRAM cells that fail with **high probability**

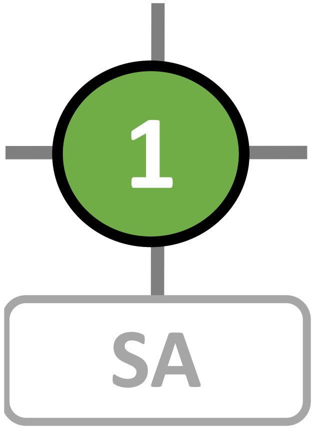


Evaluating a DRAM Latency PUF

Determine whether a **single cell's location** should be included in a DRAM latency PUF response

- **Include** if the cell **fails** with a probability greater than a **chosen threshold** when accessed with a reduced $\mathbf{t_{RCD}}$

Chosen Threshold: 50%



This Cell's Failure Rate: 60%

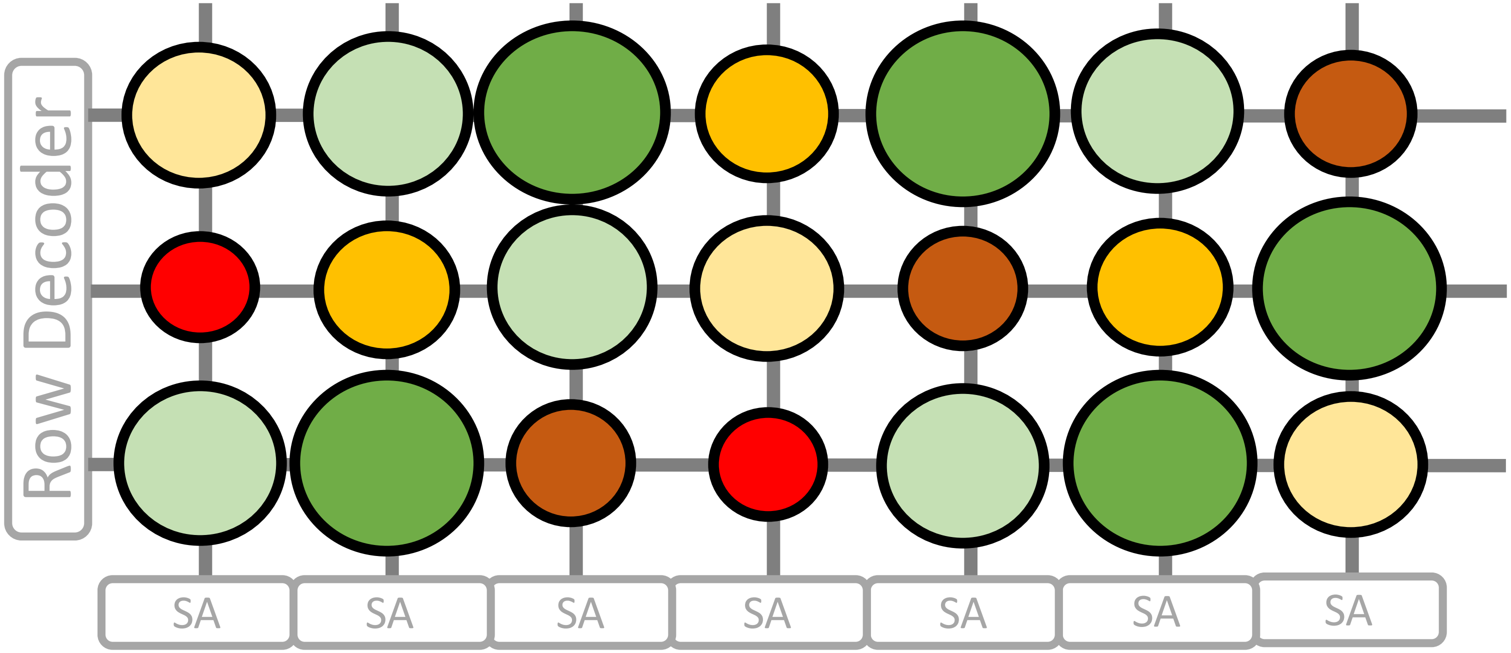
Failure rate is greater than the chosen threshold, so the cell's location should be included

1 ~~0~~ 1 ~~0~~ 1 ~~0~~ ~~0~~ 1 ~~0~~ ~~0~~

Evaluating a DRAM Latency PUF

- We induce latency failures **100 times** and use a **threshold of 10%** (i.e., use cells that fail > 10 times)
- We do this for every cell in a continuous **8KiB** memory region, that we refer to as a **PUF memory segment**

Example 21-bit PUF memory segment



Evaluating a DRAM Latency PUF

- We induce latency failures **100 times** and use a **threshold of 10%** (i.e., use cells that fail > 10 times)
- We do this for every cell in a continuous **8KiB** memory region, that we refer to as a **PUF memory segment**

PUF Response

0	0	0	1	0	0	1
1	0	0	1	1	0	0
0	0	1	1	0	0	0

Evaluating a DRAM Latency PUF

- We induce latency failures **100 times** and use a **threshold of 10%** (i.e., use cells that fail > 10 times)
- We do this for every cell in a continuous **8KiB** memory

We can evaluate the DRAM latency PUF in only 88.2ms on average regardless of temperature!

The DRAM Latency PUF Outline

Motivation

Effective PUF Characteristics

DRAM Latency PUF

- DRAM Operation
- Key Idea

Prior Best DRAM PUF: DRAM Retention PUF

- **DRAM Cell Retention**
- Key Idea
- Weaknesses

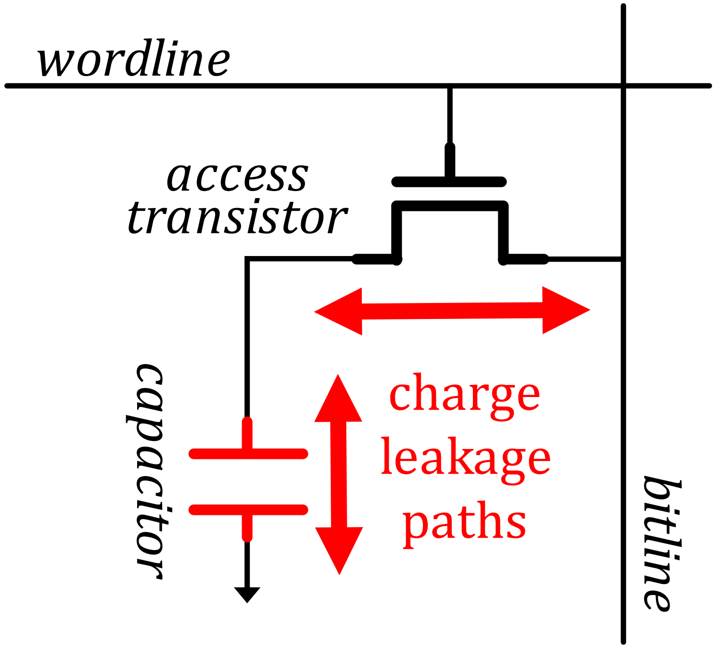
Methodology

Results

Summary

DRAM Cell Leakage

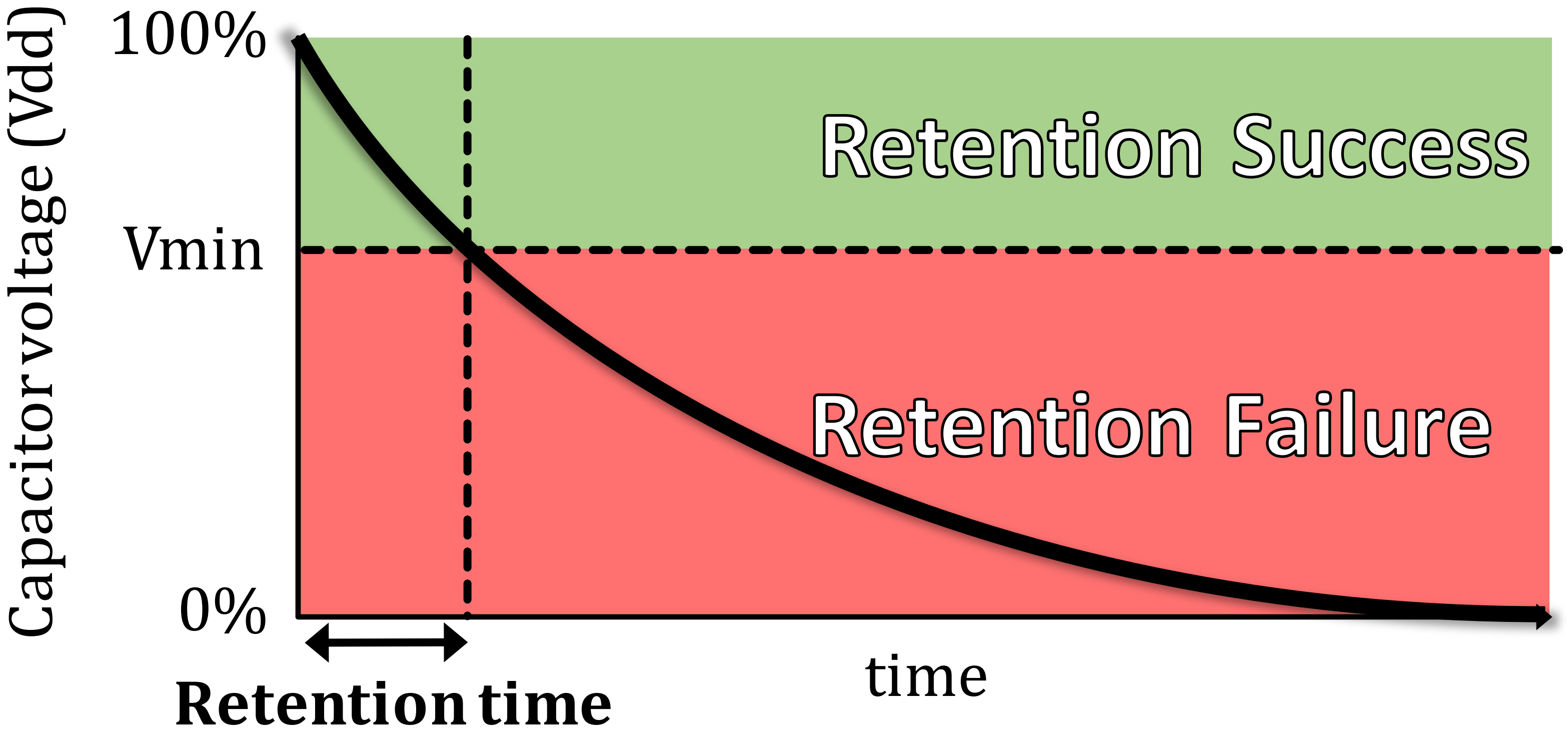
DRAM encodes information in leaky capacitors



Stored data is corrupted if too much charge leaks (i.e., the capacitor voltage degrades too much)

[Patel et al., REAPER, ISCA'17]

