Authenticache: Harnessing Cache ECC for System Authentication

Anys Bacha and Radu Teodorescu

Department of Computer Science and Engineering

The Ohio State University

http://arch.cse.ohio-state.edu

THE OHIO STATE UNIVERSITY

COMPUTER ARCHITECTURE RESEARCH LAB

Security and Everyday Computing

Security is now crucial to all computing markets, especially with the advent of IoT

Security Challenges

Security Challenges

- Password management
  - Complexity due to different accounts having policies

Security Challenges

- Password management
  - Complexity due to different accounts having policies

- Secure key storage
  - Increases complexity for low cost IoT devices

Security Challenges

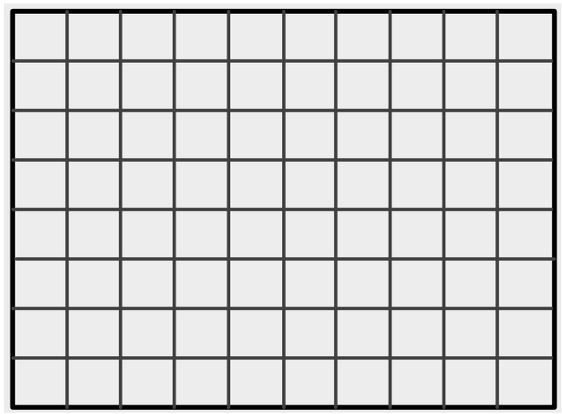
- Password management
  - Complexity due to different accounts having policies

- Secure key storage
  - Increases complexity for low cost IoT devices

- Software as a Service
  - Personal device at workplace increasing security risks

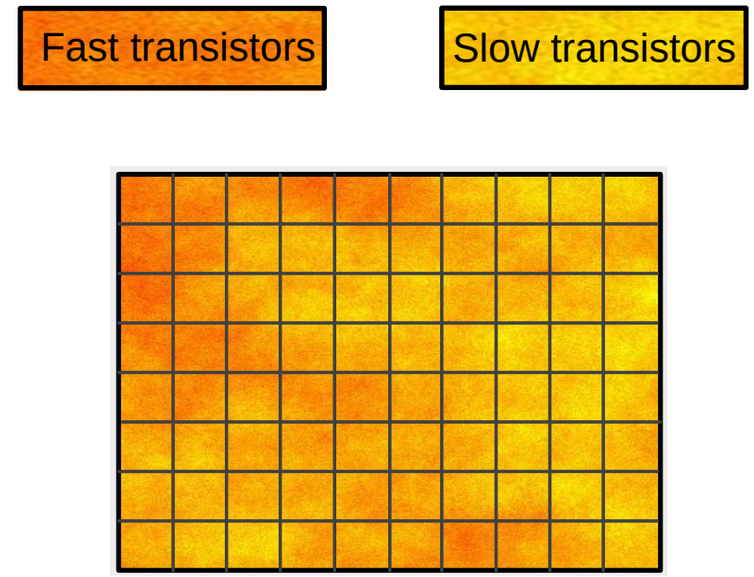
Physical Unclonable Functions (PUF)

Physical Unclonable Functions (PUF)



Chip

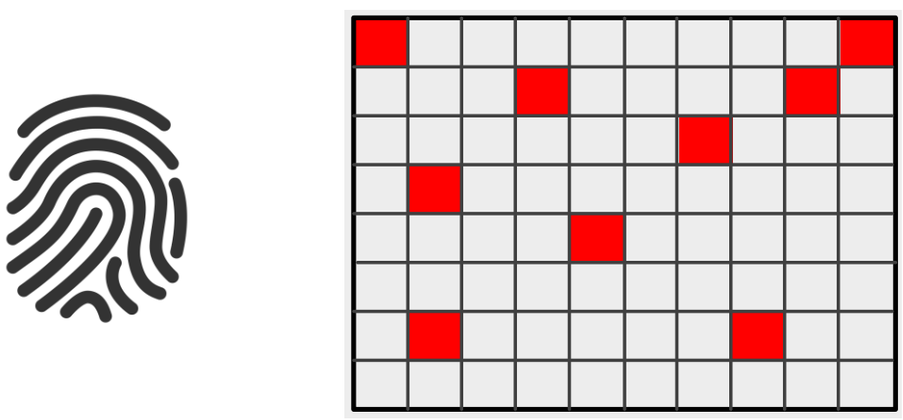
Physical Unclonable Functions (PUF)



Chip

- Exploit randomness in silicon

Physical Unclonable Functions (PUF)



Chip

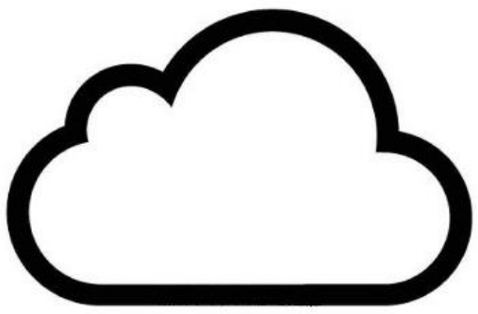
Silicon Fingerprints

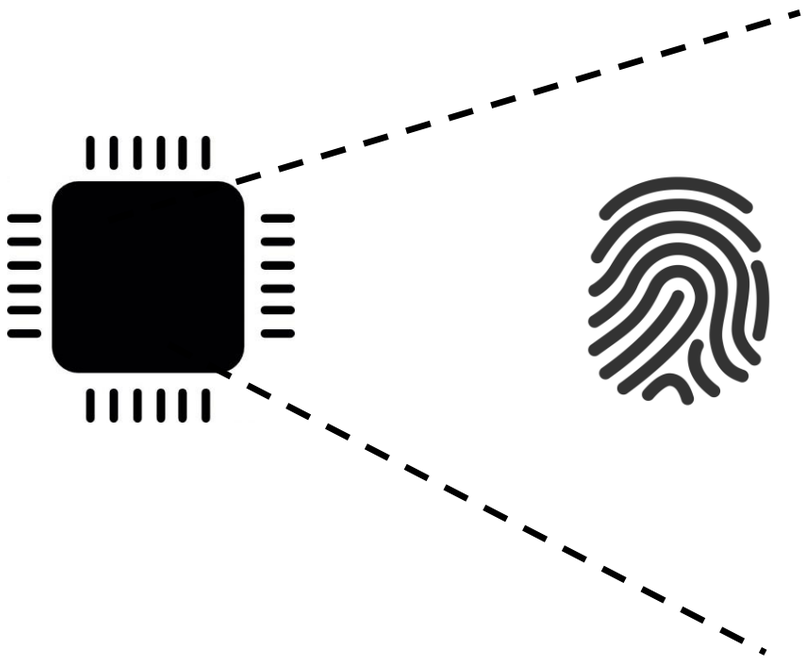
- Exploit randomness in silicon
- Systematic outputs unique to device