DRAM Cell Retention



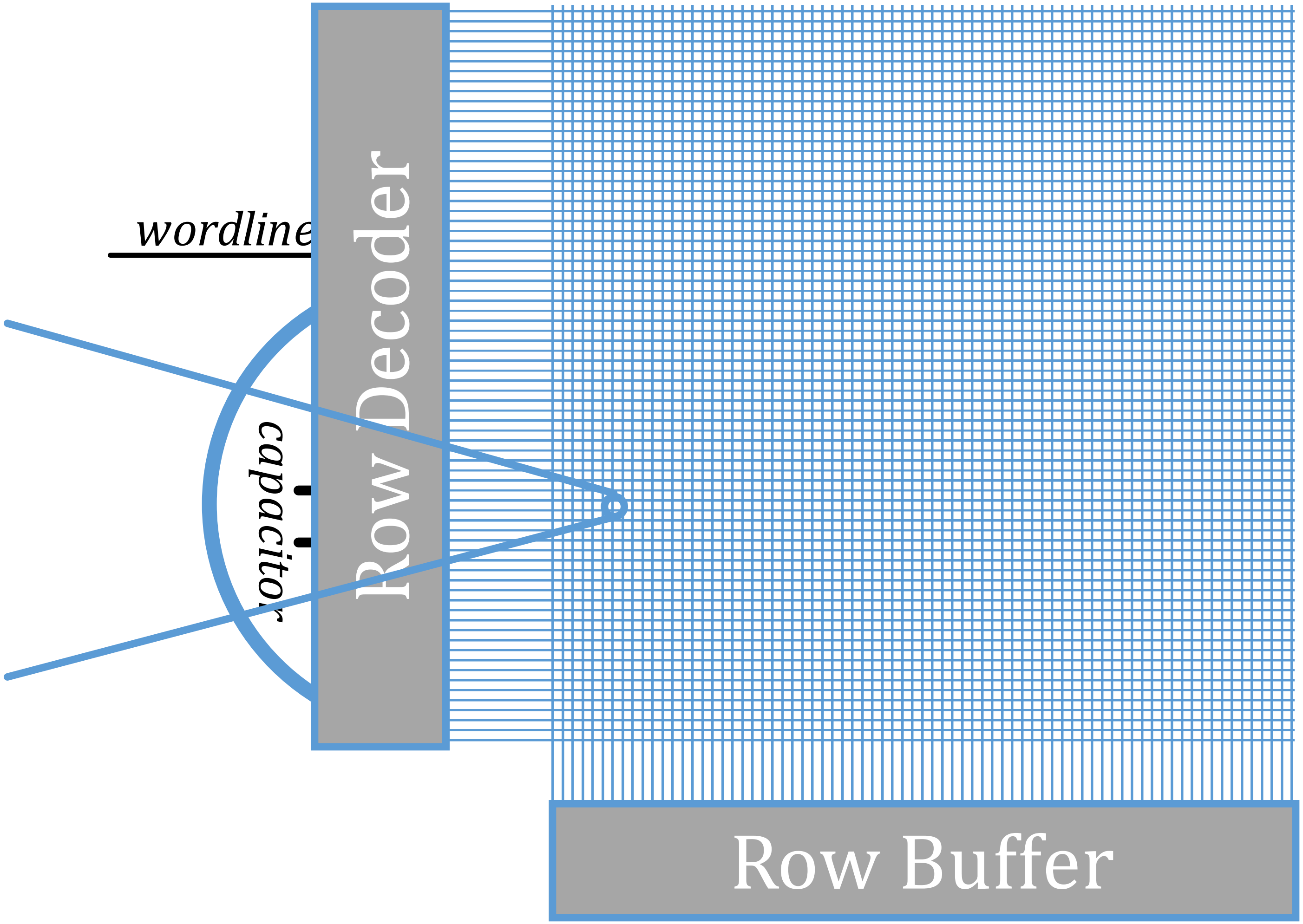
Retention failure – when leakage corrupts stored data

Retention time – how long a cell holds its value

SAFARI

[Patel et al., REAPER, ISCA’17]

Each Cell has a Different Retention Time



8GB DRAM = 6.4e10 cells

SAFARI

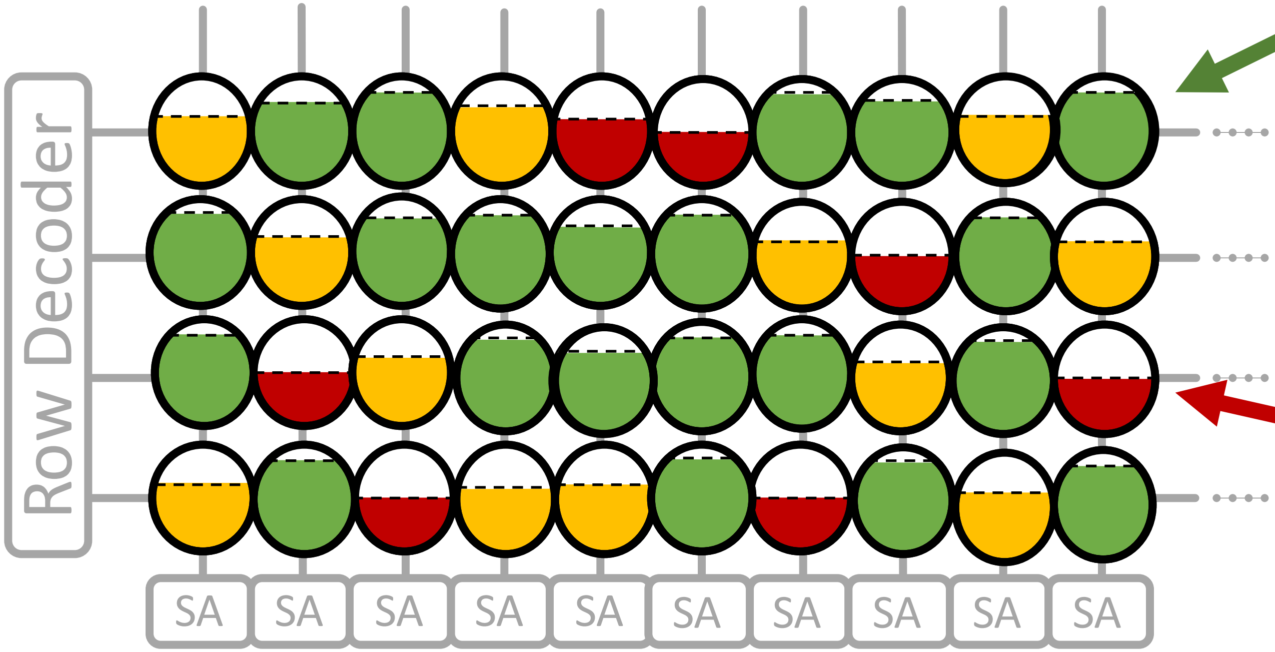
[Patel et al., REAPER, ISCA'17]

The DRAM Latency PUF Outline

- Motivation
- Effective PUF Characteristics
- DRAM Latency PUF
    - DRAM Operation
    - Key Idea
- **Prior Best DRAM PUF: DRAM Retention PUF**
    - DRAM Cell Retention
    - **Key Idea**
    - Weaknesses
- Methodology
- Results
- Summary

Evaluating a DRAM Retention PUF

Generate a **PUF response** with locations of cells in a **PUF memory segment** that **fail** with a **refresh interval *N***



Can handle a longer refresh interval

Fails with refresh interval **N**

The pattern of retention failures across a segment of DRAM is **unique** to the device

Evaluating a DRAM Retention PUF

Generate a **PUF response** with locations of cells in a **PUF memory segment** that **fail** with a **refresh interval *N***

Can handle a

We use the best methods from prior work and optimize the retention PUF for our devices

DRAM is **unique** to the device

The DRAM Latency PUF Outline

- Motivation
- Effective PUF Characteristics
- DRAM Latency PUF
    - DRAM Operation
    - Key Idea
- **Prior Best DRAM PUF: DRAM Retention PUF**
    - DRAM Cell Retention
    - Key Idea
    - **Weaknesses**
- Methodology
- Results
- Summary

DRAM Retention PUF Weaknesses

DRAM Retention PUF evaluation time is **very long** and leads to **high system interference**

Long evaluation time:

1. Most DRAM cells are strong → need to wait for long time to drain charge from capacitors
2. Especially at low temperatures

High system interference:

1. DRAM refresh can only be disabled at a **channel granularity** **(512MB – 2GB)**
2. Must issue **manual refreshes** to maintain data correctness in the rest of the channel **during entire evaluation time**
3. Manually refreshing DRAM consumes **significant** bandwidth on the DRAM bus

DRAM Retention PUF Weaknesses

Long evaluation time could be ameliorated in 2 ways:

1. **Increase temperature** – higher rate of charge leakage

   → **Observe failures faster**

Unfortunately:

   1. Difficult to control DRAM temperature in the field

   2. Operating at high temperatures is undesirable

2. **Increase PUF memory segment size** – more cells with low retention time in PUF memory segment

   → **Observe more failures faster**

Unfortunately:

   - Large PUF memory segment

     → **high DRAM capacity overhead**

The DRAM Latency PUF Outline

- Motivation
- Effective PUF Characteristics
- DRAM Latency PUF
  - DRAM Operation
  - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- **Methodology**
- Results
- Summary

Methodology

- **223 2y-nm LPDDR4 DRAM devices**
  - **2GB** device size
  - From **3 major DRAM manufacturers**

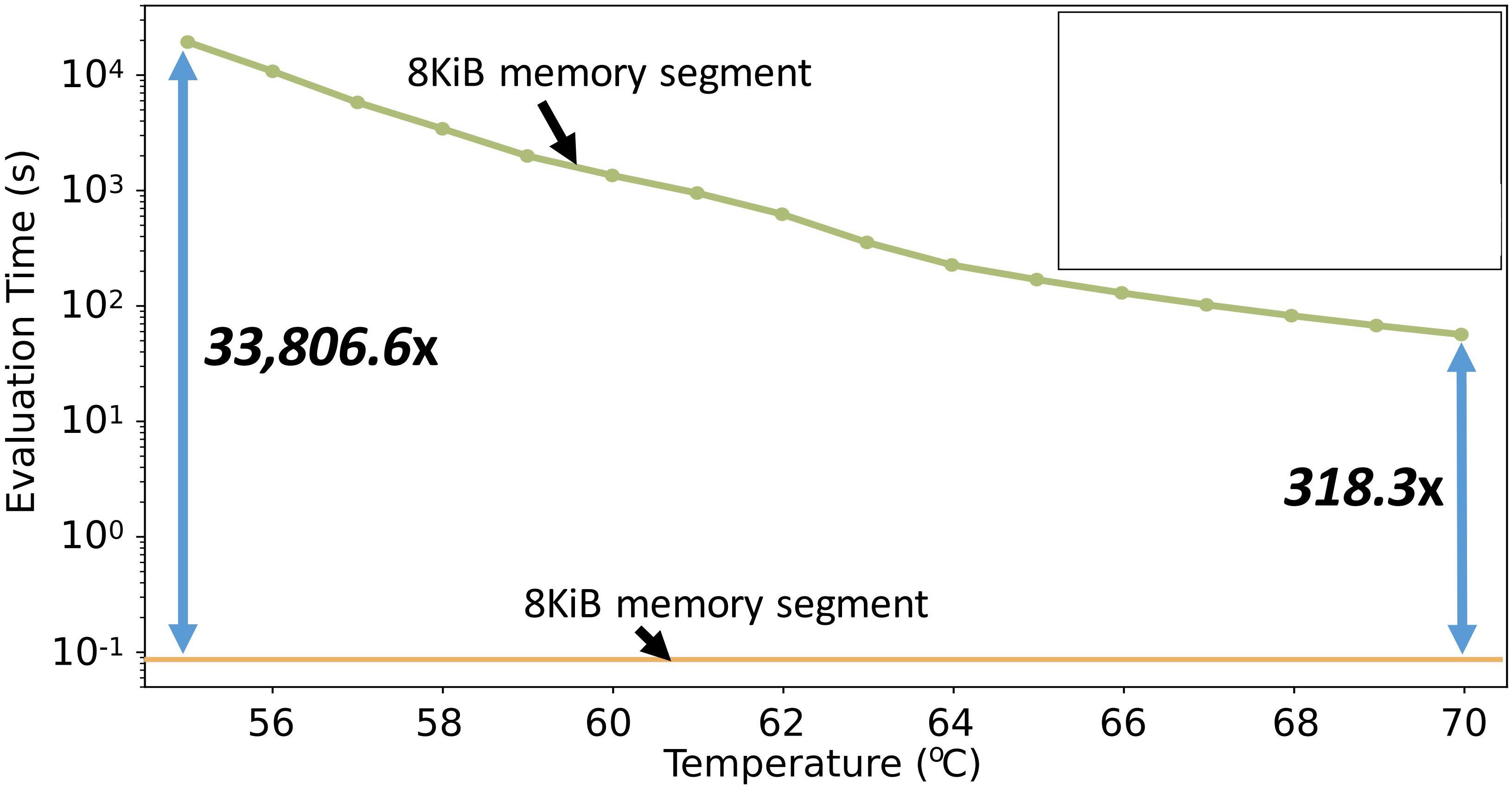
- **Thermally controlled testing chamber**
  - Ambient temperature range: **{40°C – 55°C} ± 0.25°C**
  - DRAM temperature is held at 15°C above ambient

- **Precise control over DRAM commands and timing parameters**
  - Test retention time effects by **disabling refresh**
  - Test reduced latency effects by **reducing t_{RCD} parameter**

The DRAM Latency PUF Outline

- Motivation
- Effective PUF Characteristics
- DRAM Latency PUF
  - DRAM Operation
  - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- Methodology
- **Results**
- Summary

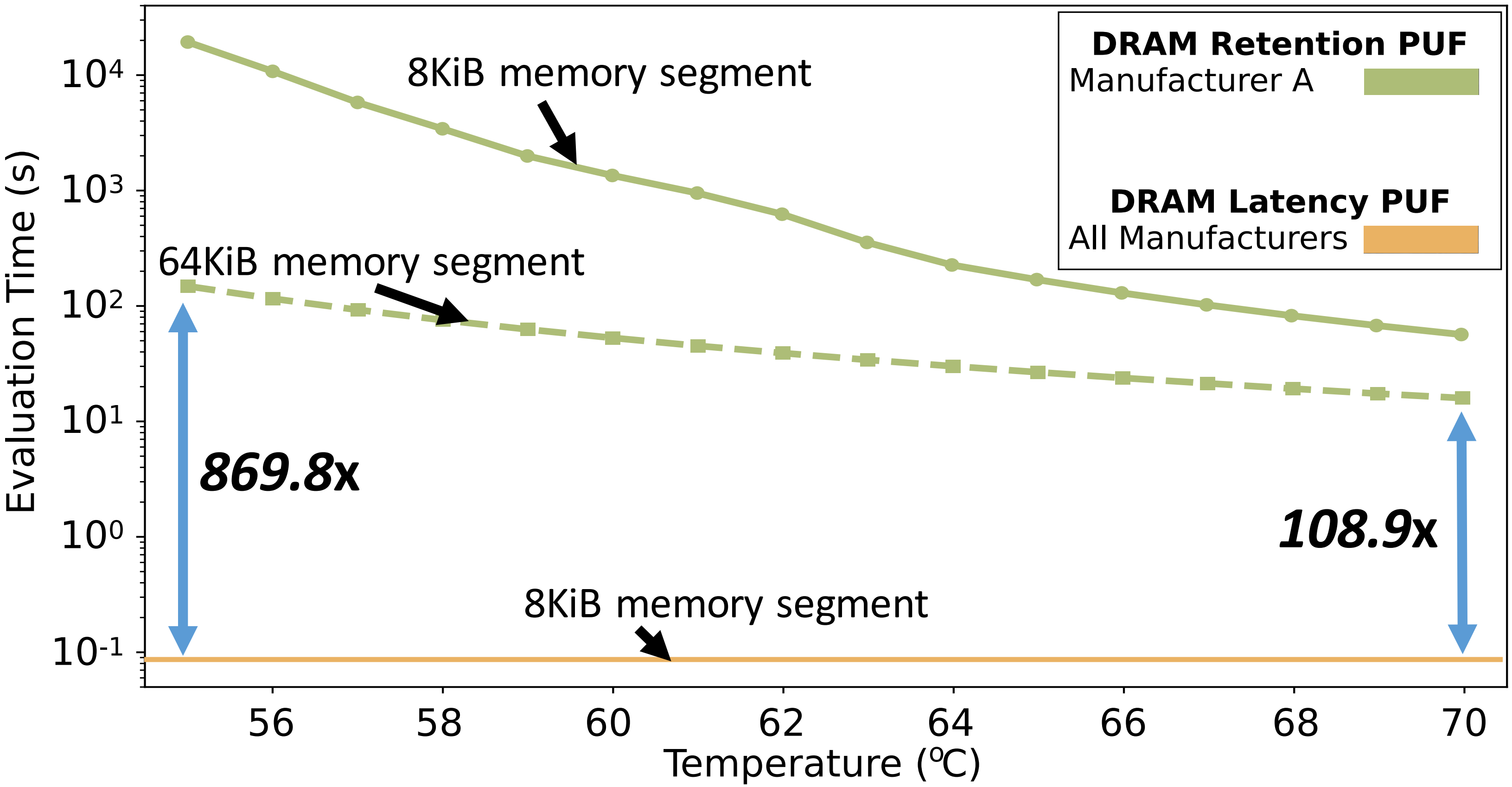
Results – PUF Evaluation Latency



DRAM latency PUF is

1. **Fast and constant latency (88.2ms)**

Results – PUF Evaluation Latency

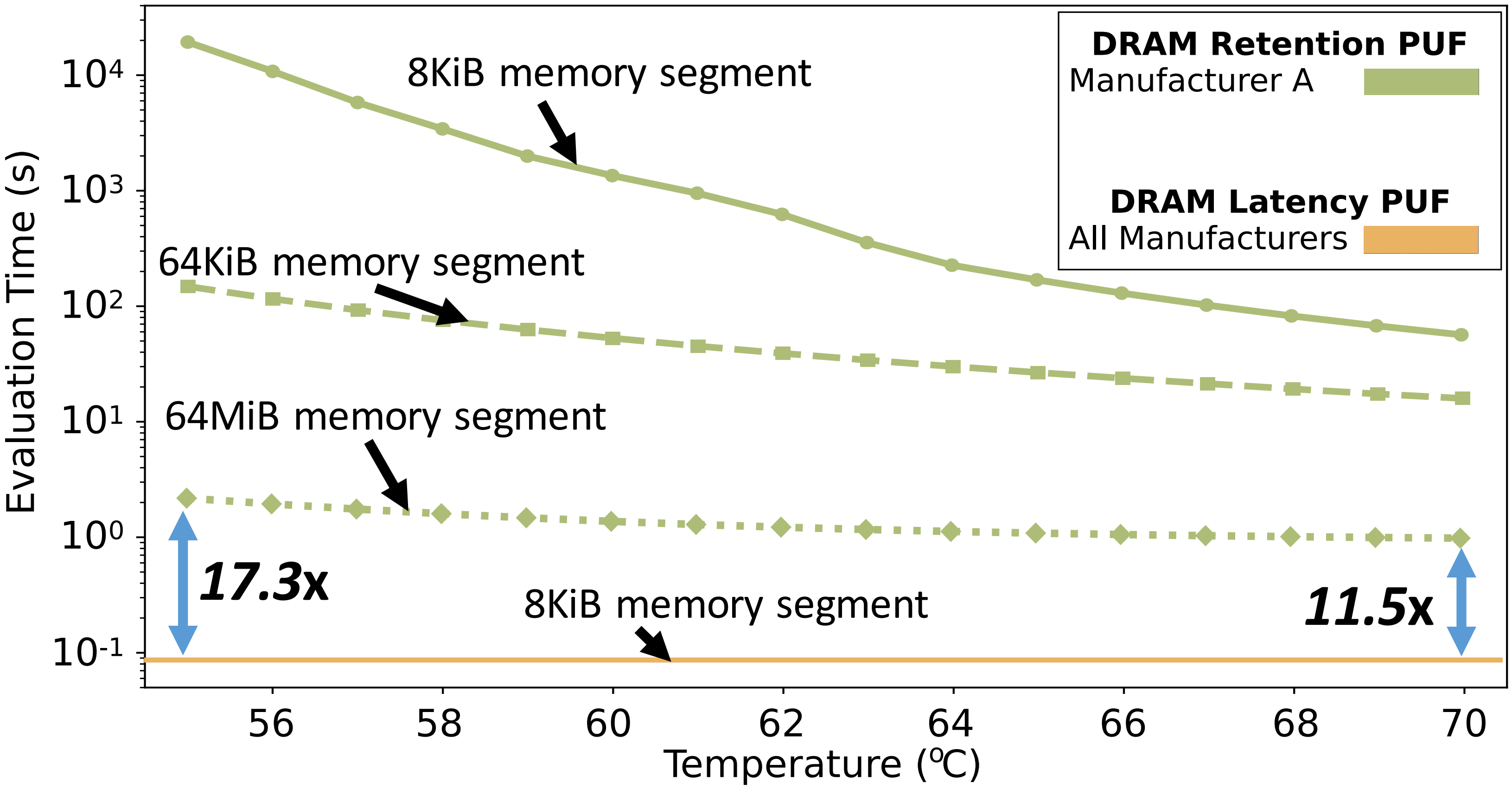


DRAM latency PUF is

1. **Fast and constant latency (88.2ms)**

SAFARI

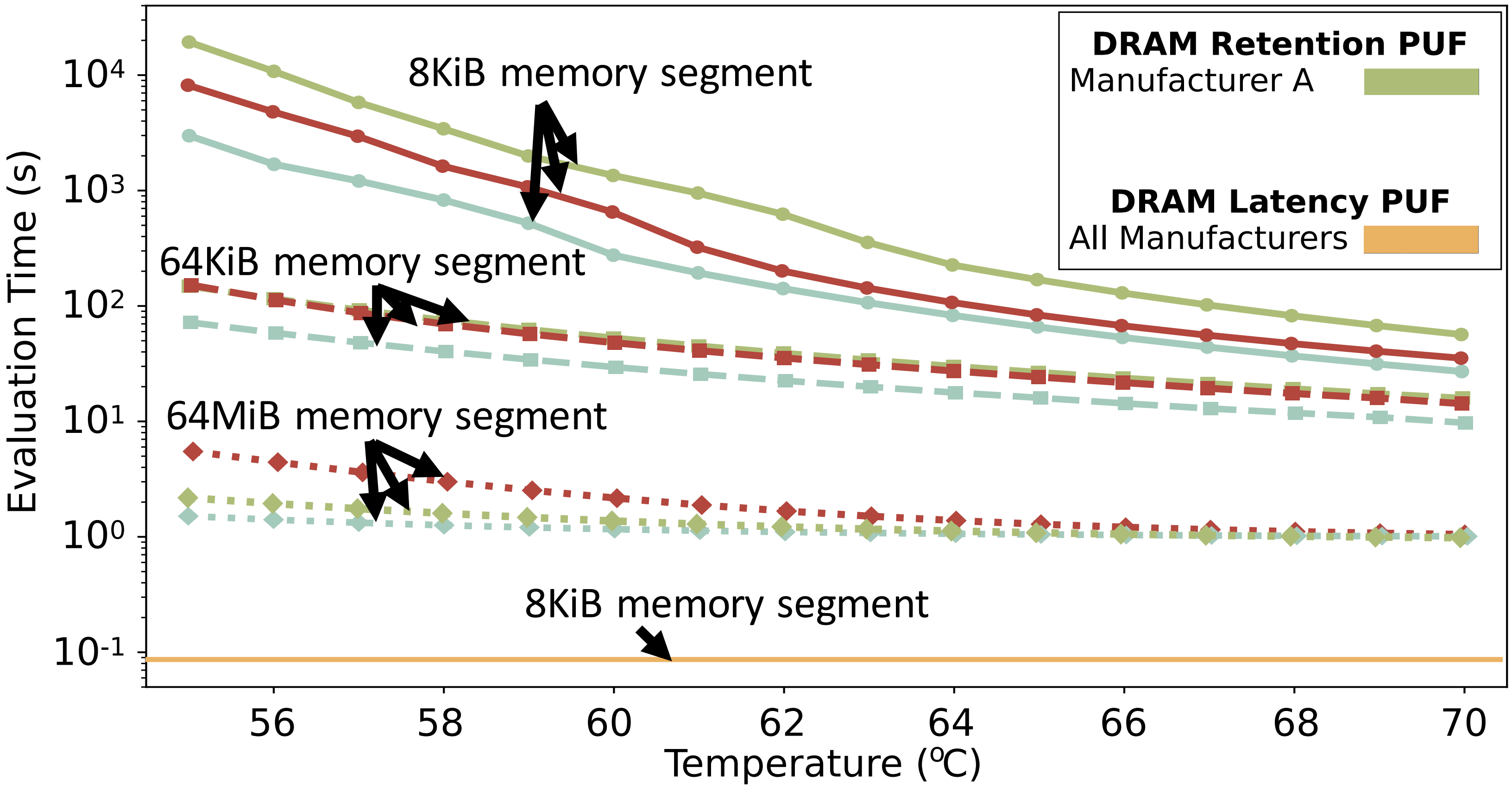
Results – PUF Evaluation Latency



DRAM latency PUF is

1. Fast and constant latency (88.2ms)

Results – PUF Evaluation Latency



DRAM latency PUF is

1. **Fast and constant latency (88.2ms)**
2. **On average, 102x/860x faster than the previous DRAM PUF with the same DRAM capacity overhead (64KiB)**

Results – System Interference

During PUF evaluation on commodity devices:

- **The DRAM Retention PUF**
  - Disables refresh at channel granularity **(~512MB – 2GB)**
    - **Issue manual refresh operations** to rows in channel but not in PUF memory segment to prevent data corruption
  - Has **long evaluation time** at low temperatures

- **The DRAM Latency PUF**
  - Does not require disabling refresh
  - Has short evaluation time **at any operating temperature**

Other Results in the Paper

- **How the DRAM latency PUF meets the basic requirements for an effective PUF**
- **A detailed analysis on:**
  - Devices of **the three major DRAM manufacturers**
  - The **evaluation time** of a PUF
- **Further discussion on:**
  - **Optimizing** retention PUFs
  - **System interference** of DRAM retention and latency PUFs
  - Algorithm to **quickly and reliably** evaluate DRAM latency PUF
  - **Design considerations** for a DRAM latency PUF
  - The DRAM Latency PUF overhead analysis

The DRAM Latency PUF Outline

- Motivation
- Effective PUF Characteristics
- DRAM Latency PUF
  - DRAM Operation
  - Key Idea
- Prior Best DRAM PUF: DRAM Retention PUF
  - DRAM Cell Retention
  - Key Idea
  - Weaknesses
- Methodology
- Results