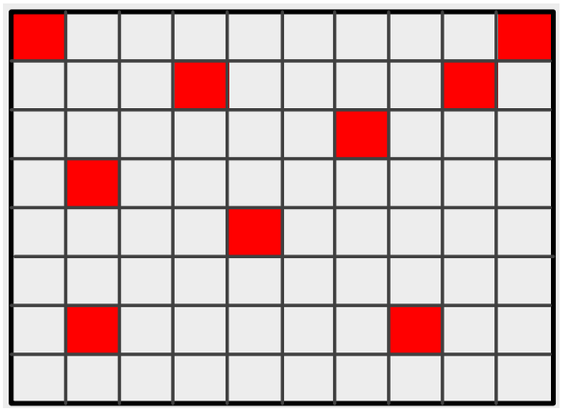
PUF System Authentication

PUF System Authentication

Enrollment



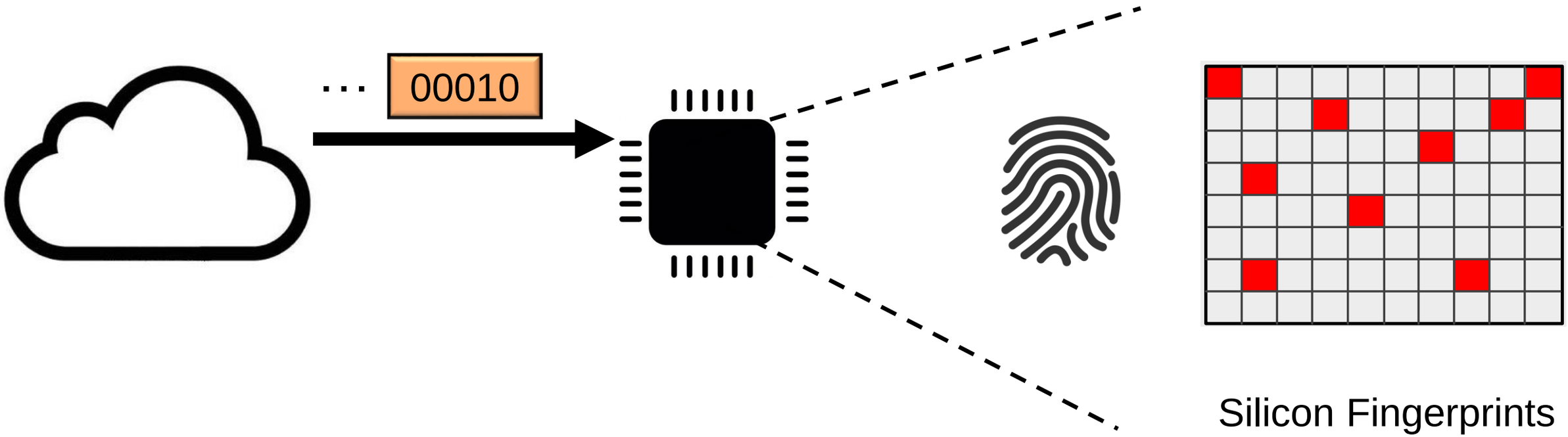




Silicon Fingerprints

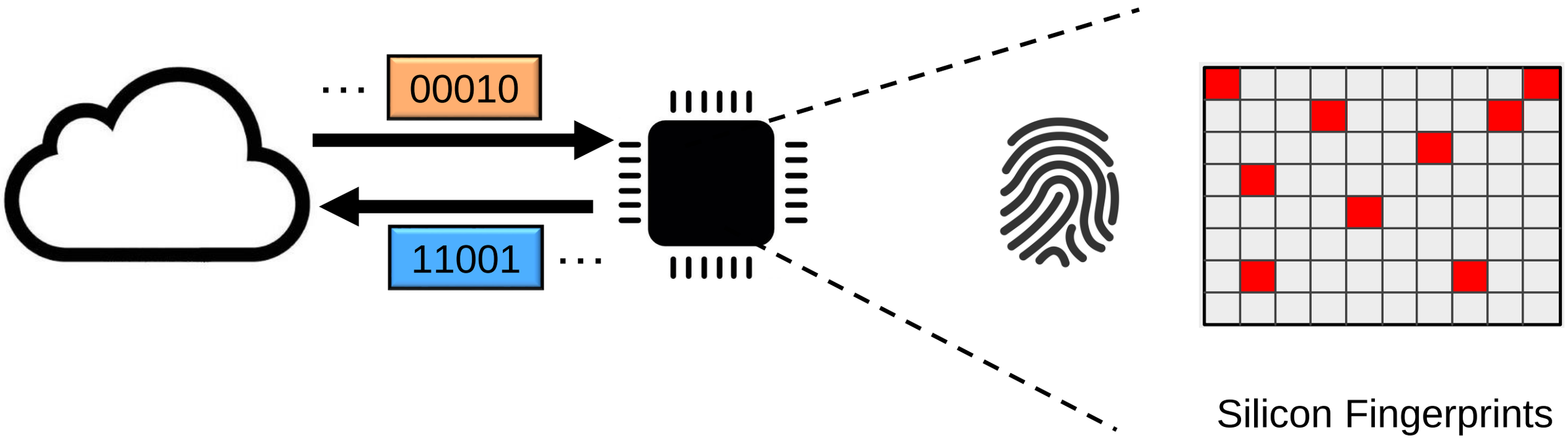
PUF System Authentication

Enrollment



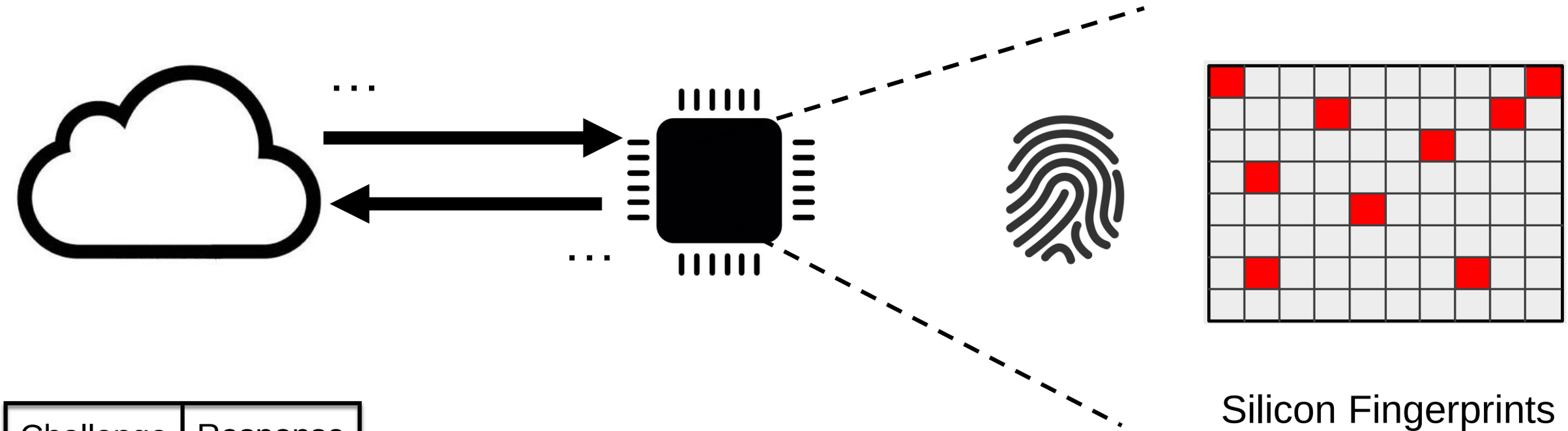
PUF System Authentication

Enrollment



PUF System Authentication

Enrollment

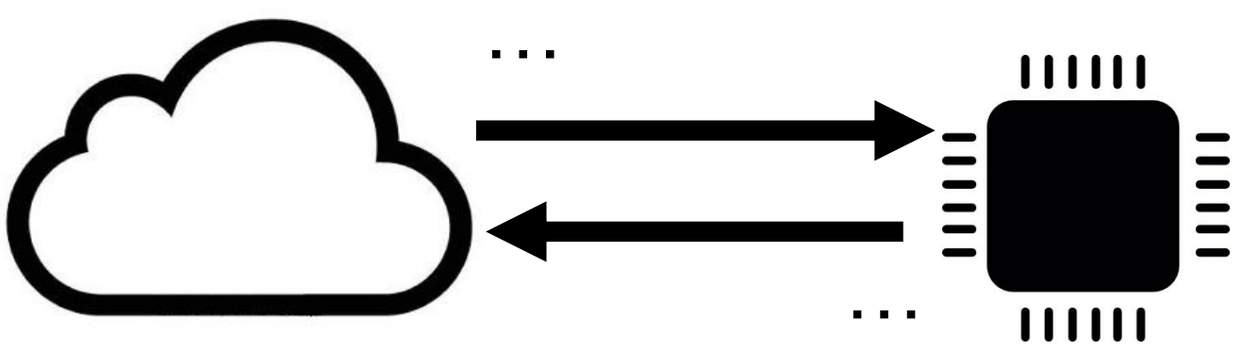


Challenge	Response
01100	00110
…	…
00010	11001

Silicon Fingerprints

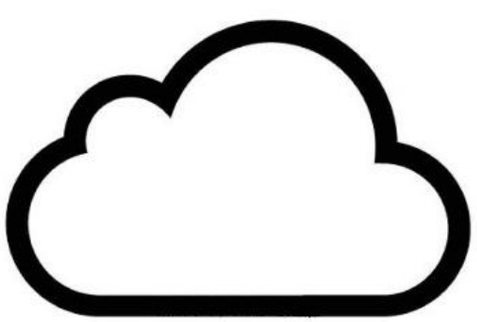
PUF System Authentication

Enrollment

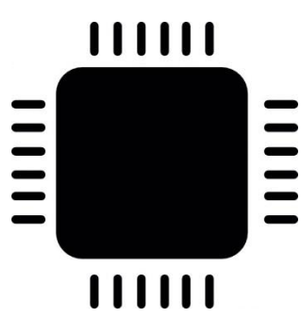


Challenge	Response
01100	00110
…	…
00010	11001

Authentication

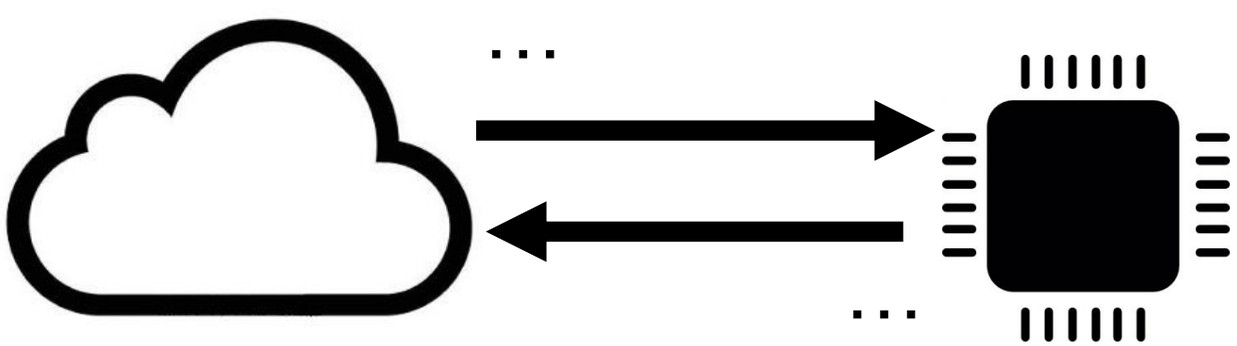


Challenge	Response
01100	00110
…	…
00010	11001



PUF System Authentication

Enrollment



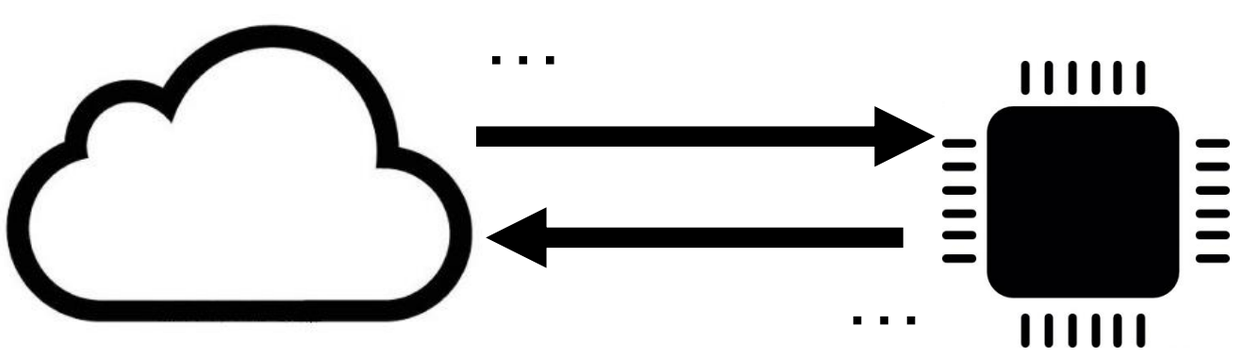
Challenge	Response
01100	00110
…	…
00010	11001

Authentication

Challenge	Response
01100	00110
…	…
00010	11001

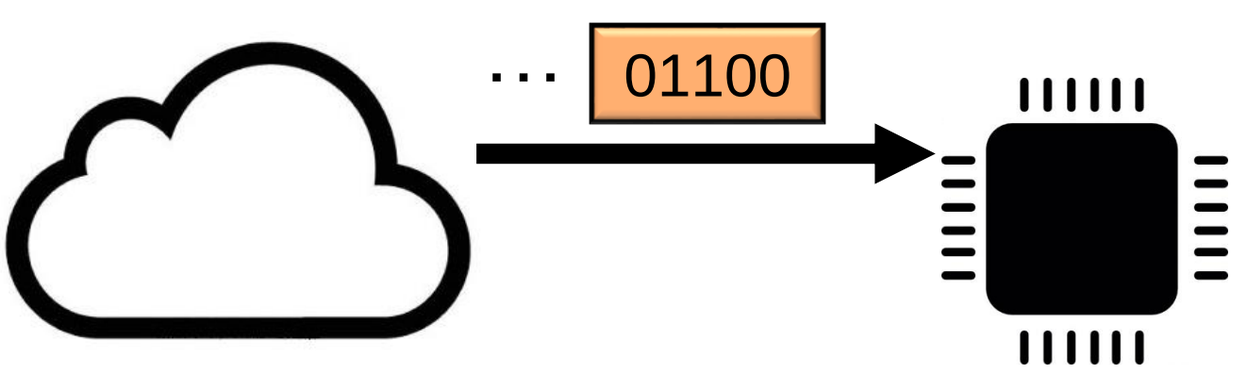
PUF System Authentication

Enrollment



Challenge	Response
01100	00110
…	…
00010	11001

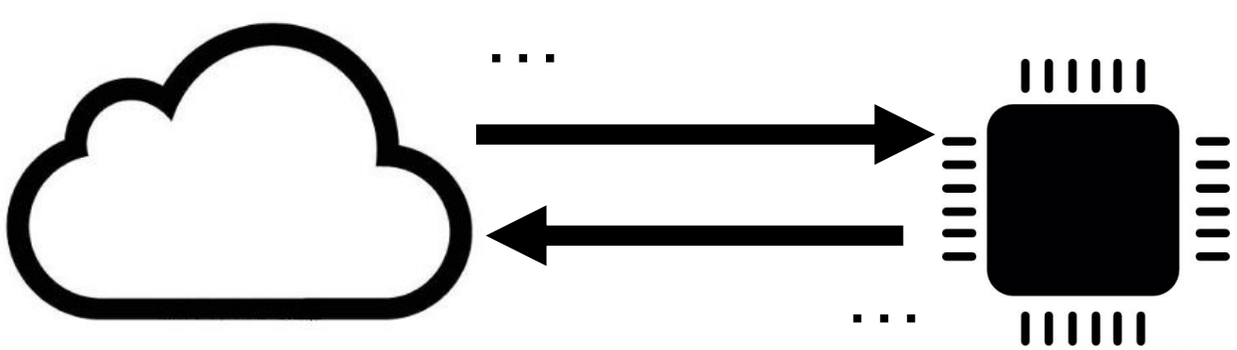
Authentication



Challenge	Response
01100	00110
…	…
00010	11001

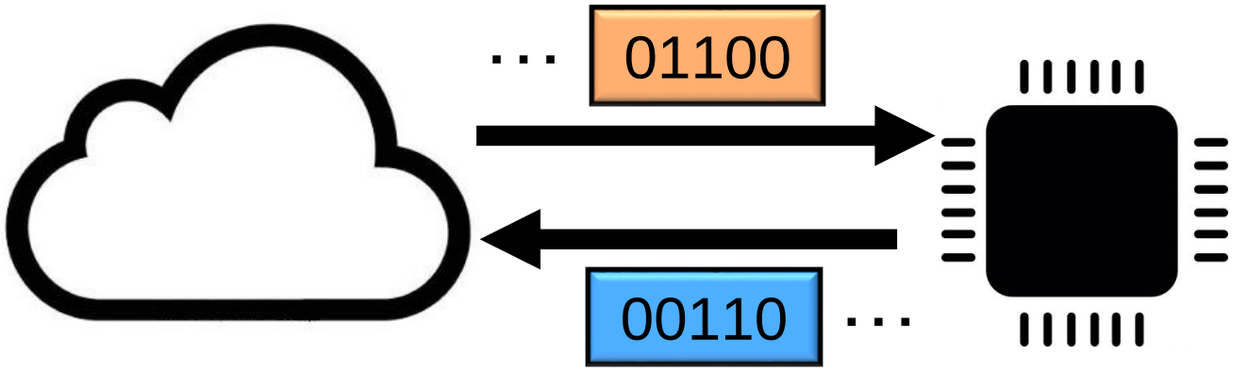
PUF System Authentication

Enrollment



Challenge	Response
01100	00110
…	…
00010	11001

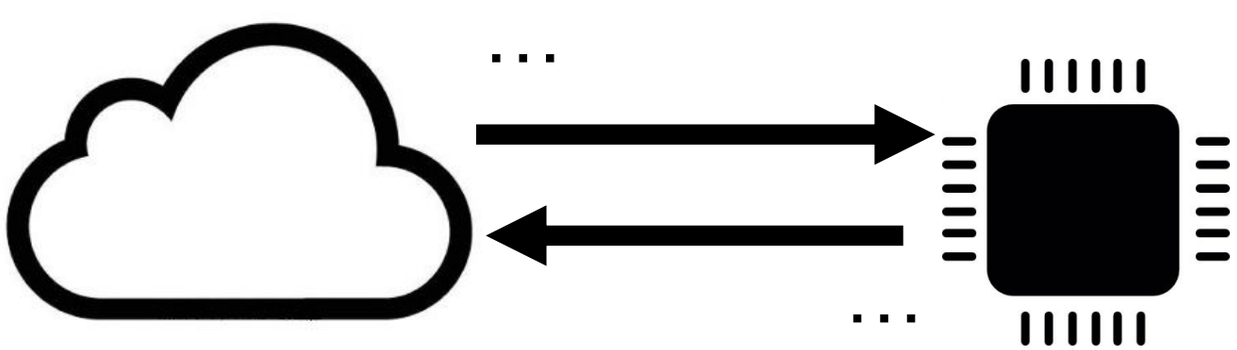
Authentication



Challenge	Response
01100	00110
…	…
00010	11001

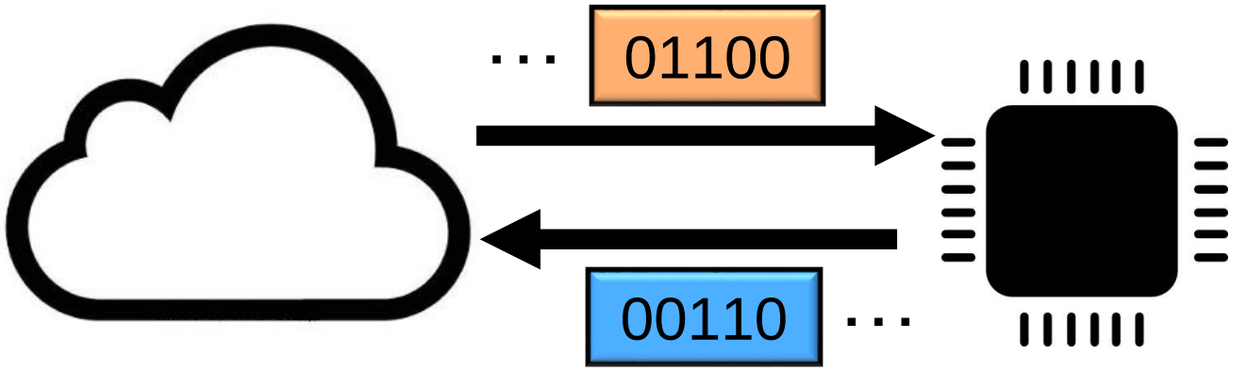
PUF System Authentication

Enrollment



Challenge	Response
01100	00110
…	…
00010	11001

Authentication

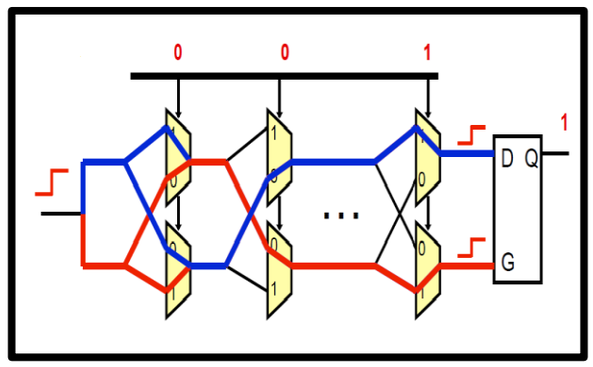


Challenge	Response
01100	00110
…	…
00010	11001

Related Work

Related Work

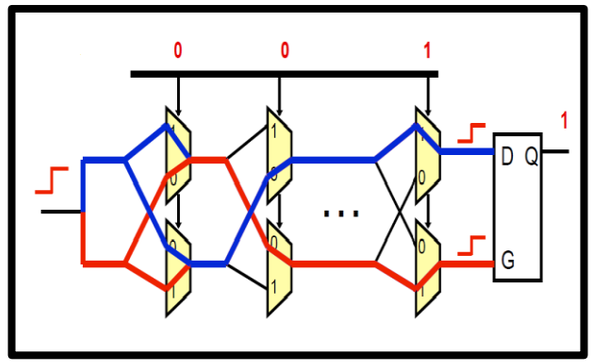
- Arbiter PUF (Lee et al. VLSI'04)
  - Signal traversing maze of cascaded switch blocks