- **Summary**

Executive Summary

- **Motivation**:
  - We can authenticate a system via **unique signatures** if we can evaluate a **Physical Unclonable Function (PUF)** on it
  - Signatures **(PUF response)** reflect inherent properties of a device
  - DRAM is a promising substrate for PUFs because it is **widely** used
- **Problem**: Current DRAM PUFs are 1) very slow, 2) require a DRAM reboot, or 3) require additional custom hardware
- **Goal**: To develop a novel and effective PUF for **existing** commodity DRAM devices with **low-latency evaluation time** and **low system interference** across **all operating temperatures**
- **DRAM Latency PUF:** Reduce DRAM access latency **below reliable values** and exploit the resulting error patterns as **unique identifiers**
- **Evaluation:**
  1. Experimentally characterize **223 real LPDDR4 DRAM devices**
  2. **DRAM latency PUF** (88.2 ms) achieves a speedup of **102x/860x** at 70°C/55°C over prior DRAM PUF evaluation mechanisms

The DRAM Latency PUF:

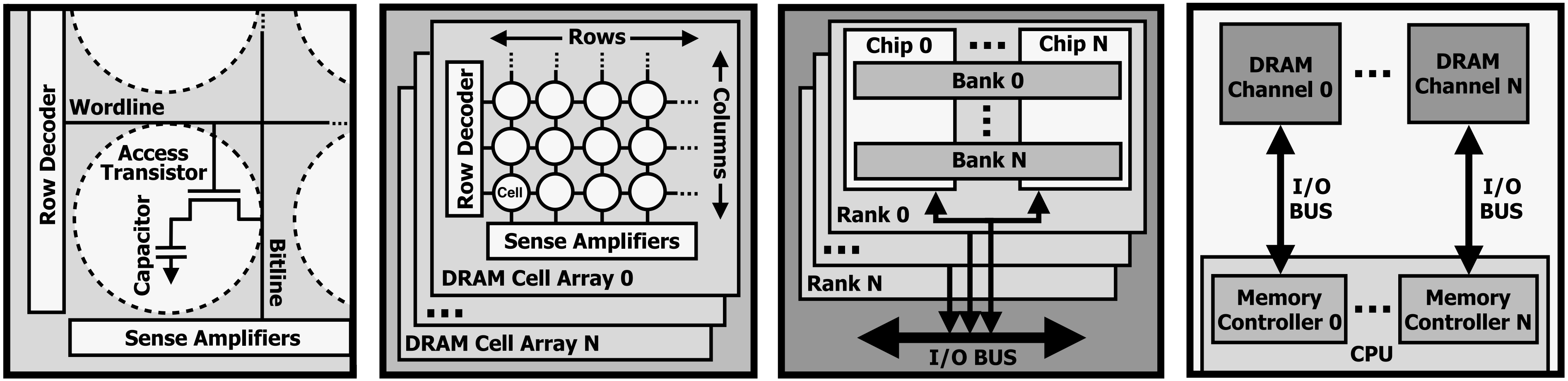
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim   Minesh Patel
Hasan Hassan   Onur Mutlu

Systems@ETH Zürich   SAFARI

ETH Zürich   Carnegie Mellon

DRAM Architecture Background



(a) DRAM Cell Array (b) DRAM Bank (c) DRAM Channel (d) DRAM-Based System

Evaluating DRAM Retention PUFs

Algorithm 1: Evaluate Retention PUF [103, 120, 121, 124, 135]

```
evaluate_DRAM_retention_PUF(seg_id, wait_time):
    rank_id ← DRAM rank containing seg_id
    disable refresh for Rank[rank_id]
    start_time ← current_time()
    while current_time() - start_time < wait_time:
        foreach row in Rank[rank_id]:
            if row not in Segment[seg_id]:
                issue refresh to row                    // refresh all other rows
    enable refresh for Rank[rank_id]
    return data at Segment[seg_id]
```

	#Chips	#Tested Memory Segments
A	91	17,408
B	65	12,544
C	67	10,580

Table 1: The number of tested PUF memory segments across the tested chips from each of the three manufacturers.

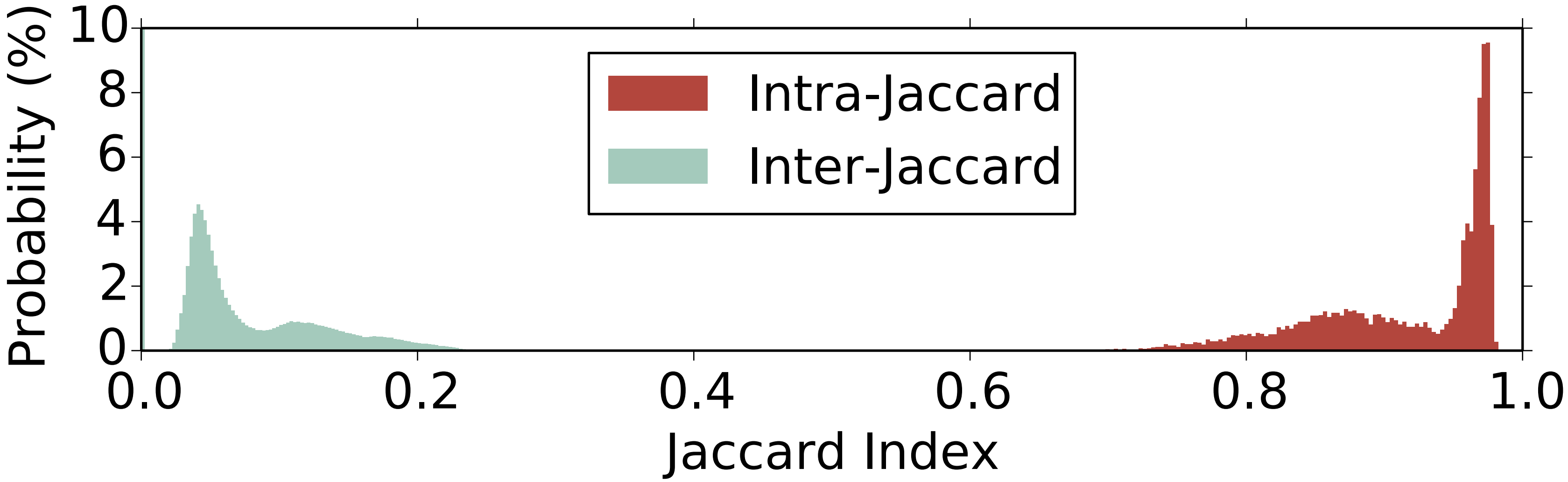


Figure 3: Distributions of Jaccard indices calculated across every possible pair of PUF responses across all tested PUF memory segments from each of 223 LPDDR4 DRAM chips.

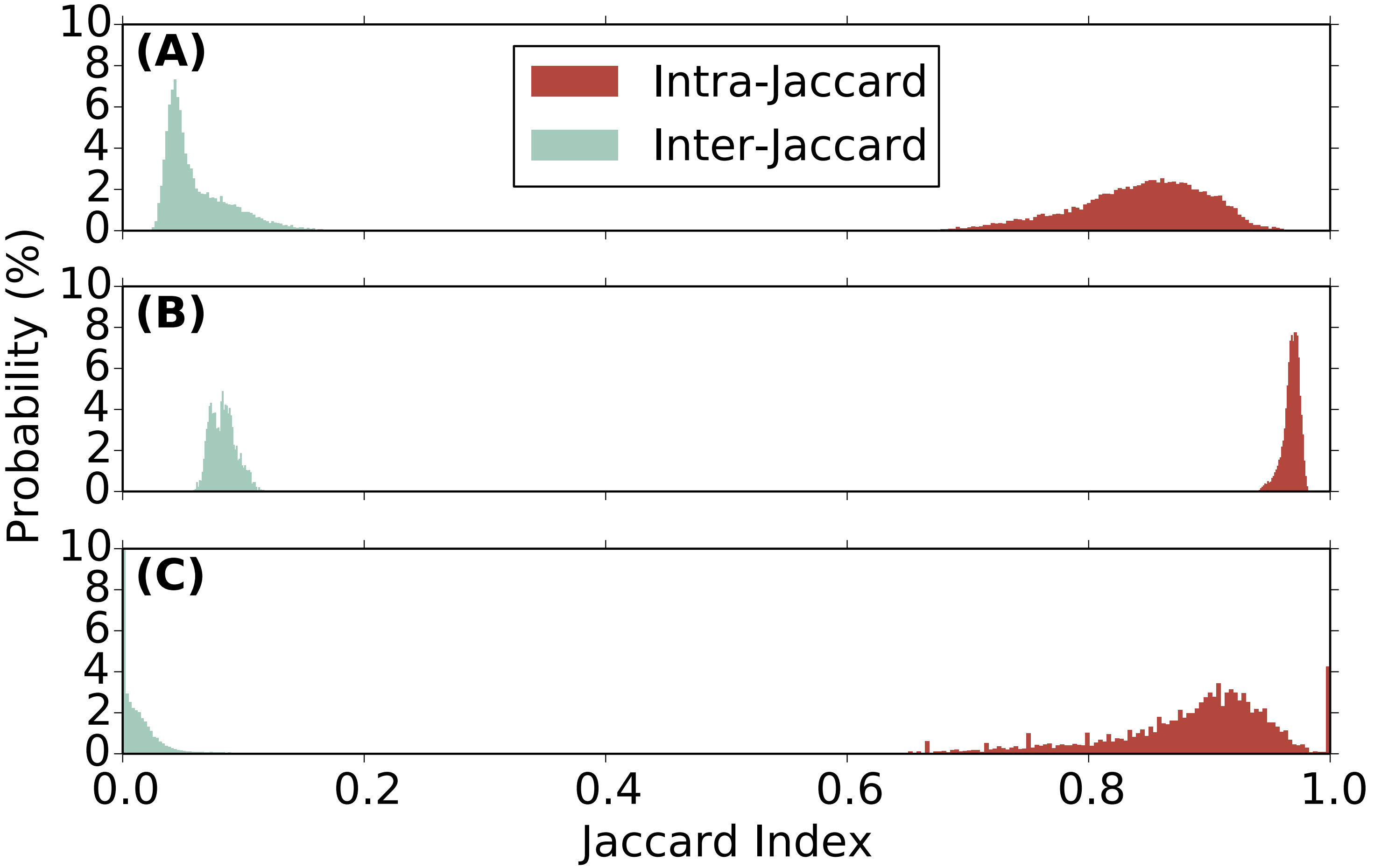


Figure 4: Distributions of Jaccard indices calculated between PUF responses of DRAM chips from a single manufacturer.

	#Chips	#Total Memory Segments
A	19	589,824
B	12	442,879
C	14	437,990

Table 2: Number of PUF memory segments tested for 30 days.

	%Memory Segments per Chip	
	Intra-Jaccard index range <0.1	Intra-Jaccard index range <0.2
A	100.00 [99.08, 100.00]	100.00 [100.00, 100.00]
B	90.39 [82.13, 99.96]	96.34 [95.37, 100.00]
C	95.74 [89.20, 100.00]	96.65 [95.48, 100.00]

Table 3: Percentage of PUF memory segments per chip with Intra-Jaccard index ranges <0.1 or 0.2 over a 30-day period. Median [minimum, maximum] values are shown.

Temperature Effects

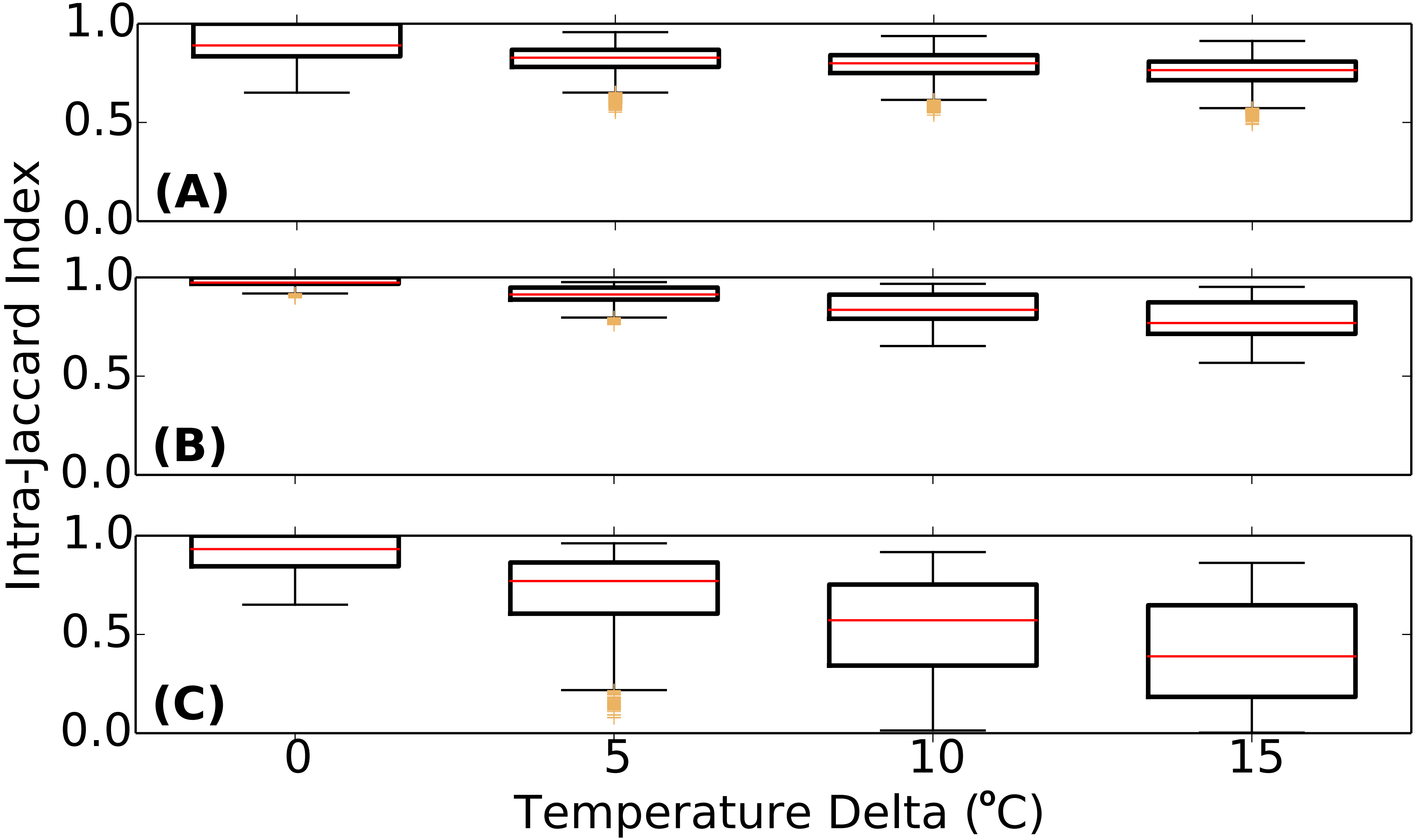


Figure 6: DRAM latency PUF repeatability vs. temperature.

Evaluating a DRAM Latency PUF

Algorithm 2: Evaluate DRAM latency PUF

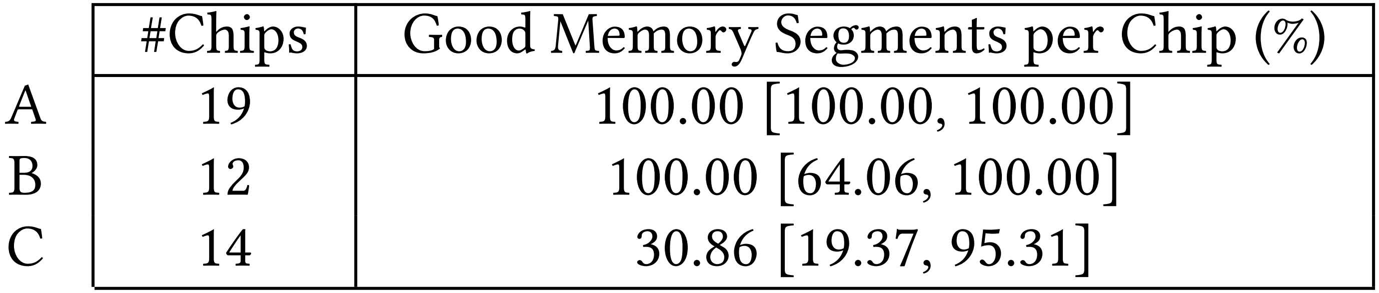
```
evaluate_DRAM_latency_PUF(seg_id):
    write known data (all 1's) to Segment[seg_id]
    rank_id ← DRAM rank containing seg_id
    obtain exclusive access to Rank[rank_id]
    set low $t_{RCD}$ for Rank[rank_id]
    for i = 1 to num_iterations :
        for col in Segment[seg_id]
            for row in Segment[seg_id]:        // column-order reads
                read()                         // induce read failures
                memory_barrier()               // one access at a time
                count_failures()               // record in another rank
    set default $t_{RCD}$ for Rank[rank_id]
    filter the PUF memory segment              // See Filtering Mechanism
    release exclusive access to Rank[rank_id]
    return error pattern at Segment[seg_id]
```

Memory Footprint. Equation 2 provides the memory footprint required by PUF evaluation:

$$mem_{total} = (size_{mem\_seg}) + (size_{counter\_buffer}) \tag{2}$$

where $size_{mem\_seg}$ is the size of the PUF memory segment and $size_{counter\_buffer}$ is the size of the counter buffer. The size of the counter buffer can be calculated using Equation 3:

$$size_{counter\_buffer} = (size_{mem\_seg}) \times \lceil \log_2 N_{iters} \rceil \tag{3}$$



	#Chips	Good Memory Segments per Chip (%)
A	19	100.00 [100.00, 100.00]
B	12	100.00 [64.06, 100.00]
C	14	30.86 [19.37, 95.31]

Table 4: Percentage of *good* memory segments per chip across manufacturers. Median [min, max] values are shown.

DRAM Characterization

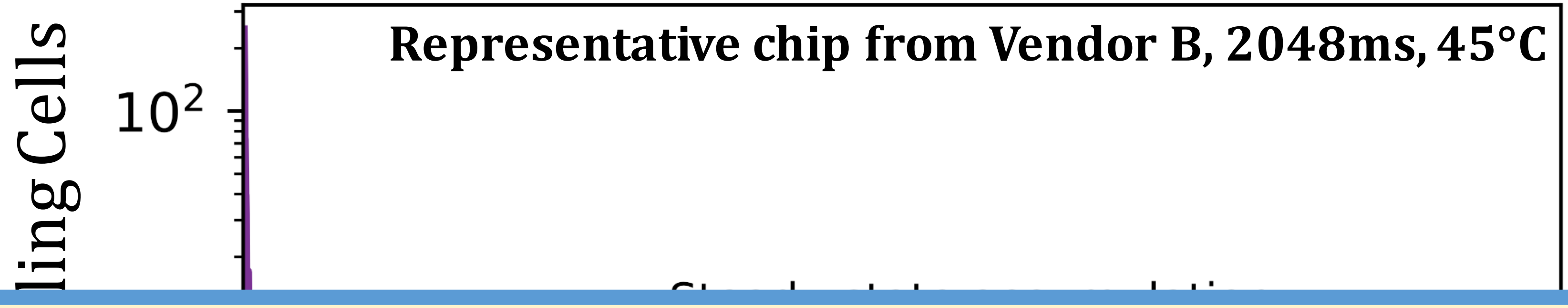
Sources of Retention Time Variation

- **Process/voltage/temperature**

- **Data pattern dependence (DPD)**
  - Retention times **change with data** in cells/neighbors
  - e.g., all 1's vs. all 0's

- **Variable retention time (VRT)**
  - Retention time changes **randomly (unpredictably)**
  - Due to a combination of various circuit effects