Aribiter PUF

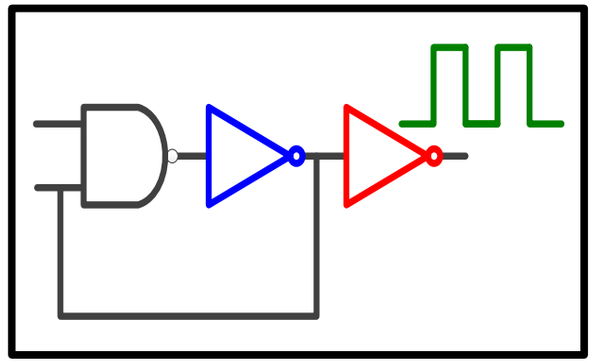
Related Work

- Arbiter PUF (Lee et al. VLSI'04)
  - Signal traversing maze of cascaded switch blocks



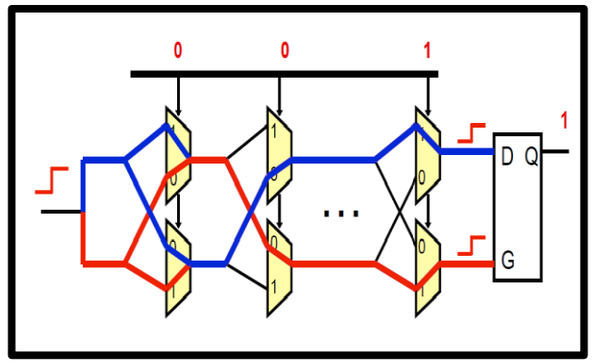
Aribiter PUF

- Ring Oscillator PUF (Suh et al. DAC'07)
  - Delay loops feeding oscillations into counters


Ring Oscillator PUF

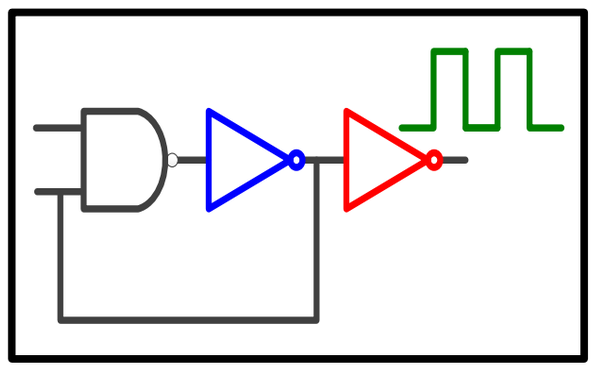
Related Work

- Arbiter PUF (Lee et al. VLSI'04)
  - Signal traversing maze of cascaded switch blocks

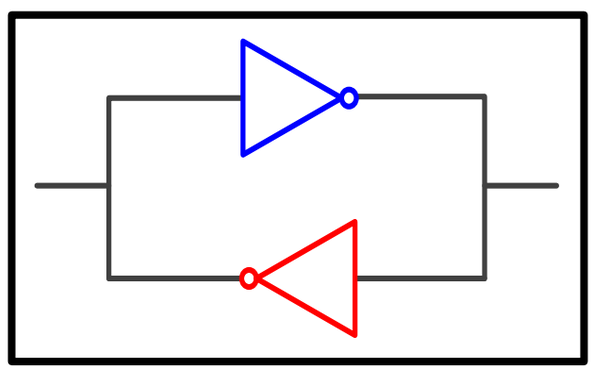


Aribiter PUF

- Ring Oscillator PUF (Suh et al. DAC'07)
  - Delay loops feeding oscillations into counters

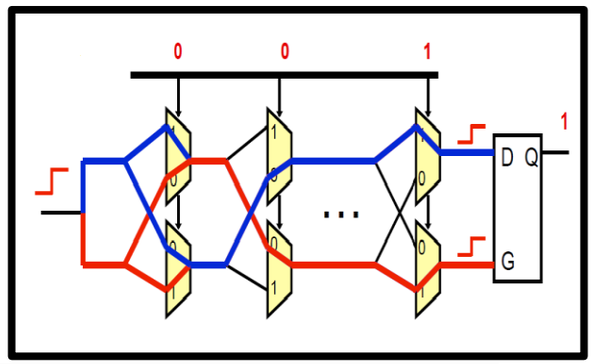

Ring Oscillator PUF

- SRAM PUF (Guajardo et al. CHES'07)
  - Power-on states of 6T SRAM cell


SRAM PUF

Related Work

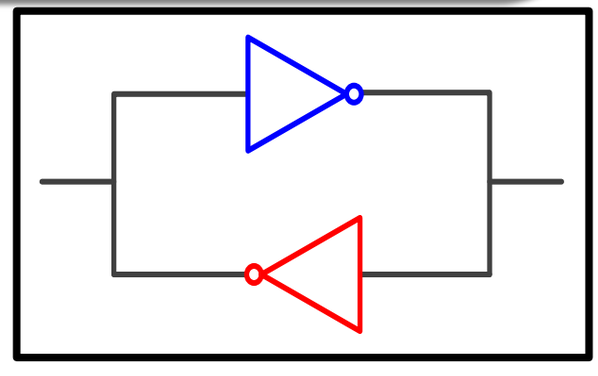
- Arbiter PUF (Lee et al. VLSI'04)
  - Signal traversing maze of cascaded switch blocks



Authenticache: No custom hardware

On-chip error correction logic in processor caches

- SRAM PUF (Guajardo et al. CHES'07)
  - Power-on states of 6T SRAM cell



SRAM PUF

Cache Errors as Silicon Fingerprints

Cache Errors as Silicon Fingerprints

- Caches optimized for density

Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation

Cache Errors as Silicon Fingerprints

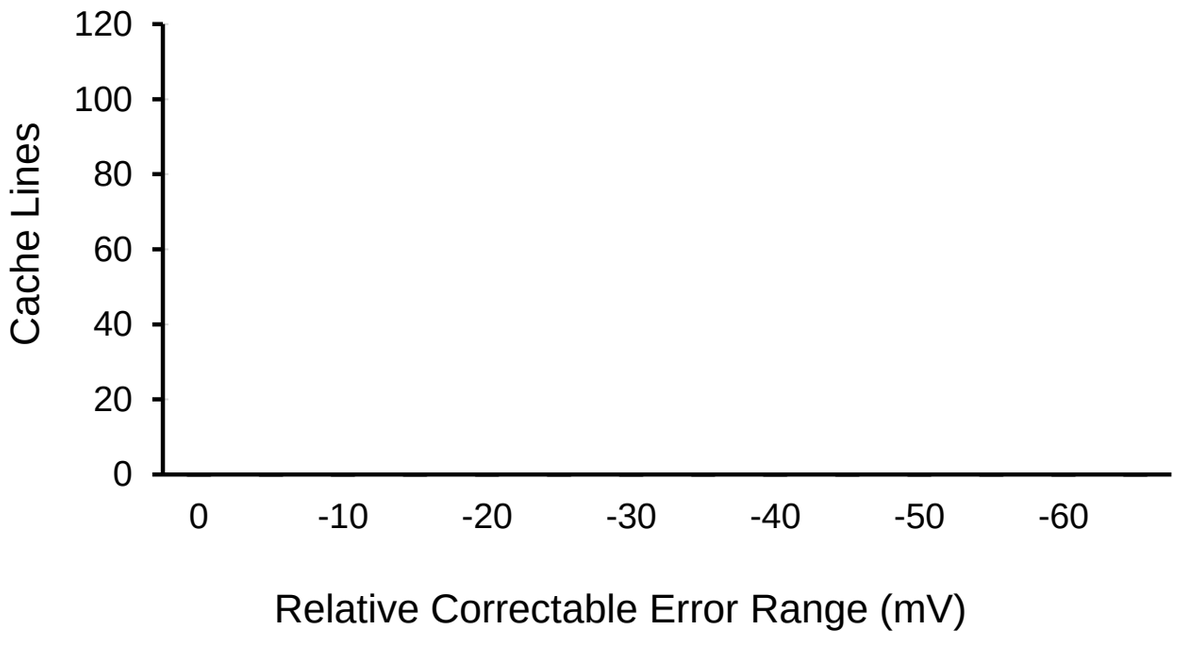
- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

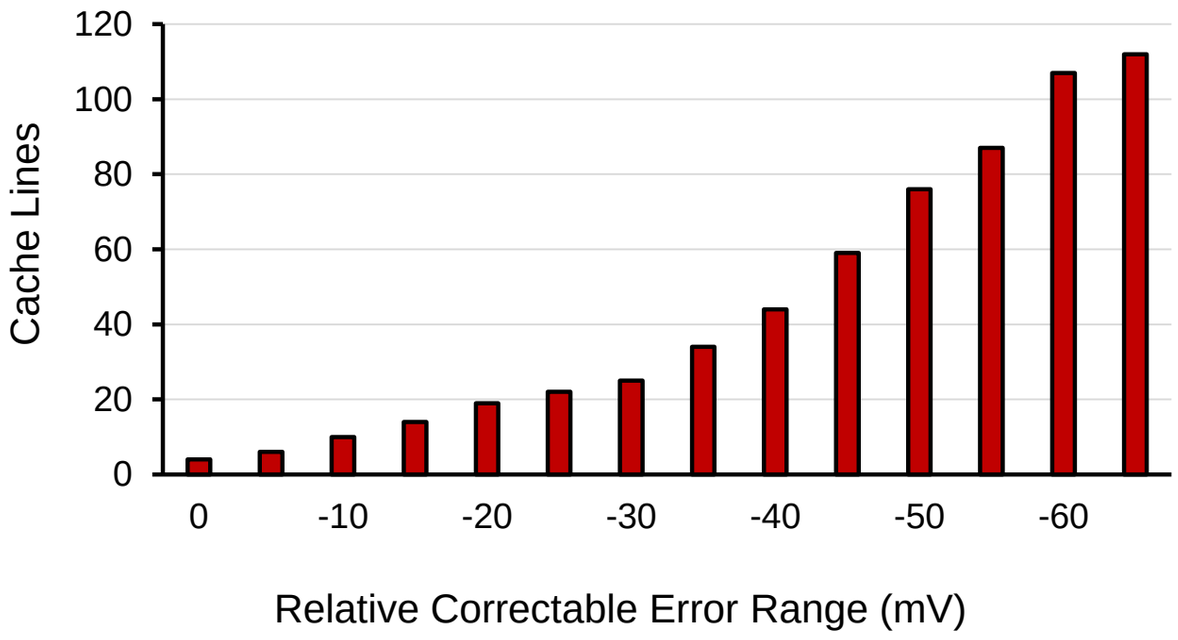
Intel 9560 Processor



Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

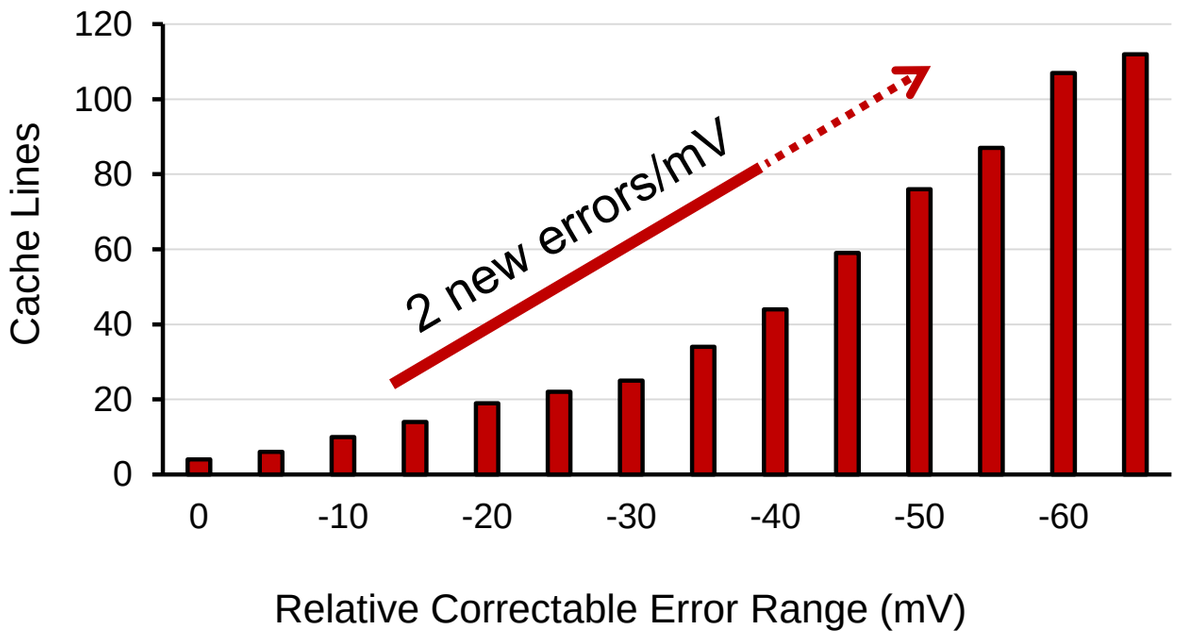


COMPUTER ARCHITECTURE RESEARCH LAB

Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

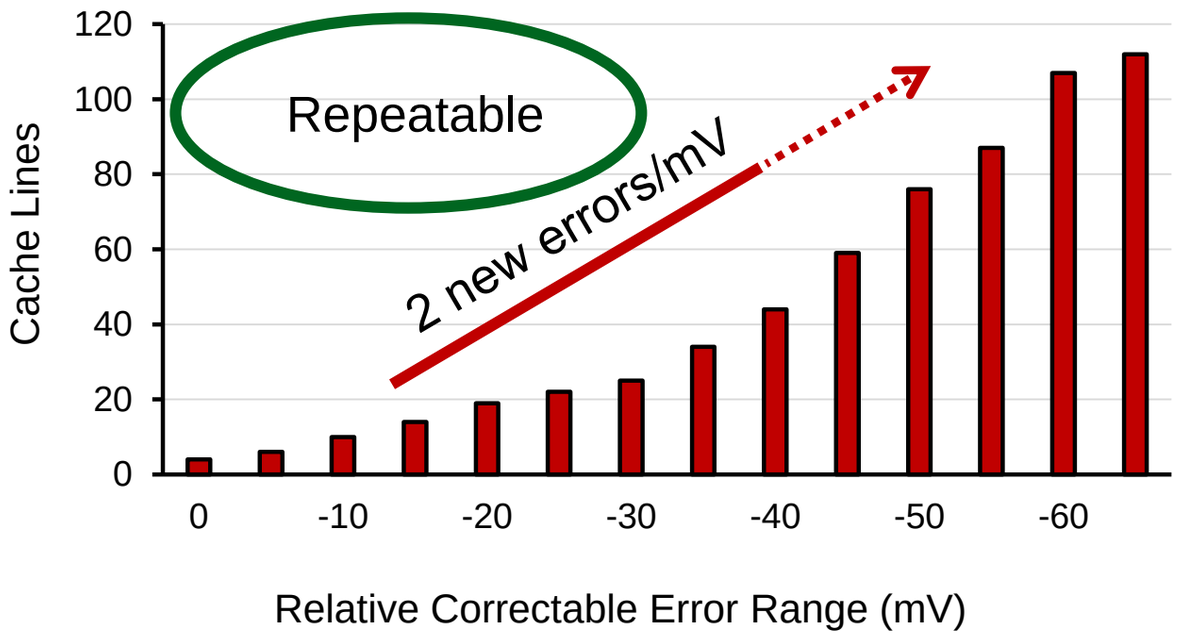
Intel 9560 Processor



Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

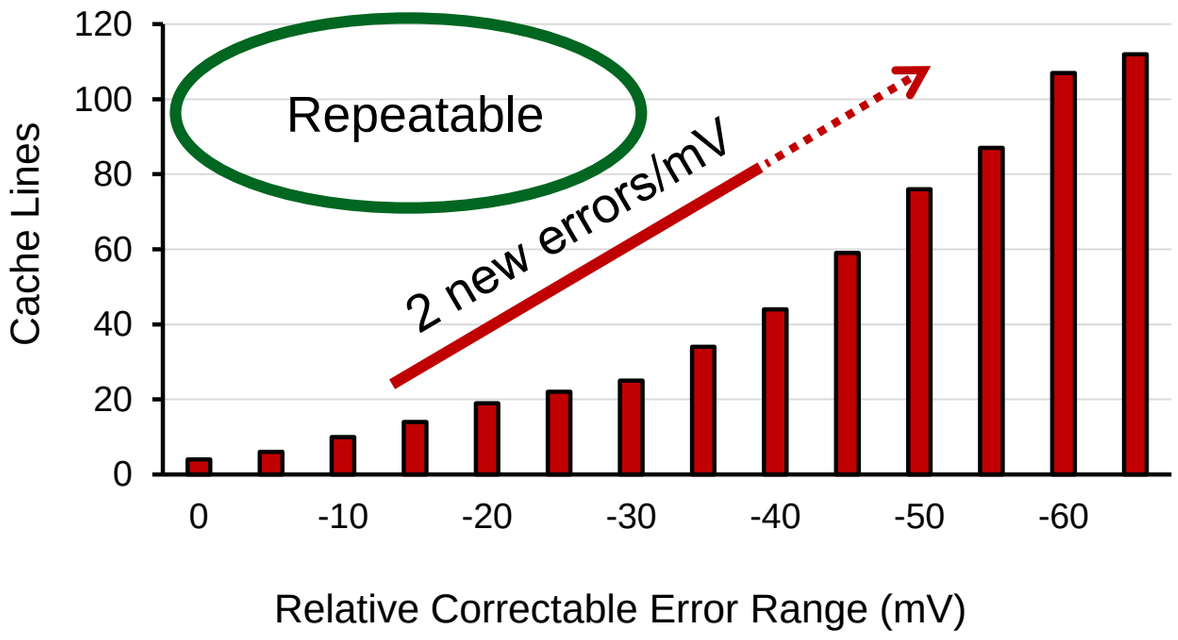
Intel 9560 Processor

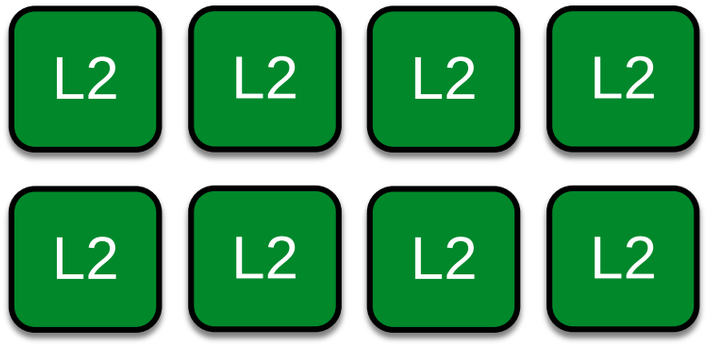


Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor



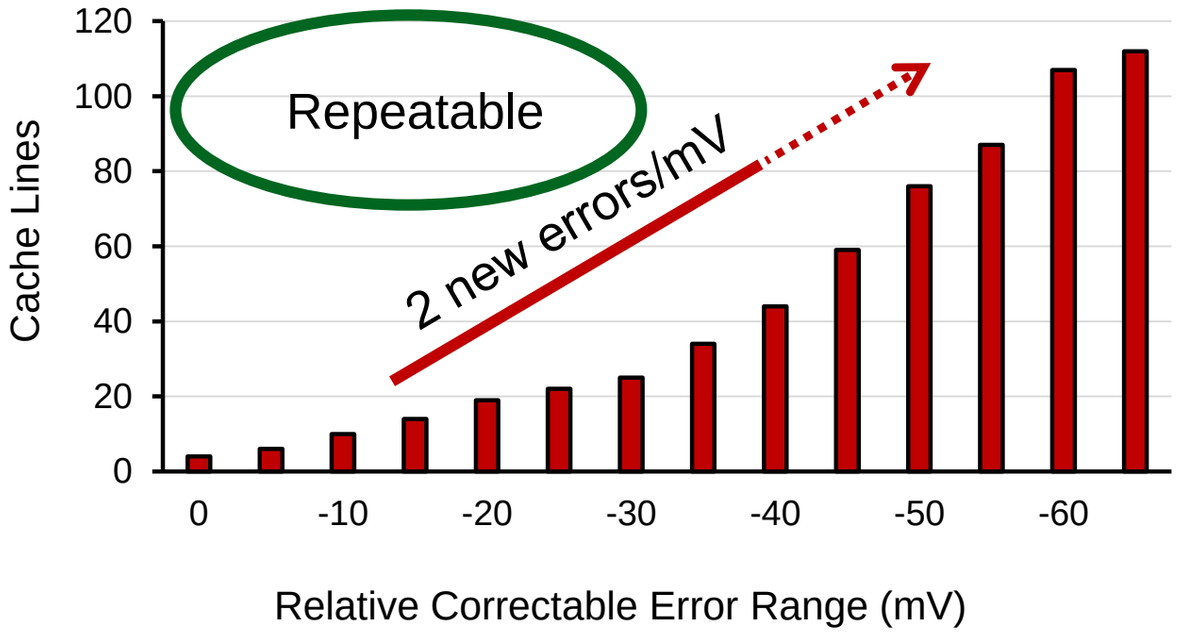


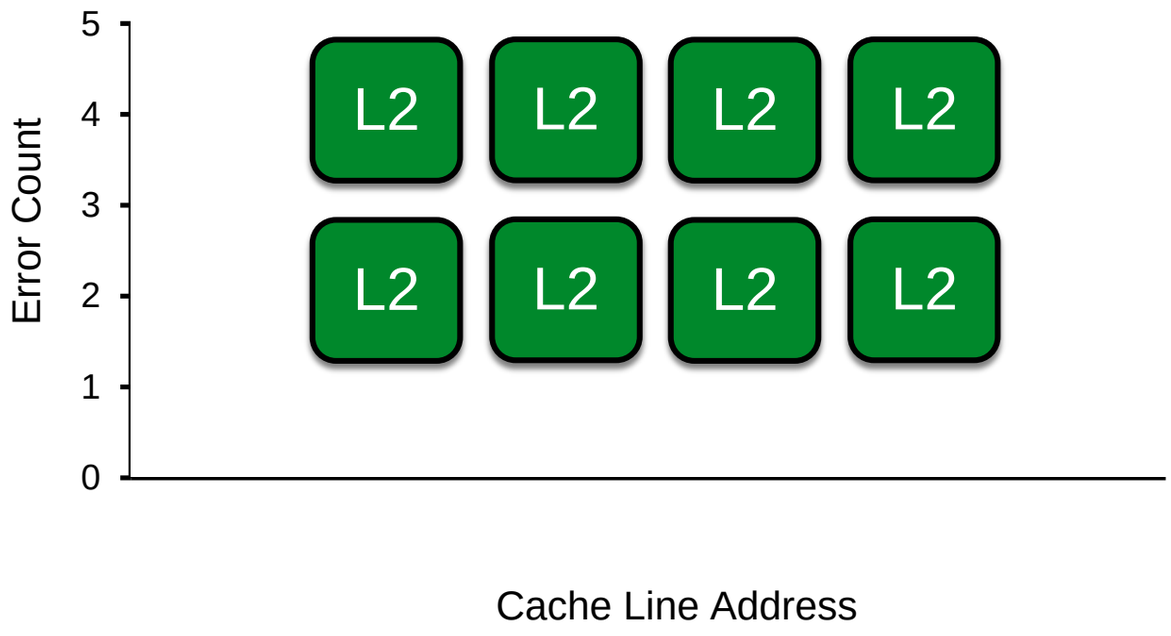
COMPUTER ARCHITECTURE RESEARCH LAB

Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

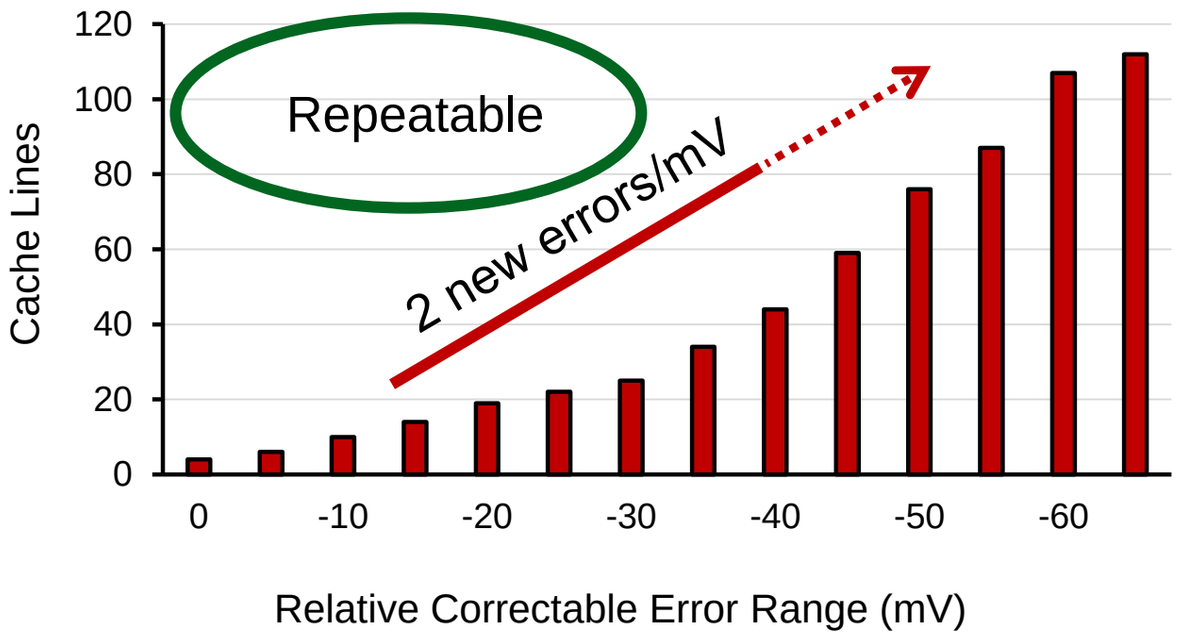