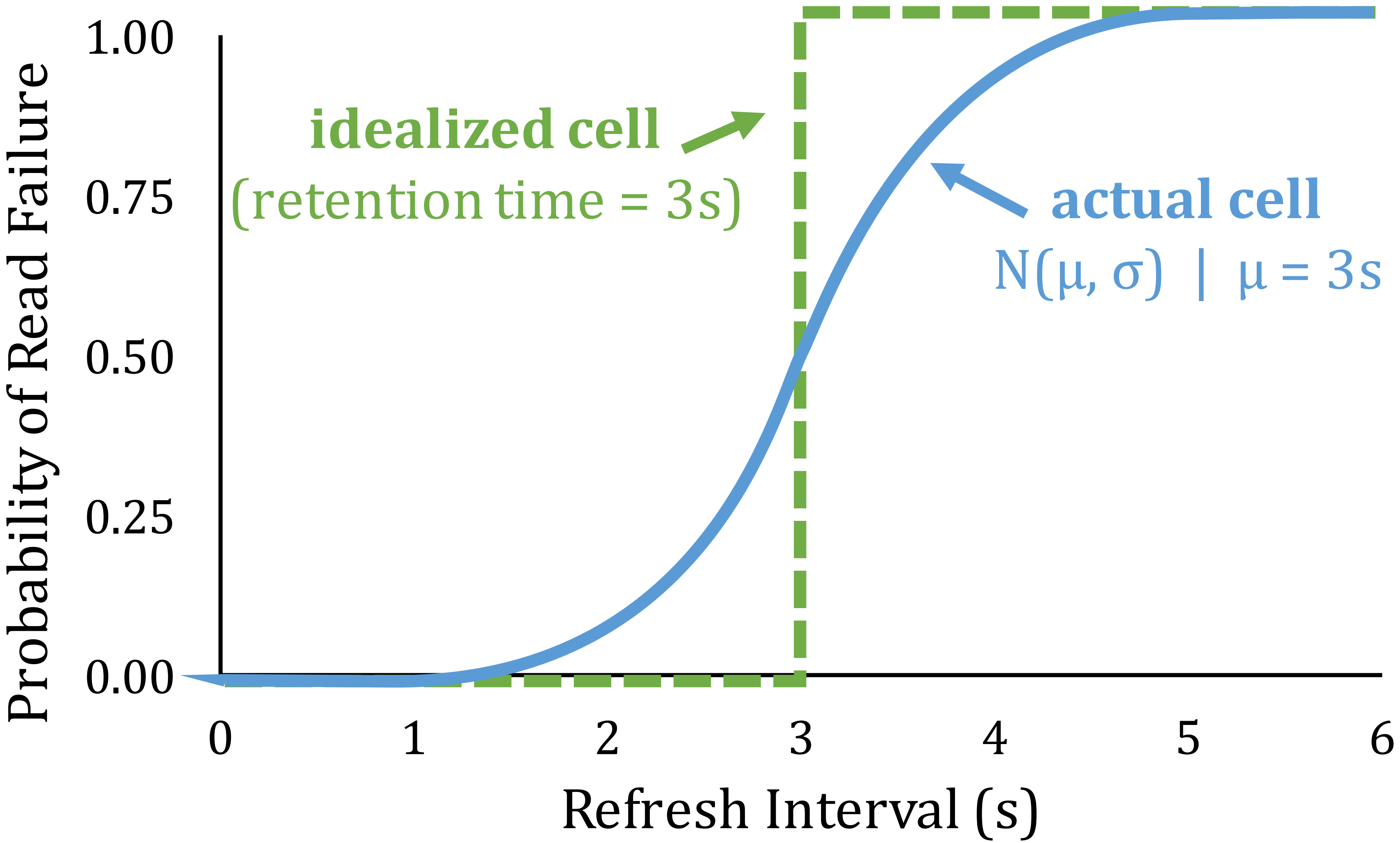
Long-term Continuous Profiling



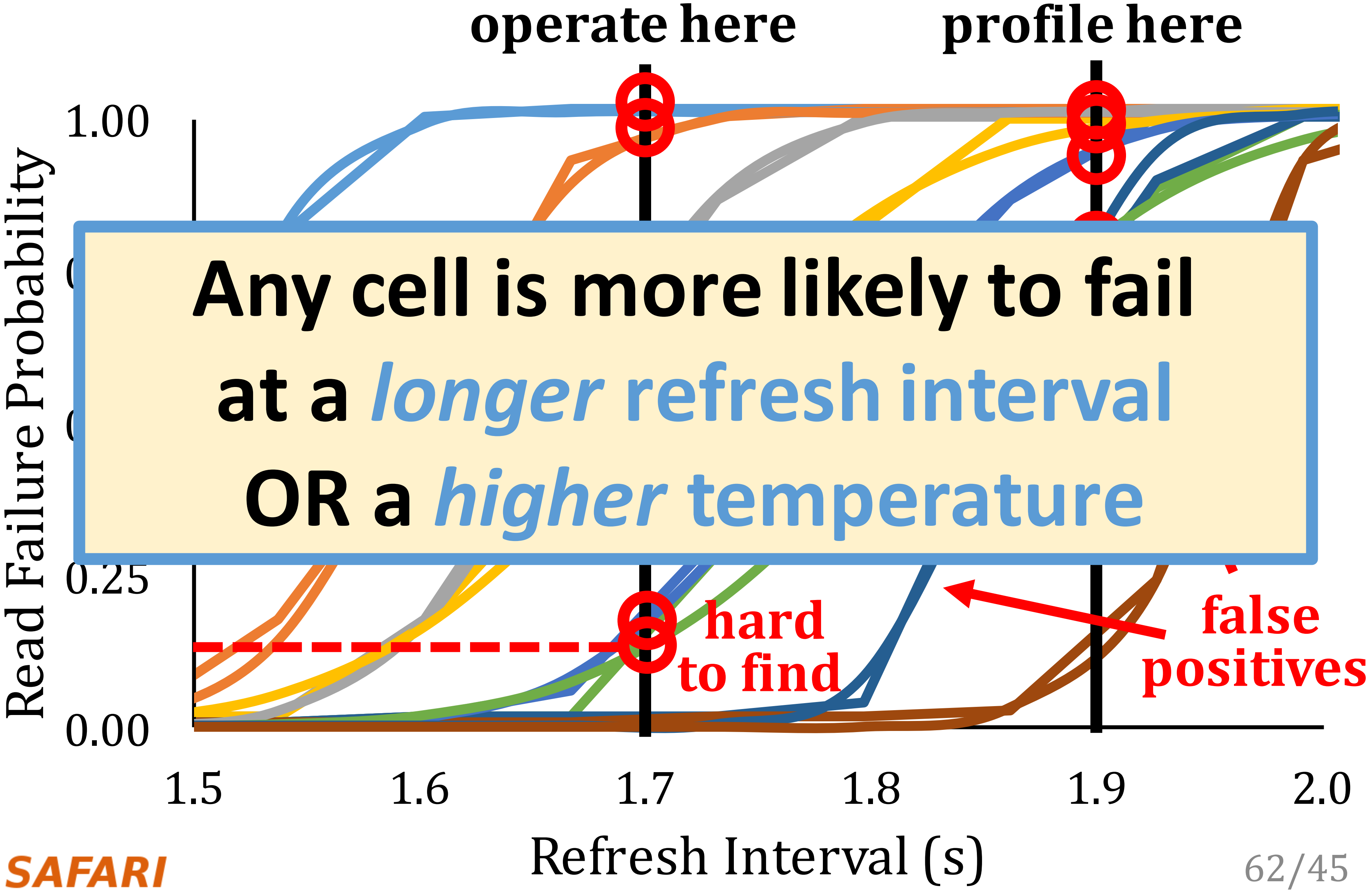
Error correction codes (ECC) and online profiling are *necessary* to manage new failing cells

- New failing cells continue to appear over time
    - Attributed to **variable retention time (VRT)**
- The set of failing cells changes over time

Single-cell Failure Probability (Cartoon)



Single-cell Failure Probability (Real)



Any cell is more likely to fail at a *longer* refresh interval OR a *higher* temperature

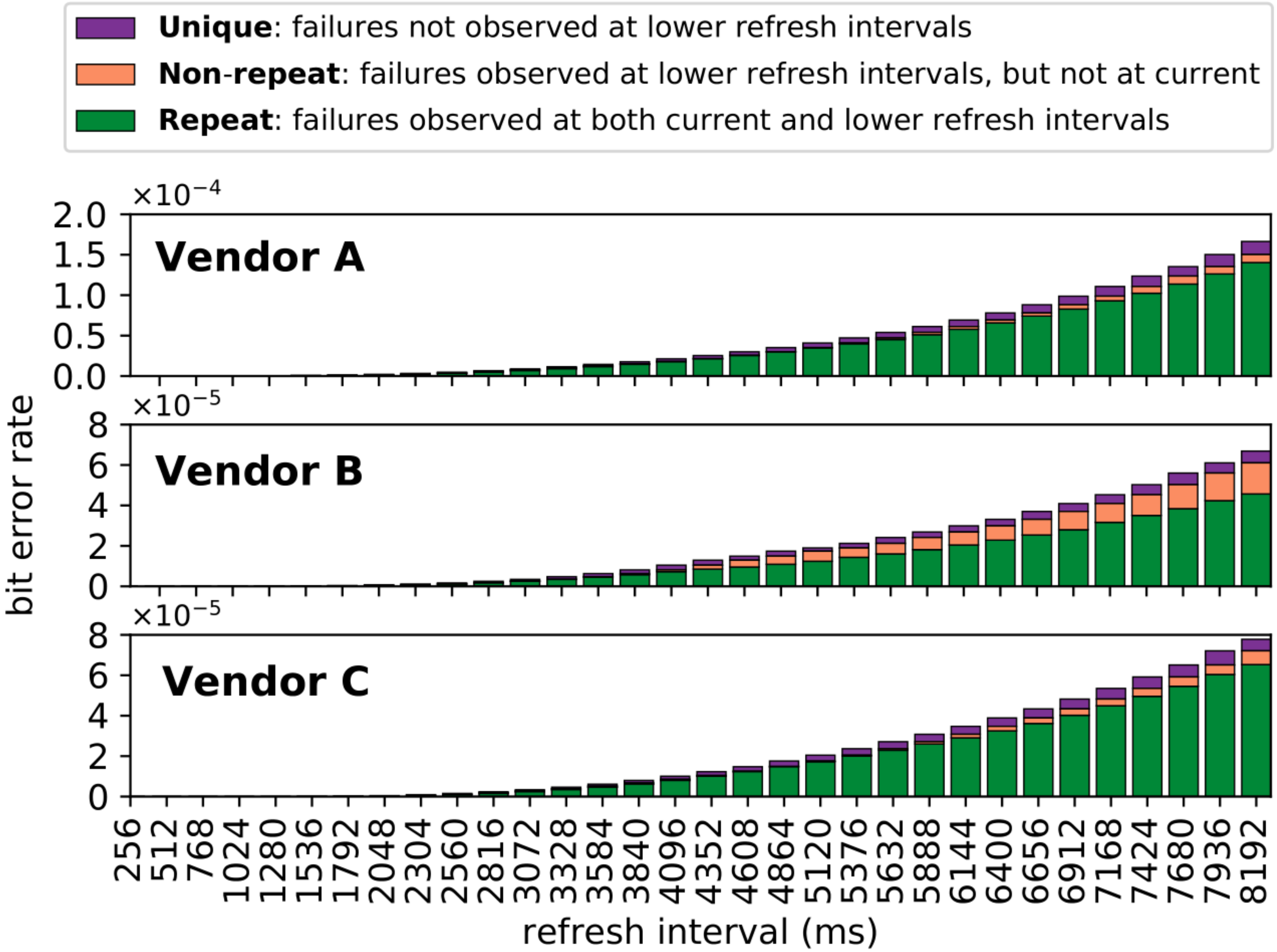
Temperature Relationship

- Well-fitting exponential relationship:

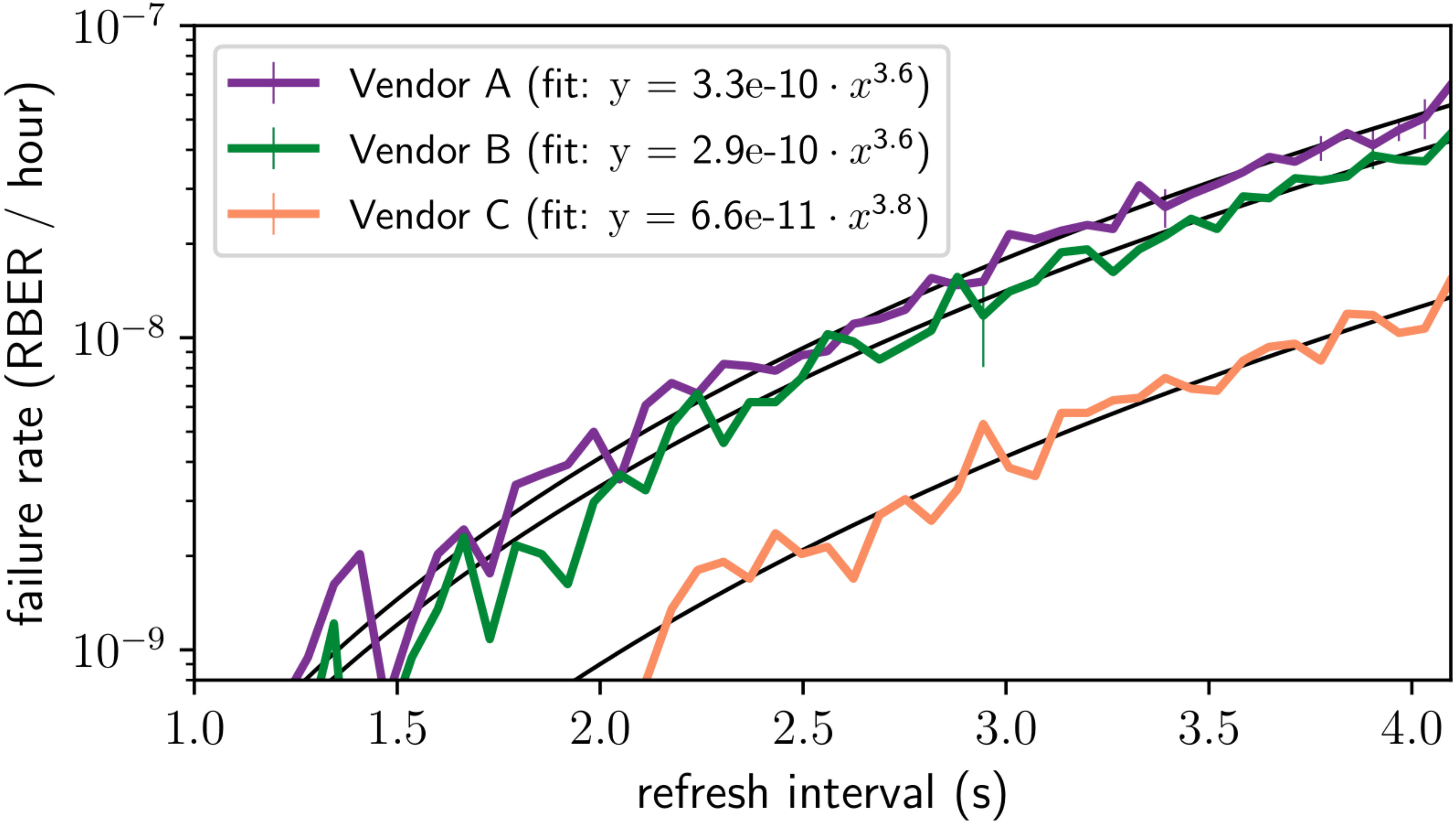
$$R_A \propto e^{0.22\Delta T} \qquad R_B \propto e^{0.20\Delta T} \qquad R_C \propto e^{0.26\Delta T}$$