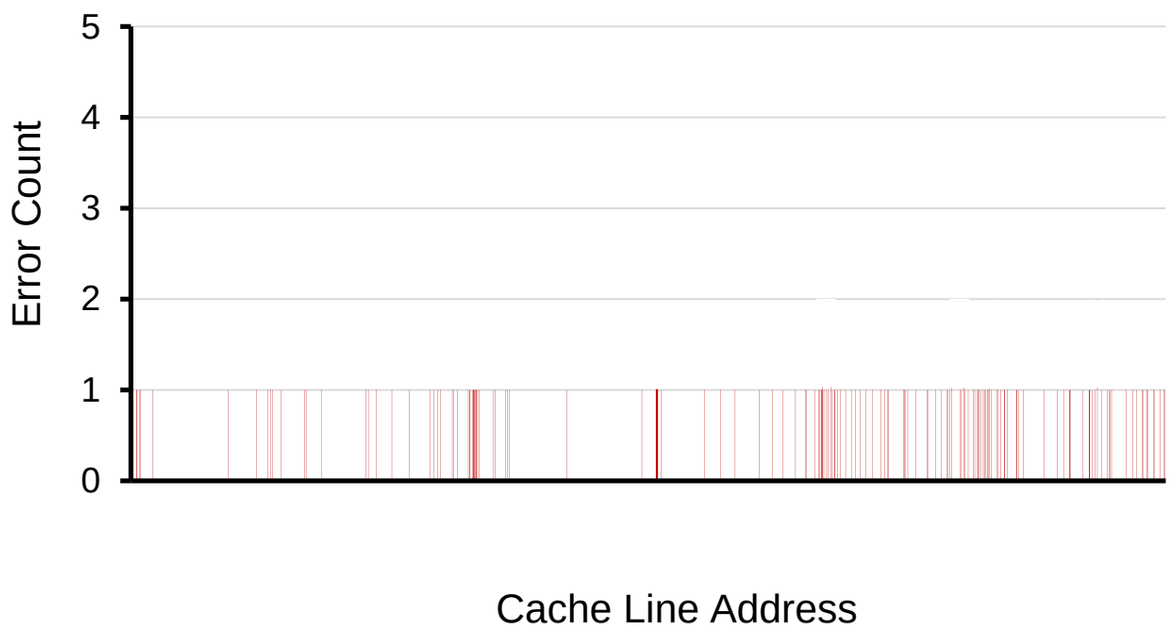


Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor



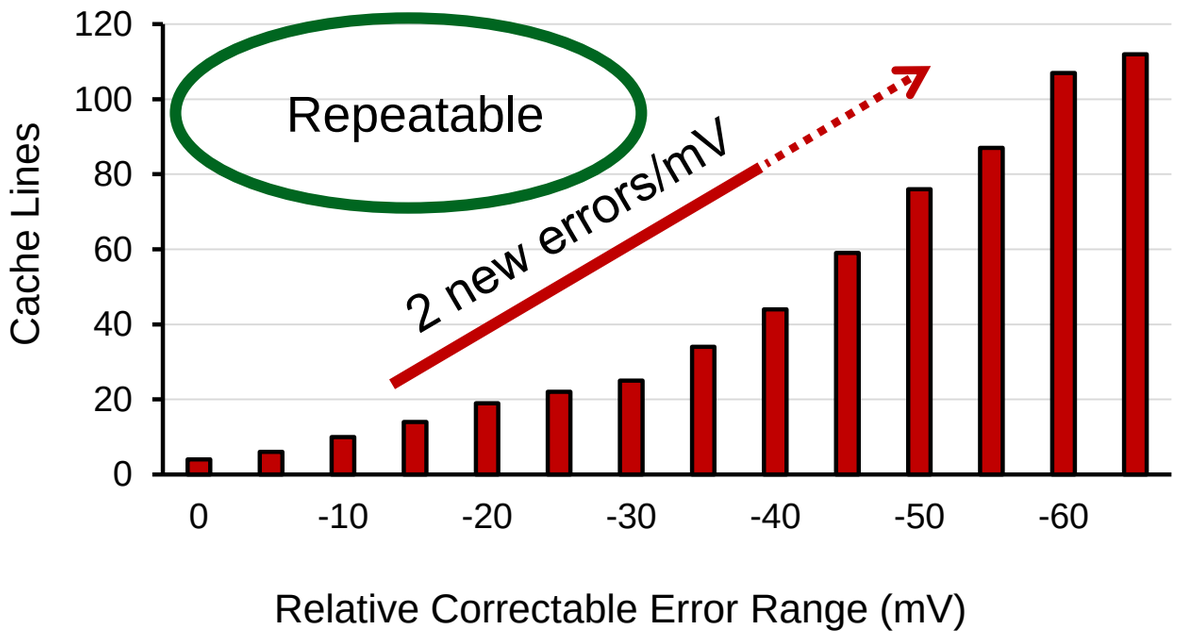


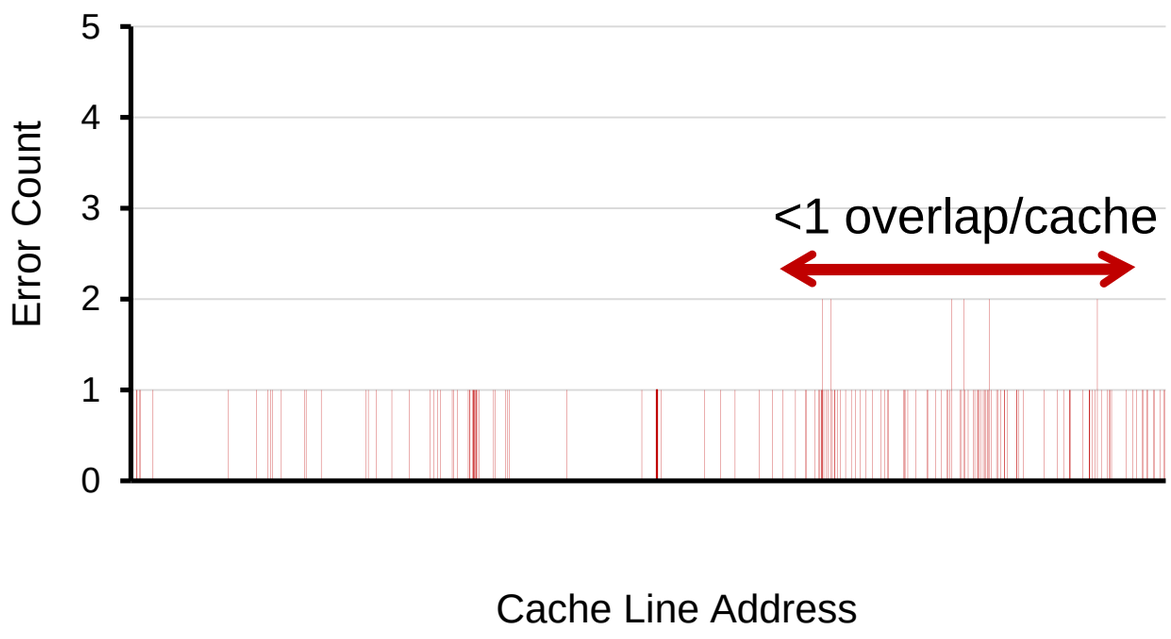
COMPUTER ARCHITECTURE RESEARCH LAB

Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

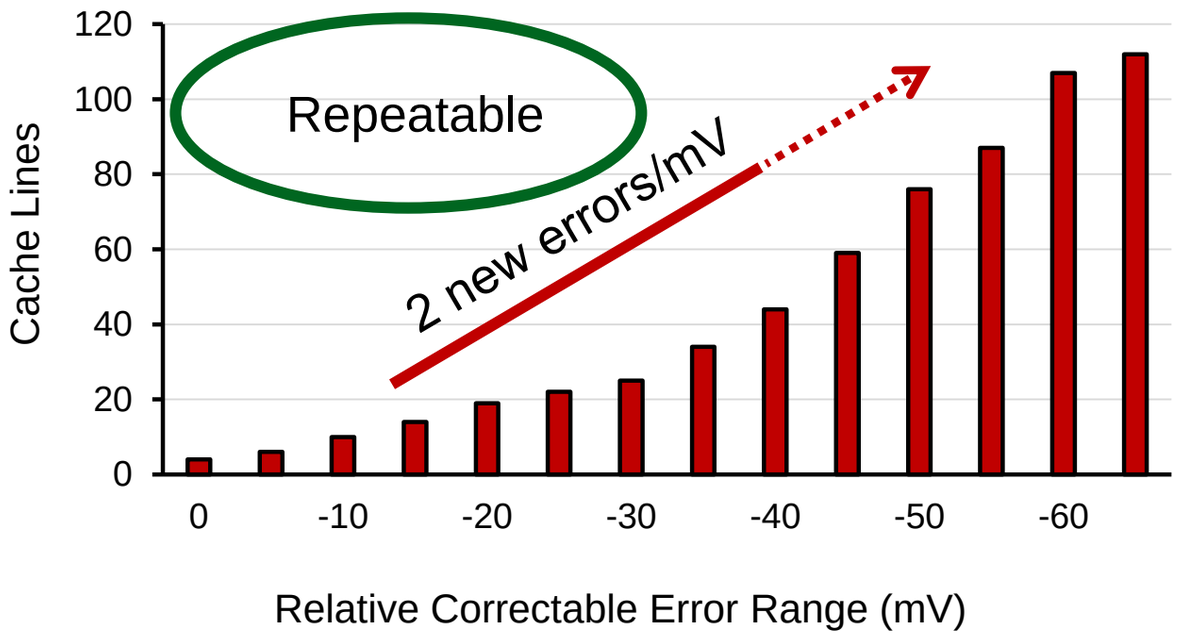


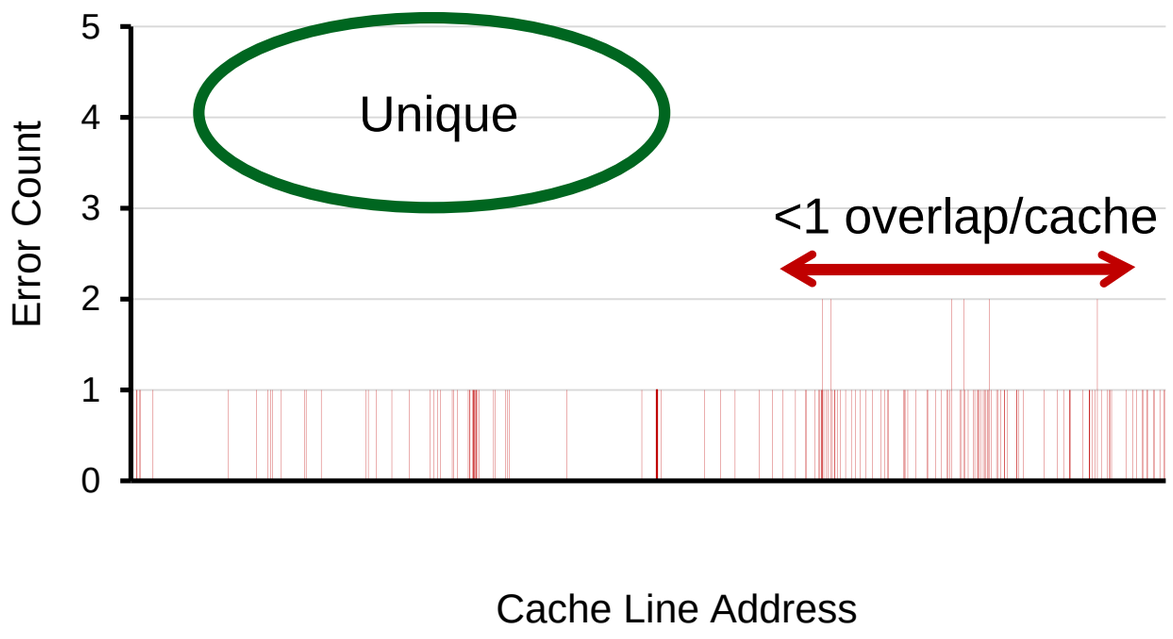


Cache Errors as Silicon Fingerprints

- Caches optimized for density
- Sensitive to process variation
- Itanium processor 8 L2 caches

Intel 9560 Processor

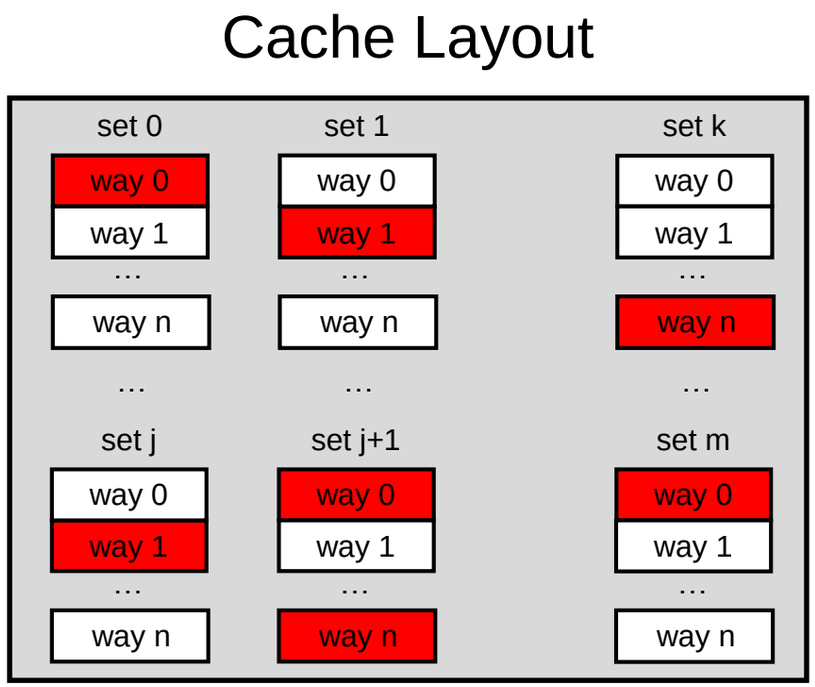




COMPUTER ARCHITECTURE RESEARCH LAB

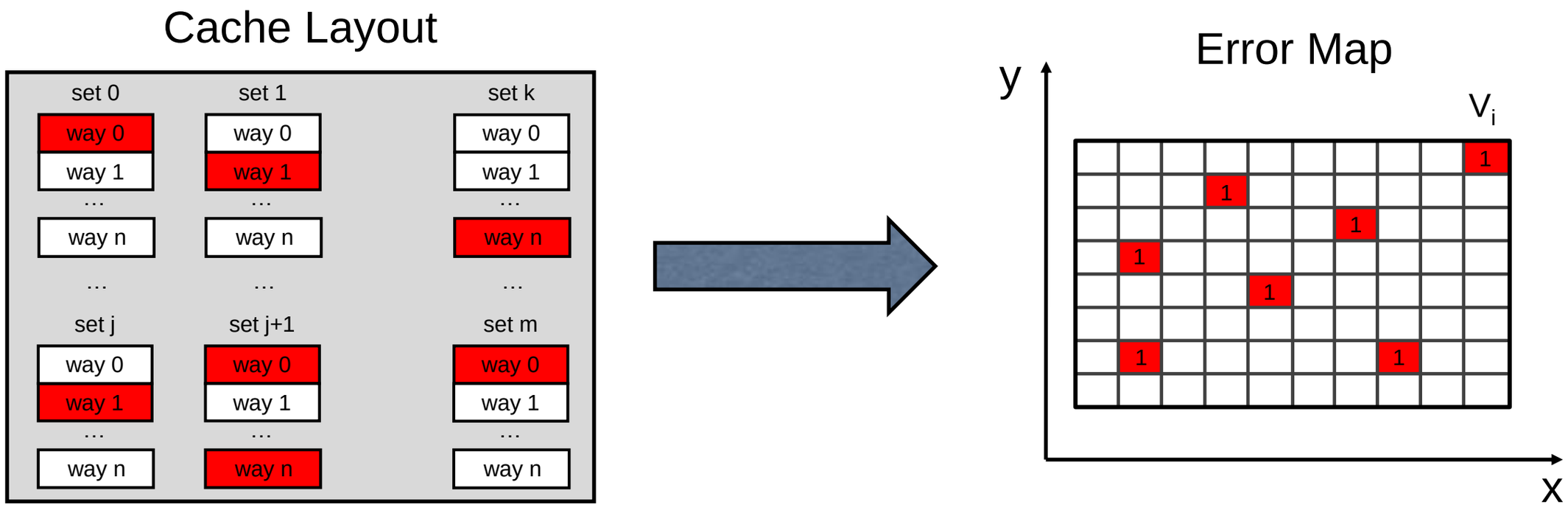
The Authenticache System

The Authenticache System



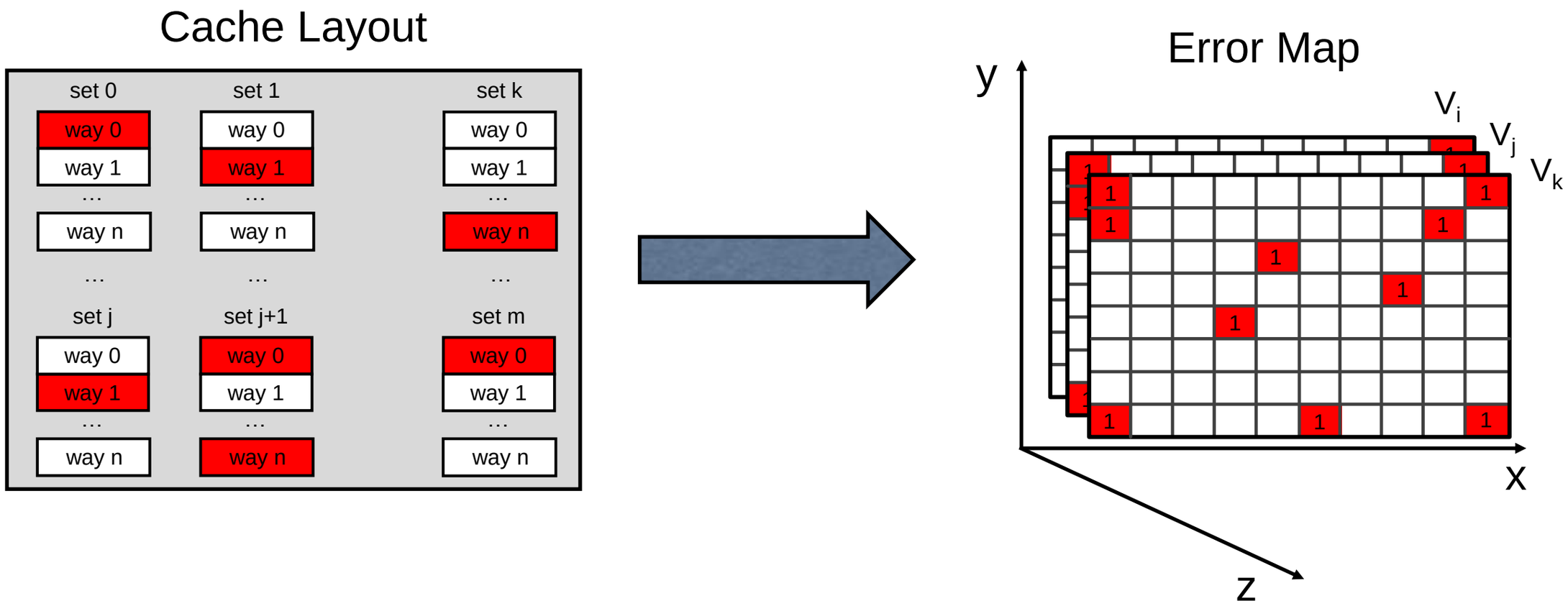
- Exploit process variation in LLC for randomness

The Authenticache System



- Exploit process variation in LLC for randomness
- Construct cache maps as a function of voltage and correctable errors

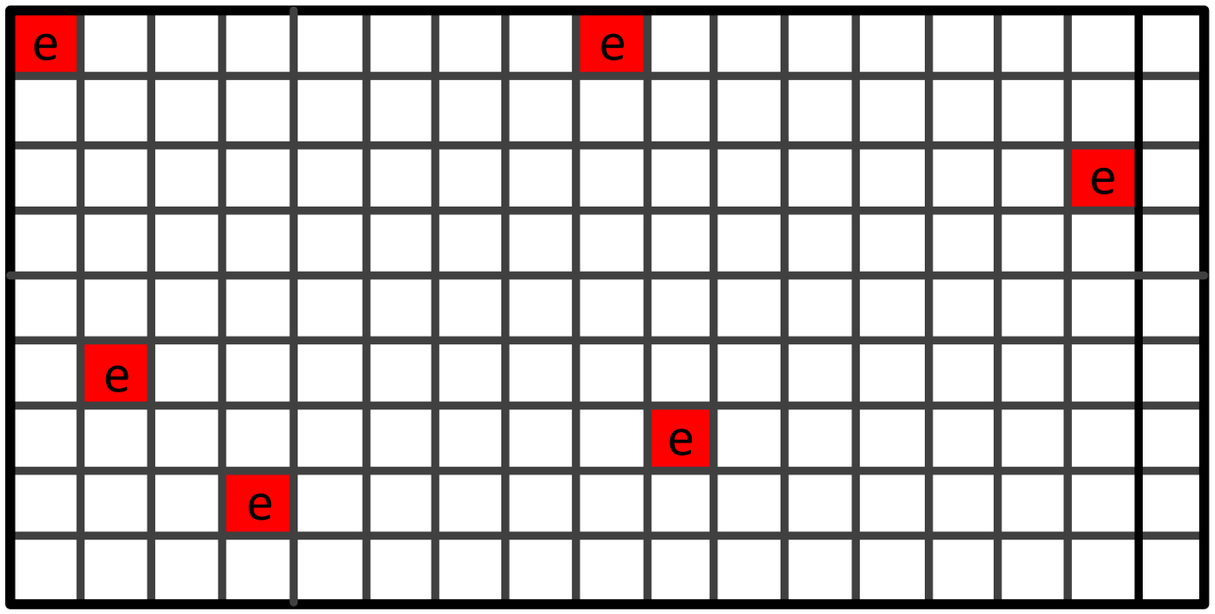
The Authenticache System



- Exploit process variation in LLC for randomness
- Construct cache maps as a function of voltage and correctable errors