- E.g., 10°C ~ 10x more failures

Retention Failures @ 45°C

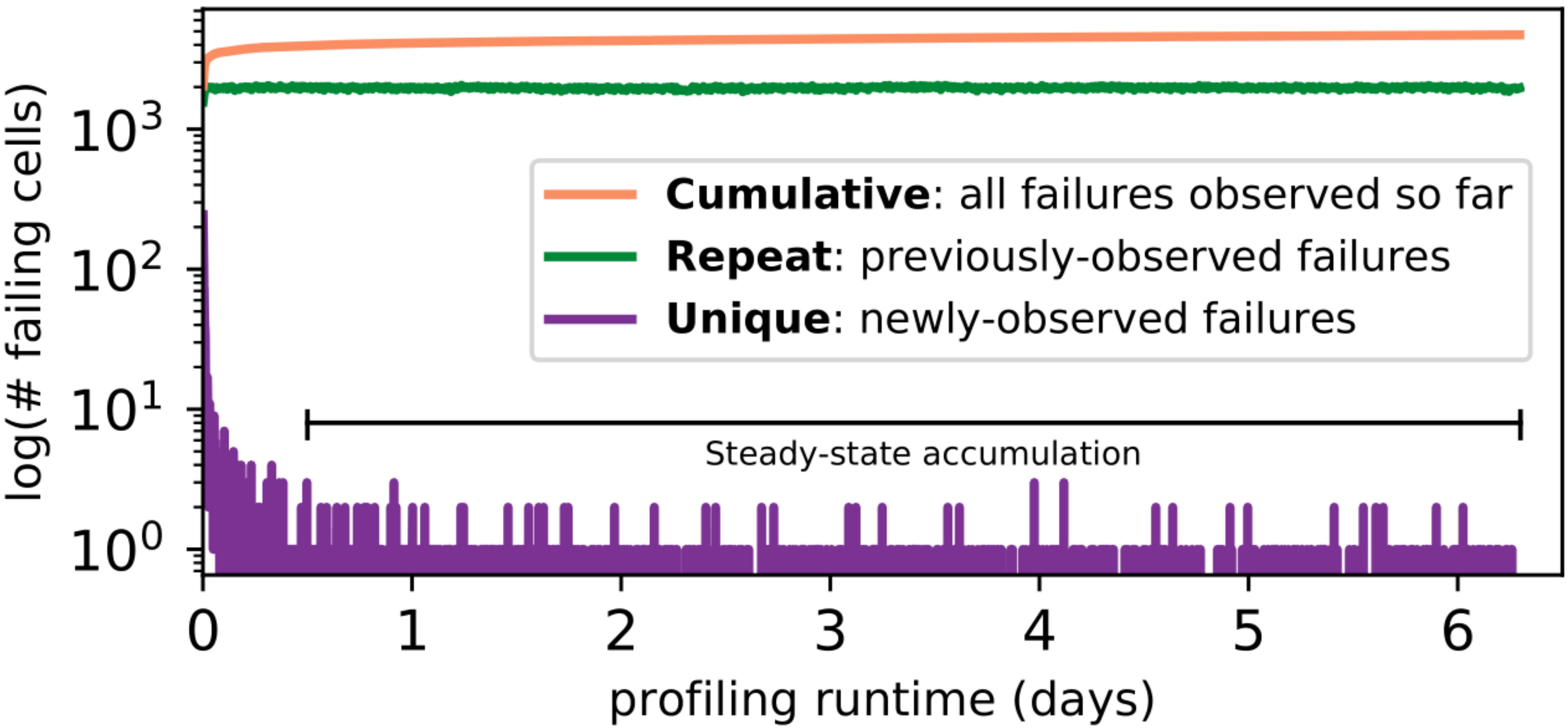


VRT Failure Accumulation Rate

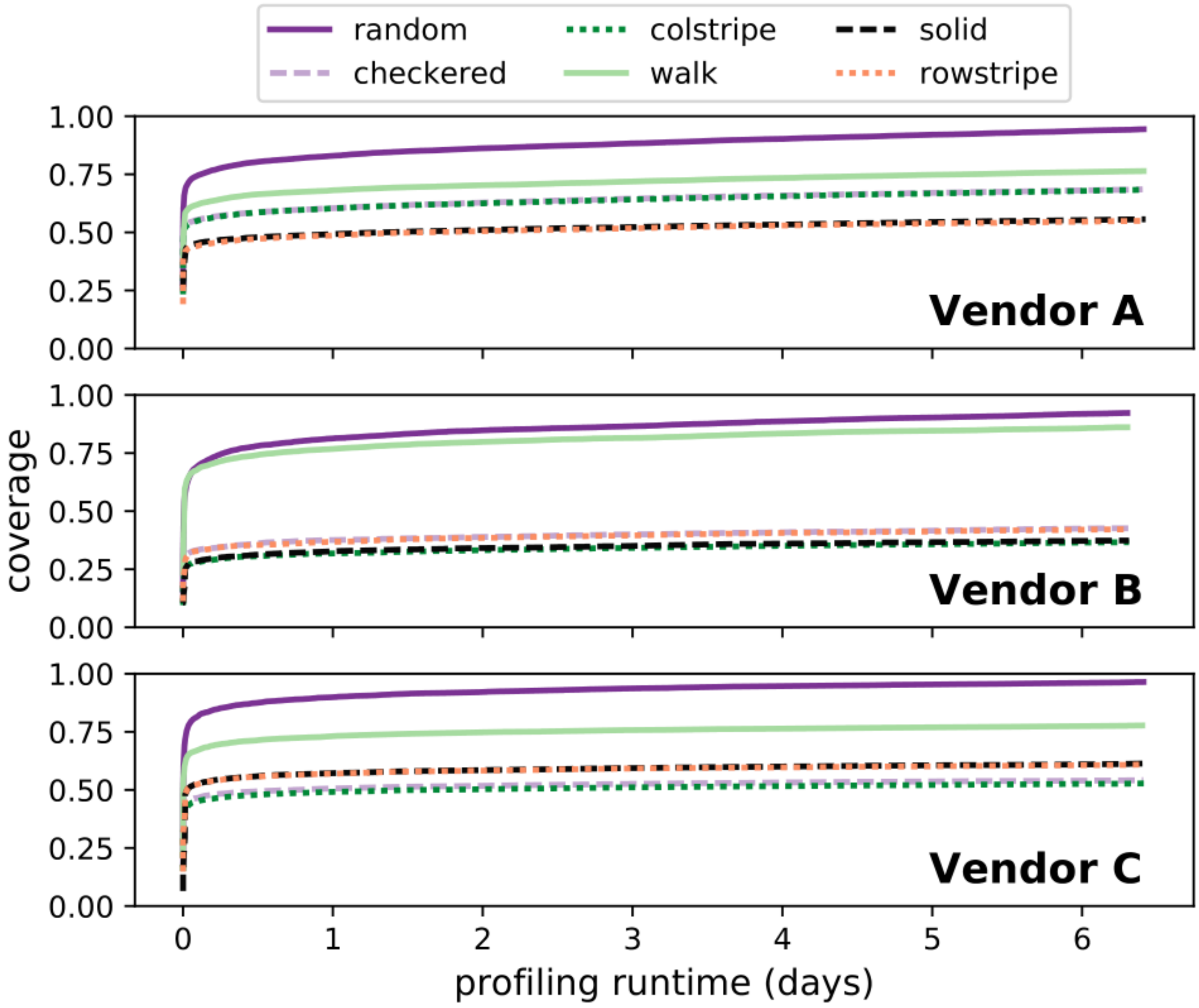


SAFARI

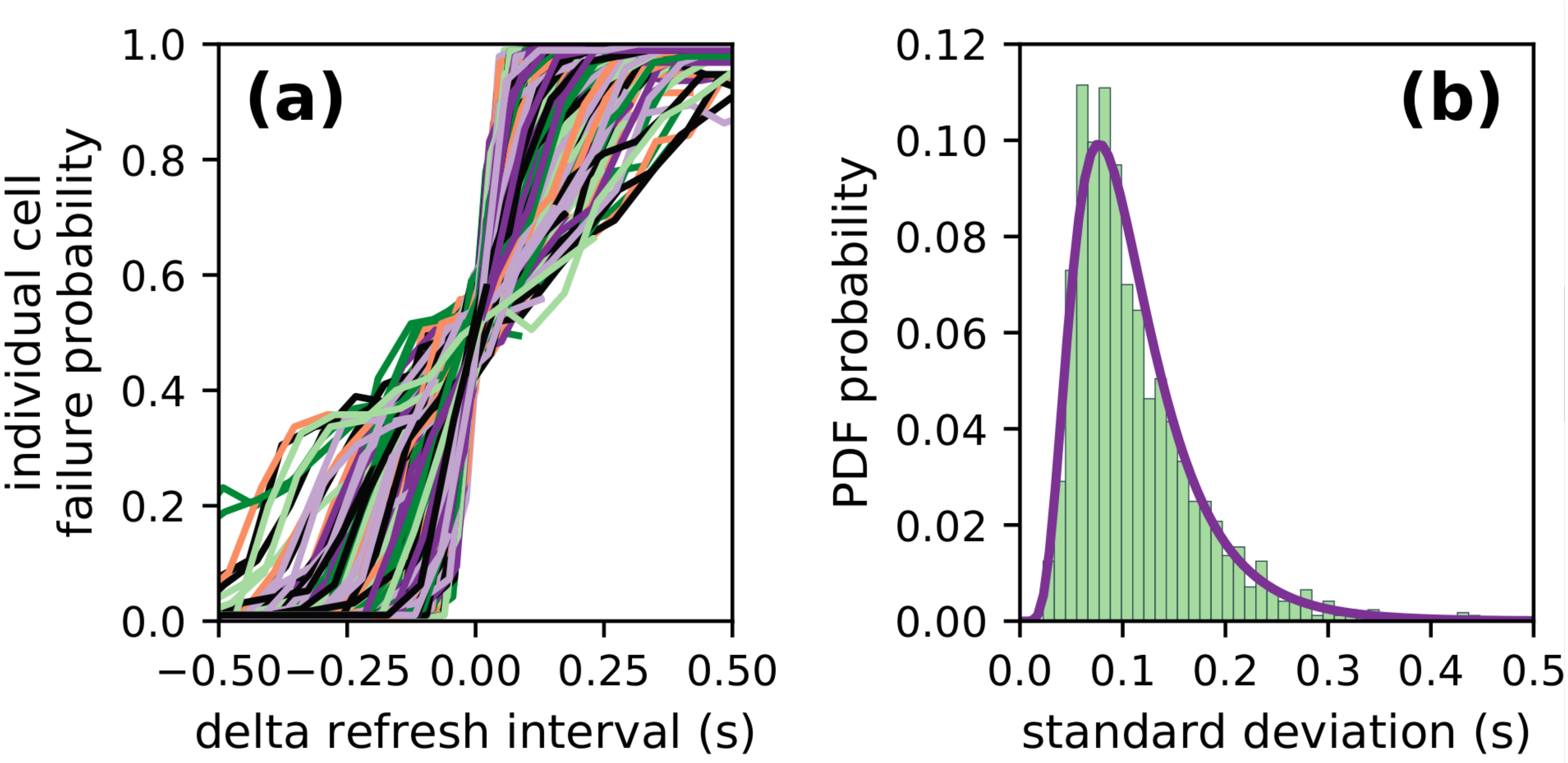
800 Rounds of Profiling @ 2048ms, 45°C



800 Rounds of Profiling @ 2048ms, 45°C

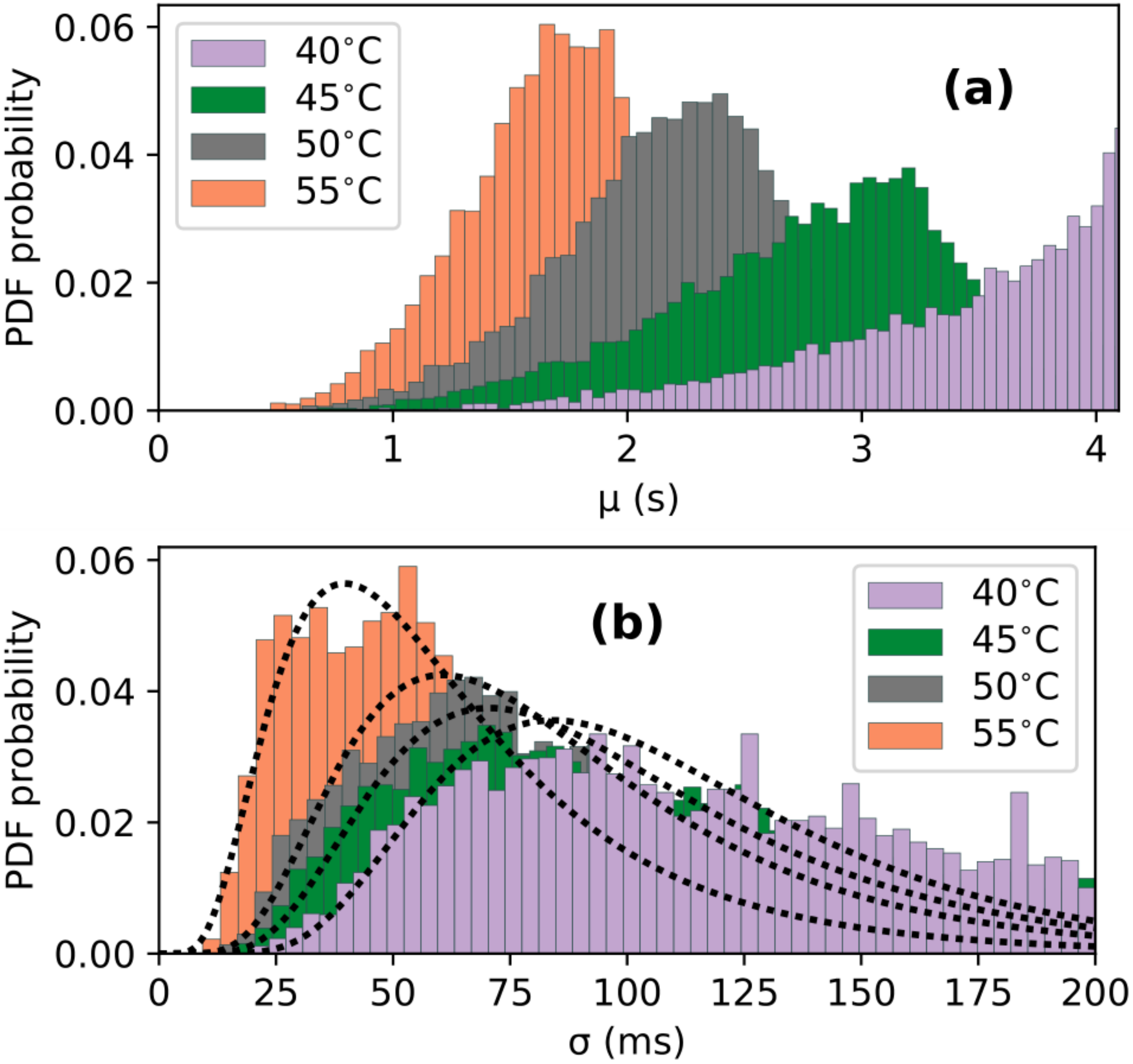


Individual Cell Failure Probabilities

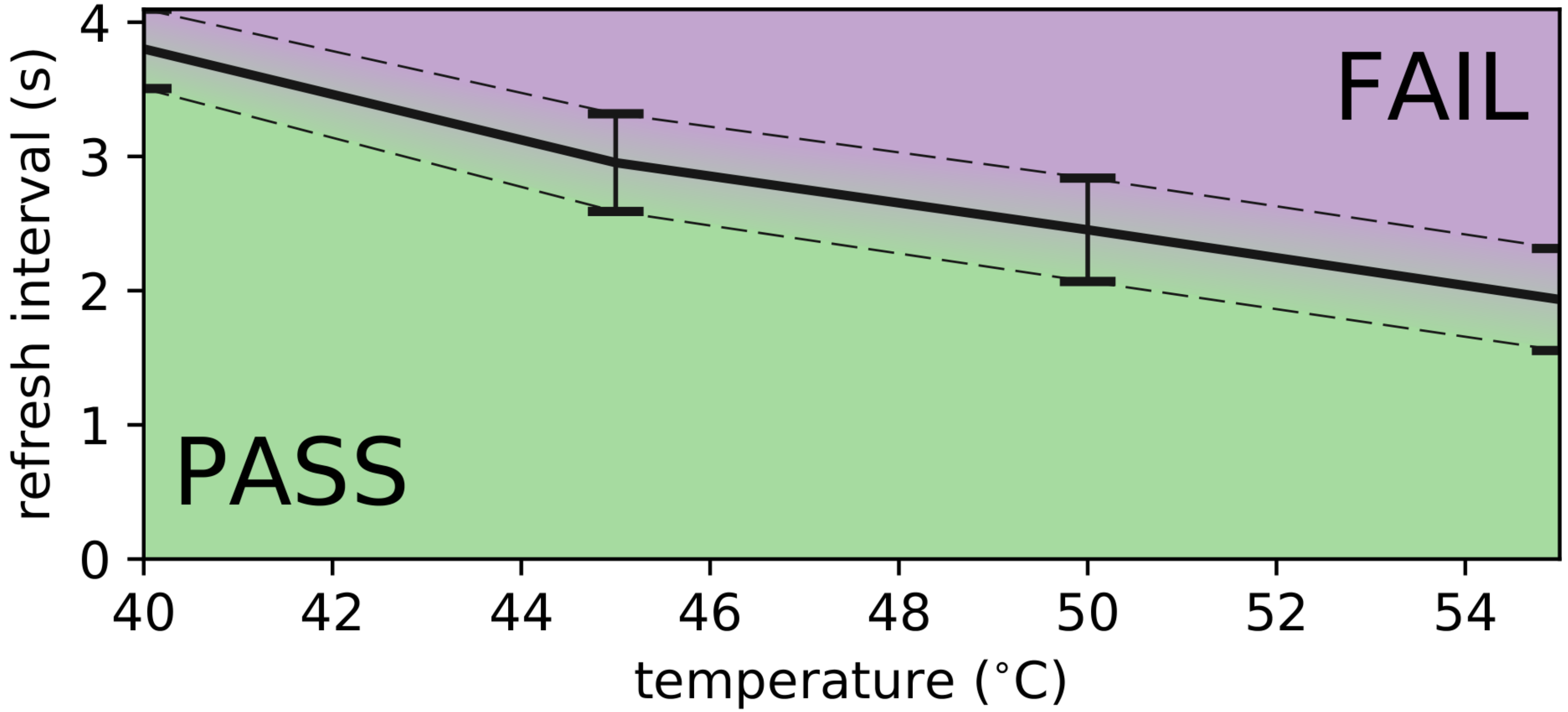


- Single representative chip of Vendor B at 40° C
- Refresh intervals ranging from 64ms to 4096ms

Individual Cell Failure Distributions



Single-cell Failures With Temperature



- Single representative chip of Vendor B
- {mean, std} for cells between 64ms and 4096ms

SAFARI