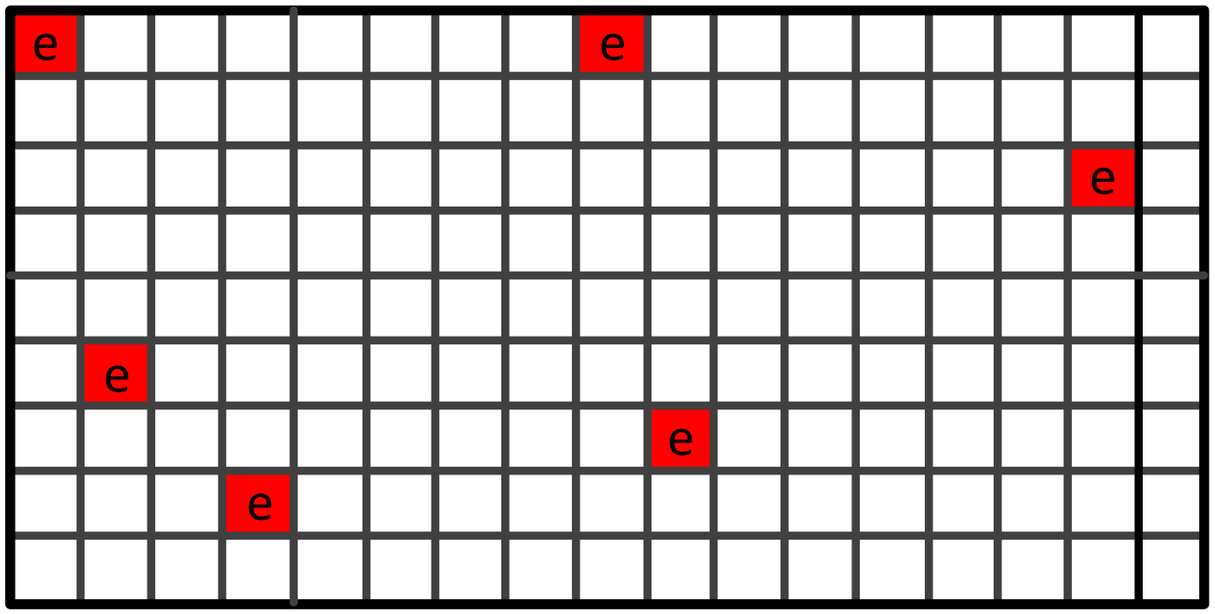
Challenge and Response

Error Map



Challenge and Response

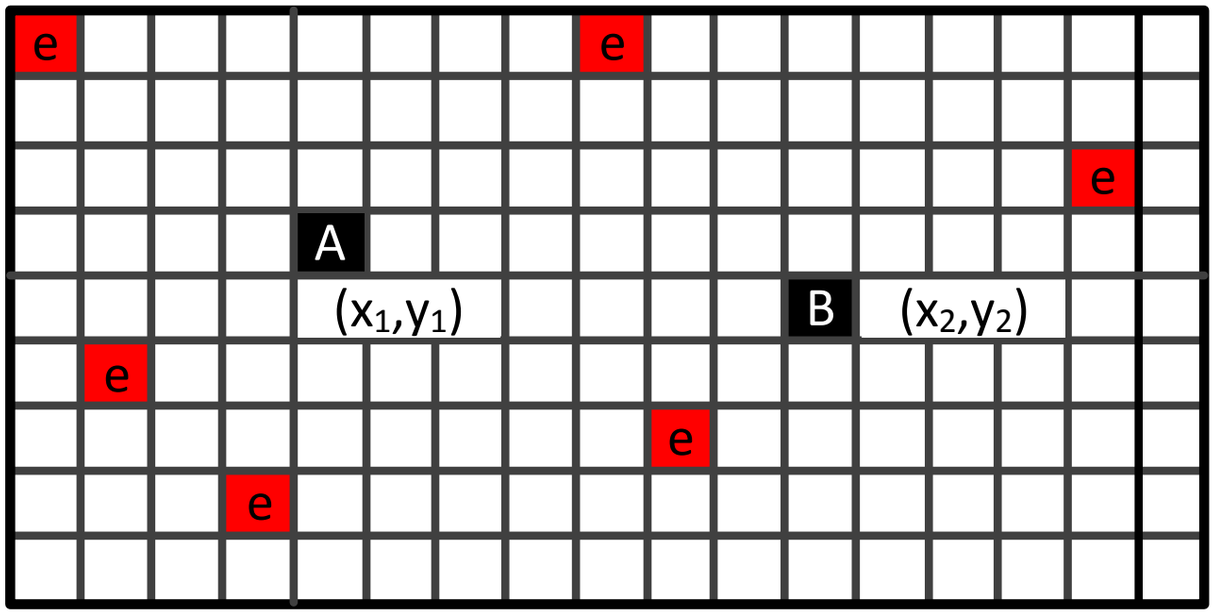
Error Map



$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

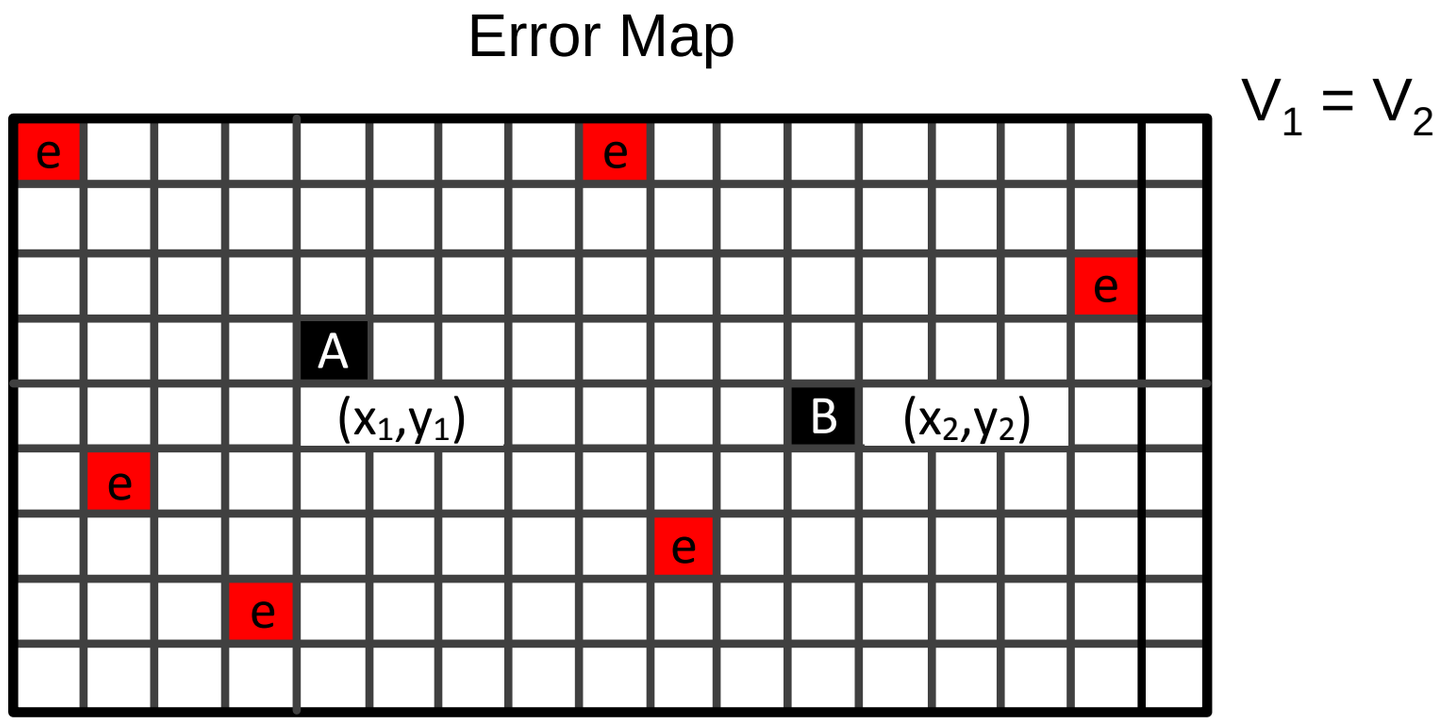
Challenge and Response

Error Map



$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

Challenge and Response

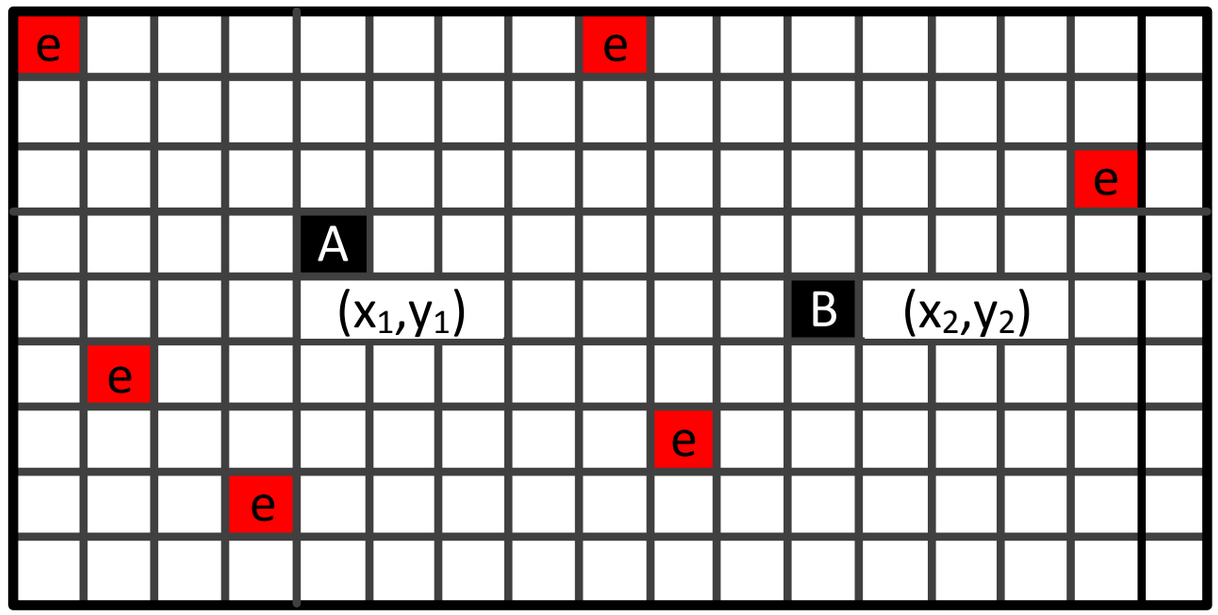


$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

Challenge and Response

Error Map

$V_1 = V_2$



$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

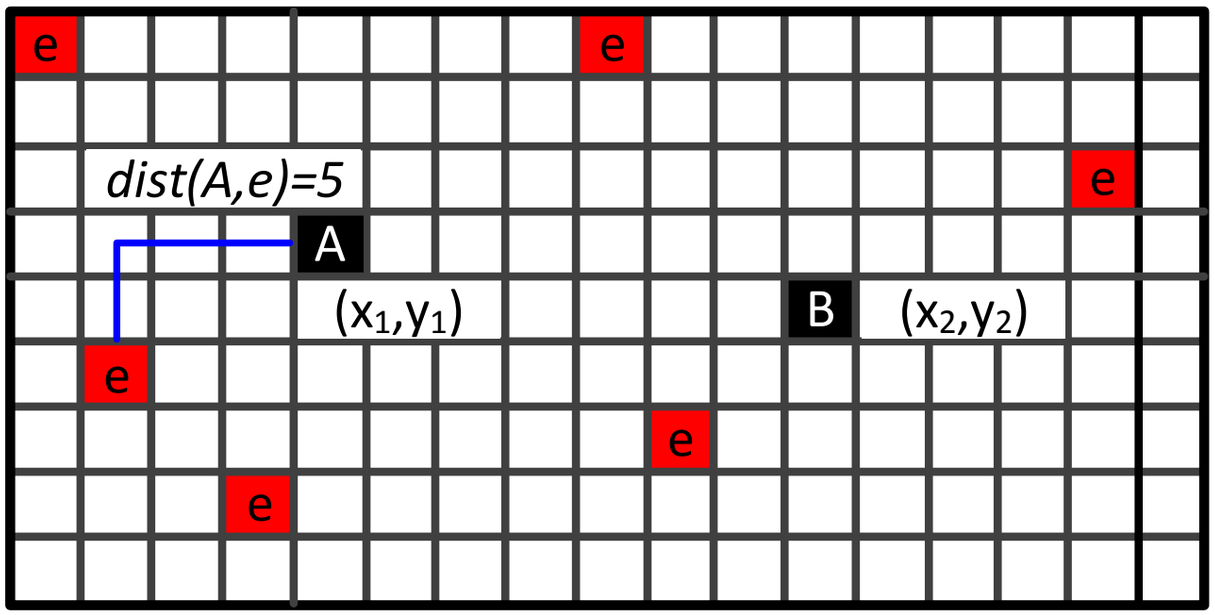
$$Response = \begin{cases} 0, dist(A, e_a) < dist(B, e_b) \\ 1, dist(A, e_a) \geq dist(B, e_b) \end{cases}$$

Manhattan Distance

Challenge and Response

Error Map

$V_1 = V_2$



$$Challenge = \underbrace{(x_1, y_1}_{A}, V_1), \underbrace{(x_2, y_2}_{B}, V_2)$$

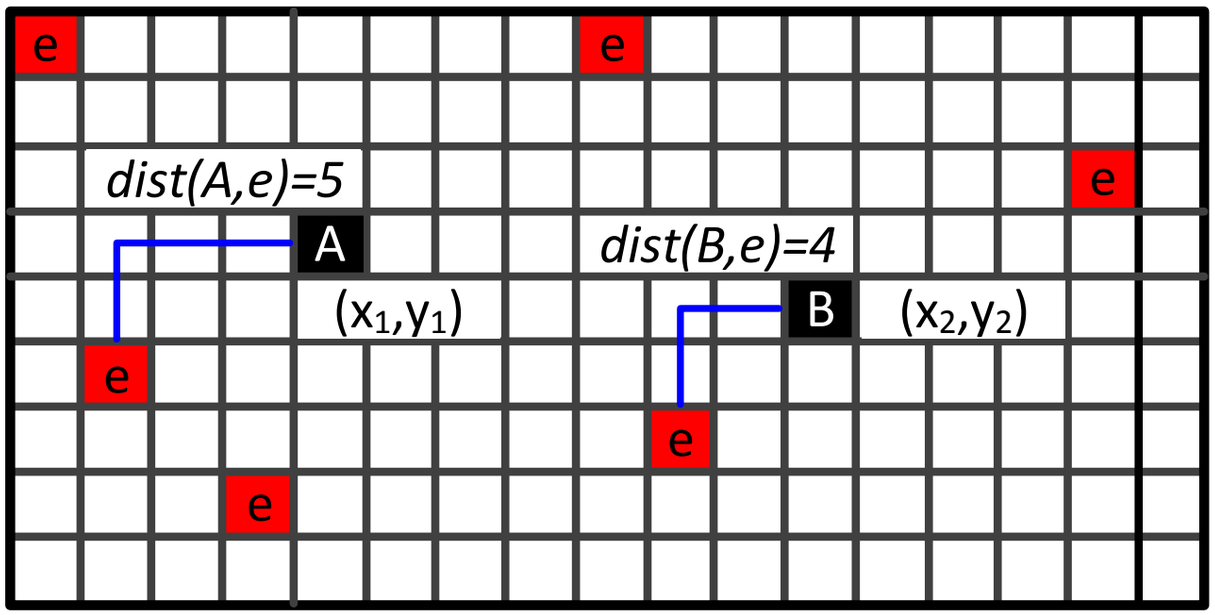
$$Response = \begin{cases} 0, dist(A, e_a) < dist(B, e_b) \\ 1, dist(A, e_a) \geq dist(B, e_b) \end{cases}$$

Manhattan Distance

Challenge and Response

Error Map

$V_1 = V_2$



$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

$$Response = \begin{cases} 0, dist(A, e_a) < dist(B, e_b) \\ 1, dist(A, e_a) \geq dist(B, e_b) \end{cases}$$

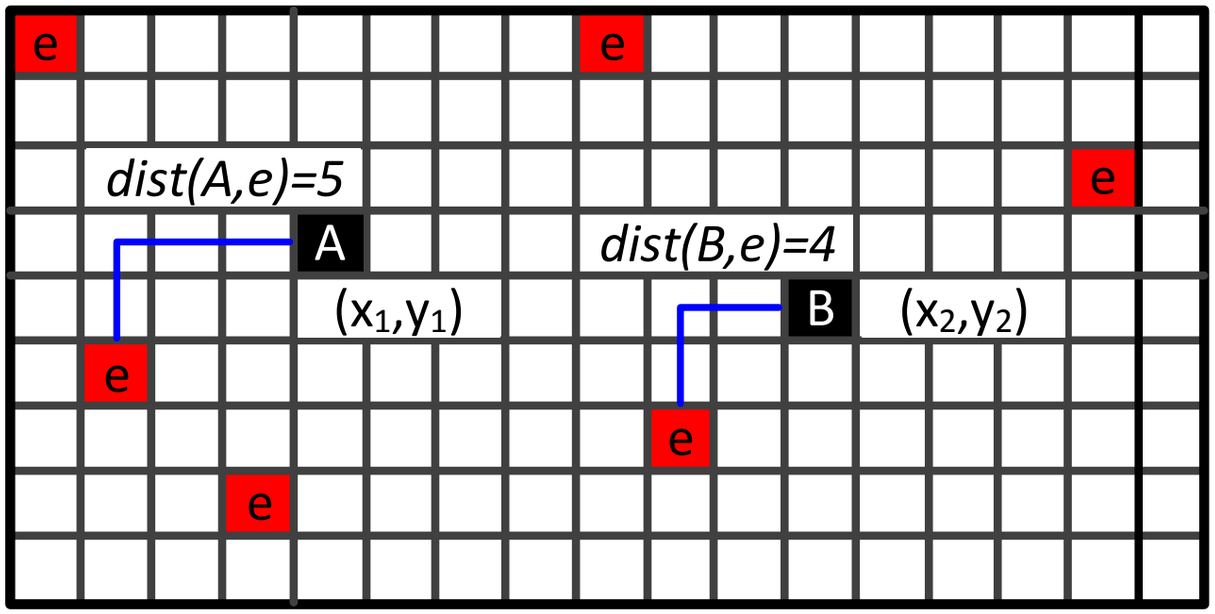
Manhattan Distance

Challenge and Response

5 > 4

Error Map

$V_1 = V_2$



$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

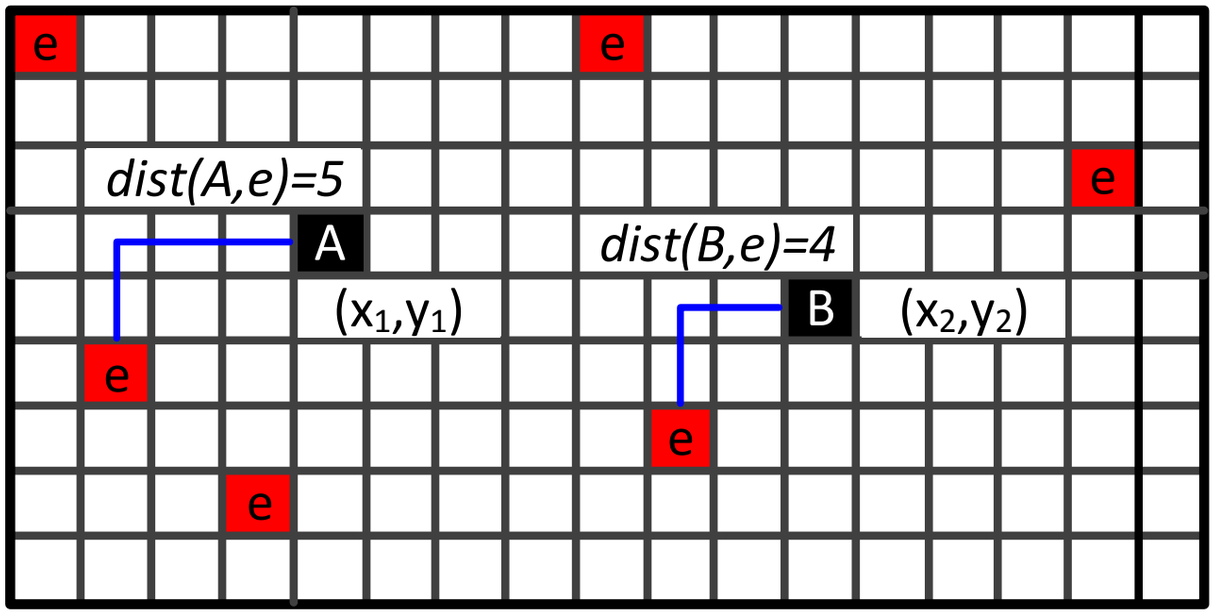
$$Response = \begin{cases} 0, dist(A, e_a) < dist(B, e_b) \\ 1, dist(A, e_a) \geq dist(B, e_b) \end{cases}$$

Manhattan Distance

Challenge and Response

5 > 4

Error Map



$V_1 = V_2$

1

$$Challenge = \underbrace{(x_1, y_1, V_1)}_{A}, \underbrace{(x_2, y_2, V_2)}_{B}$$

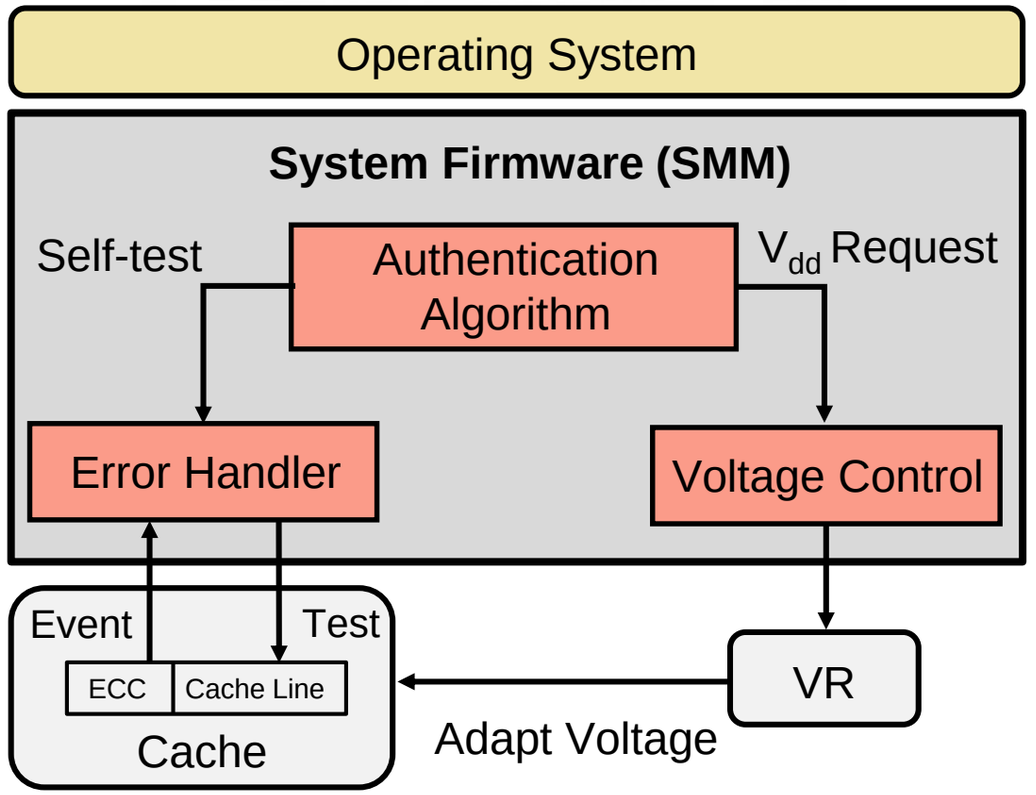
$$Response = \begin{cases} 0, dist(A, e_a) < dist(B, e_b) \\ 1, dist(A, e_a) \geq dist(B, e_b) \end{cases}$$

Manhattan Distance

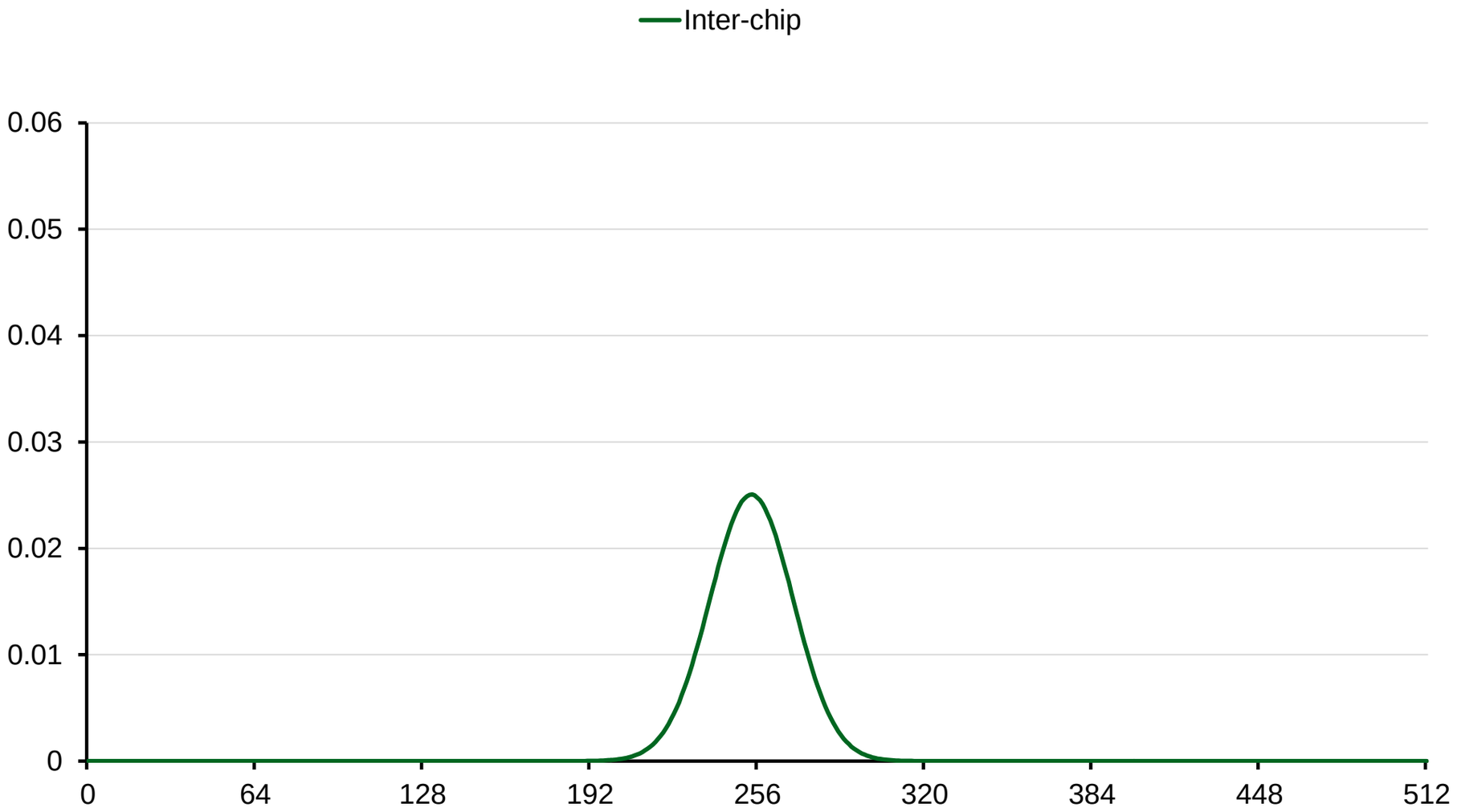
Experimental Framework

- System:
  - BL860c-i4 Integrity Server from HP
  - 2x 9560 Itanium II CPUs

- Prototype in System Firmware
  - Thermal experiments through power virus
- Monte Carlo simulations
  - Different cache sizes
  - Different error maps and noise profiles

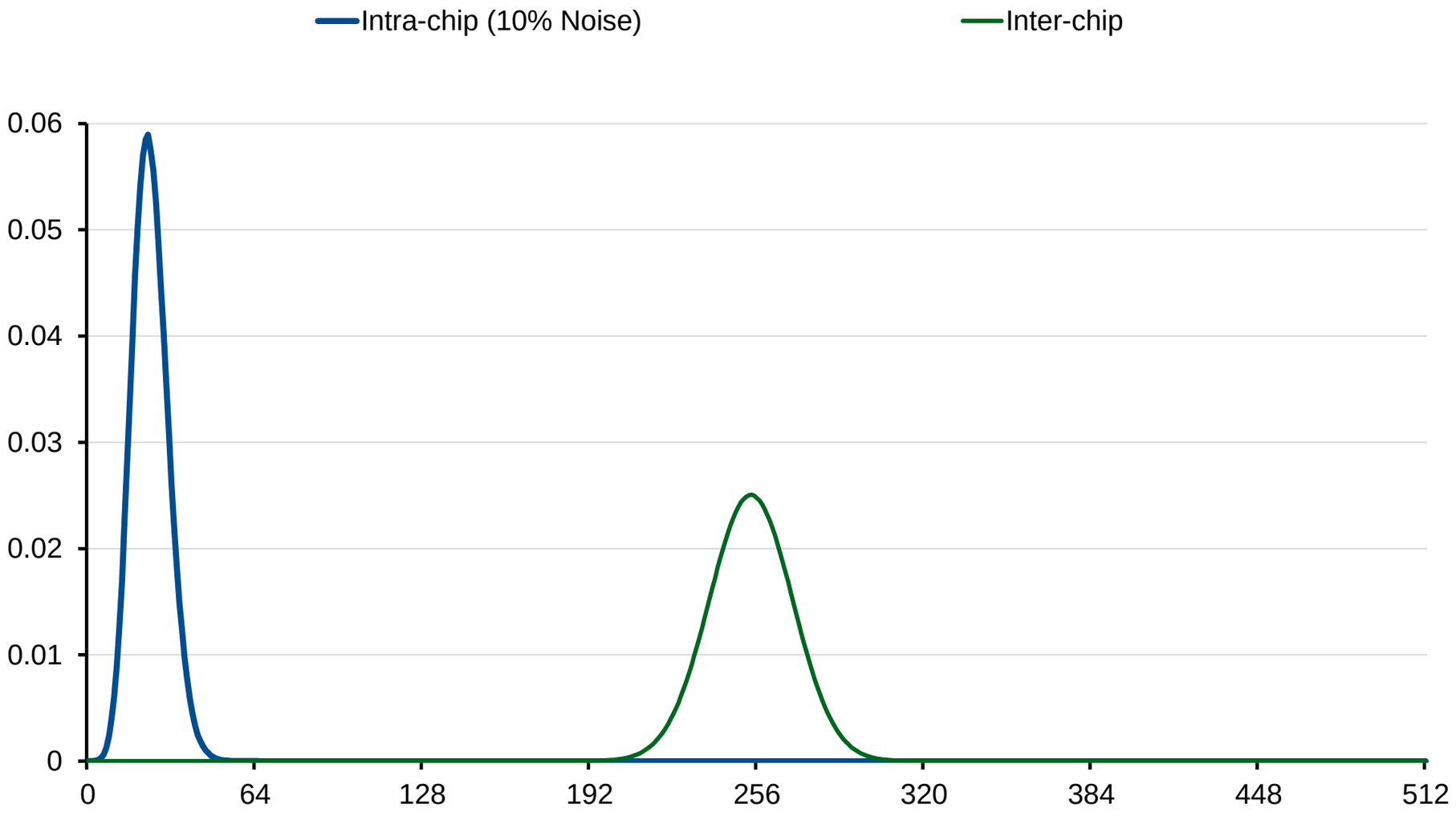


Identification and Noise



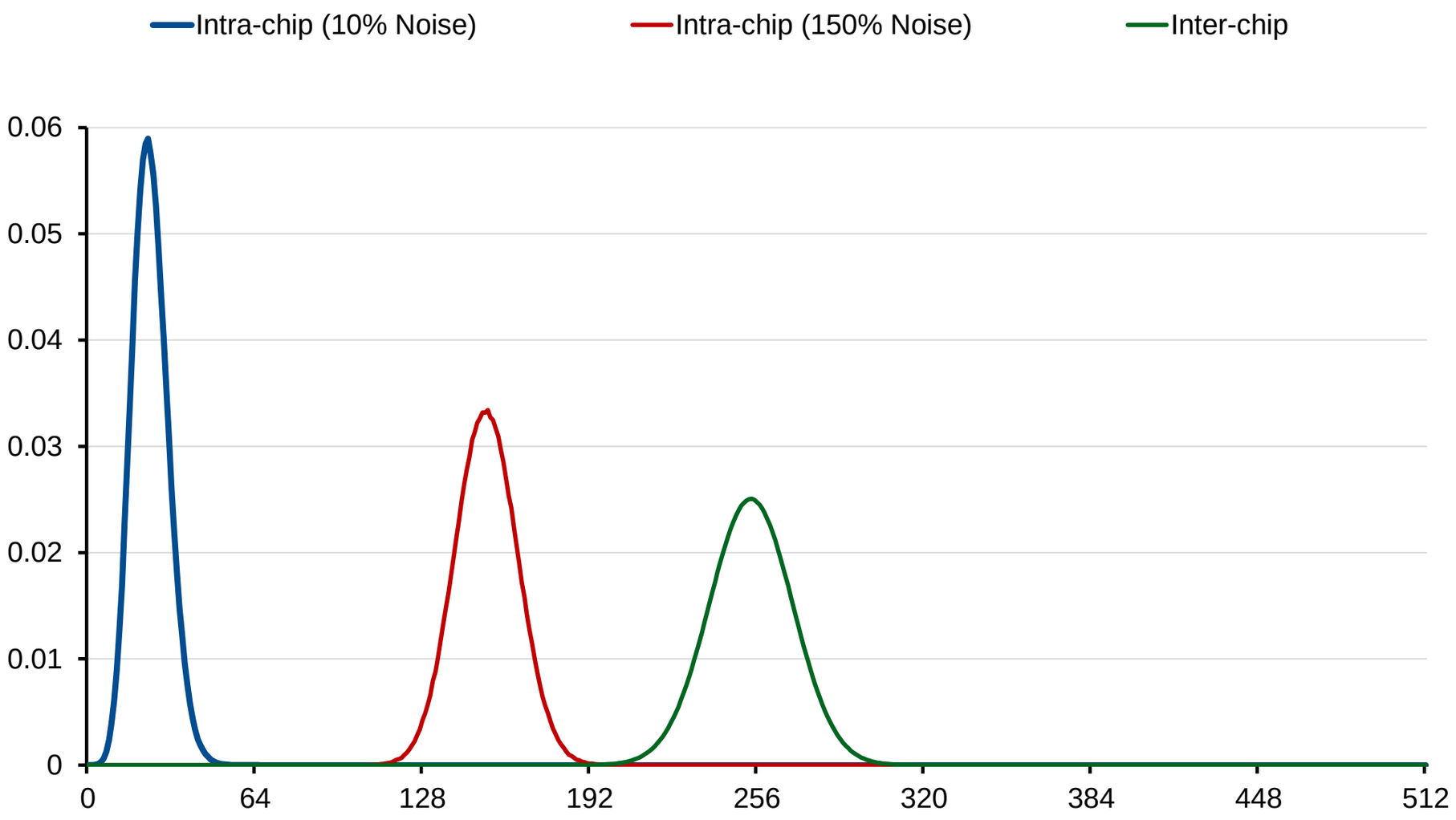
Identification in presence of environmental and measurement noise

Identification and Noise



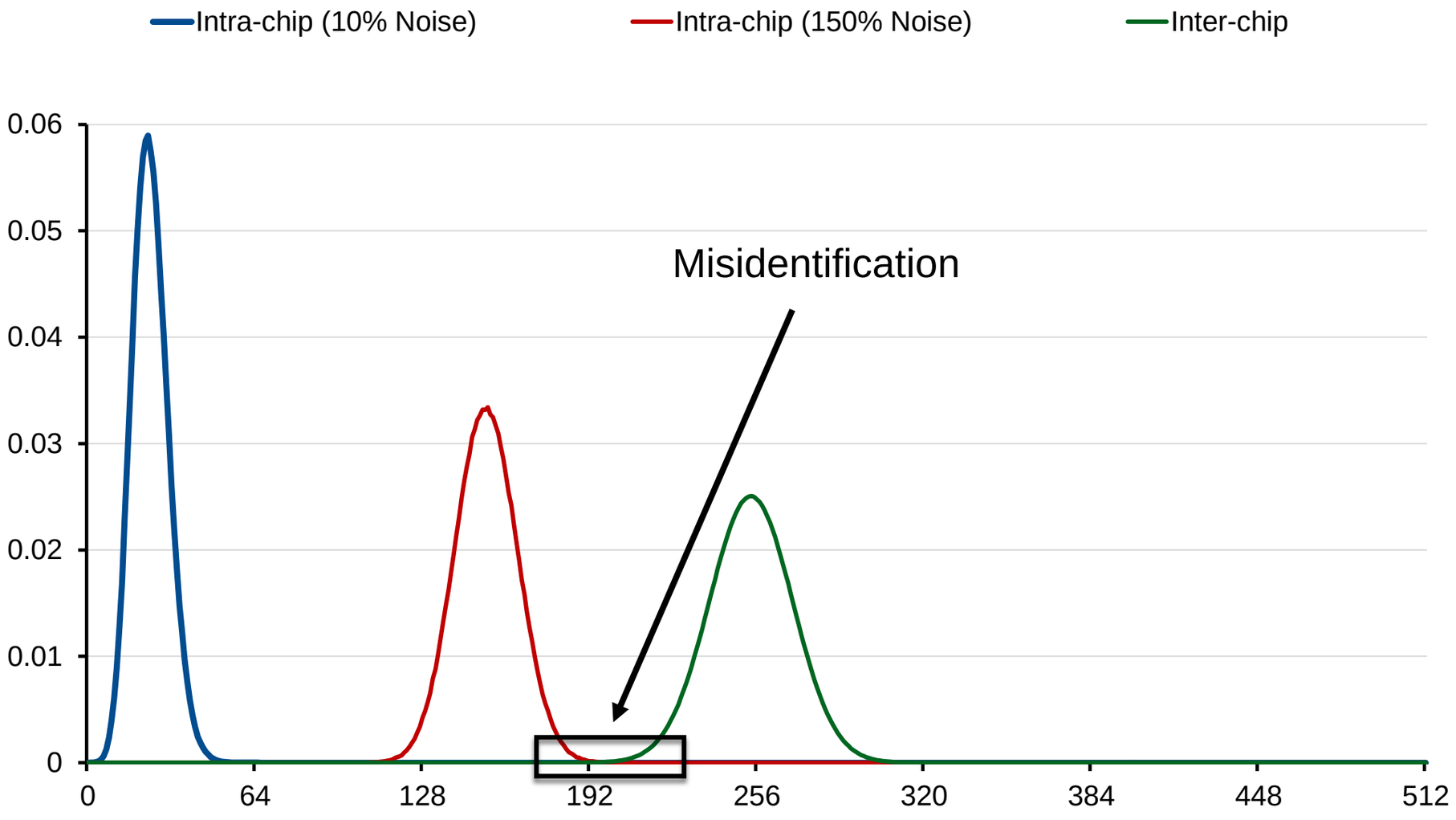
Identification in presence of environmental and measurement noise

Identification and Noise



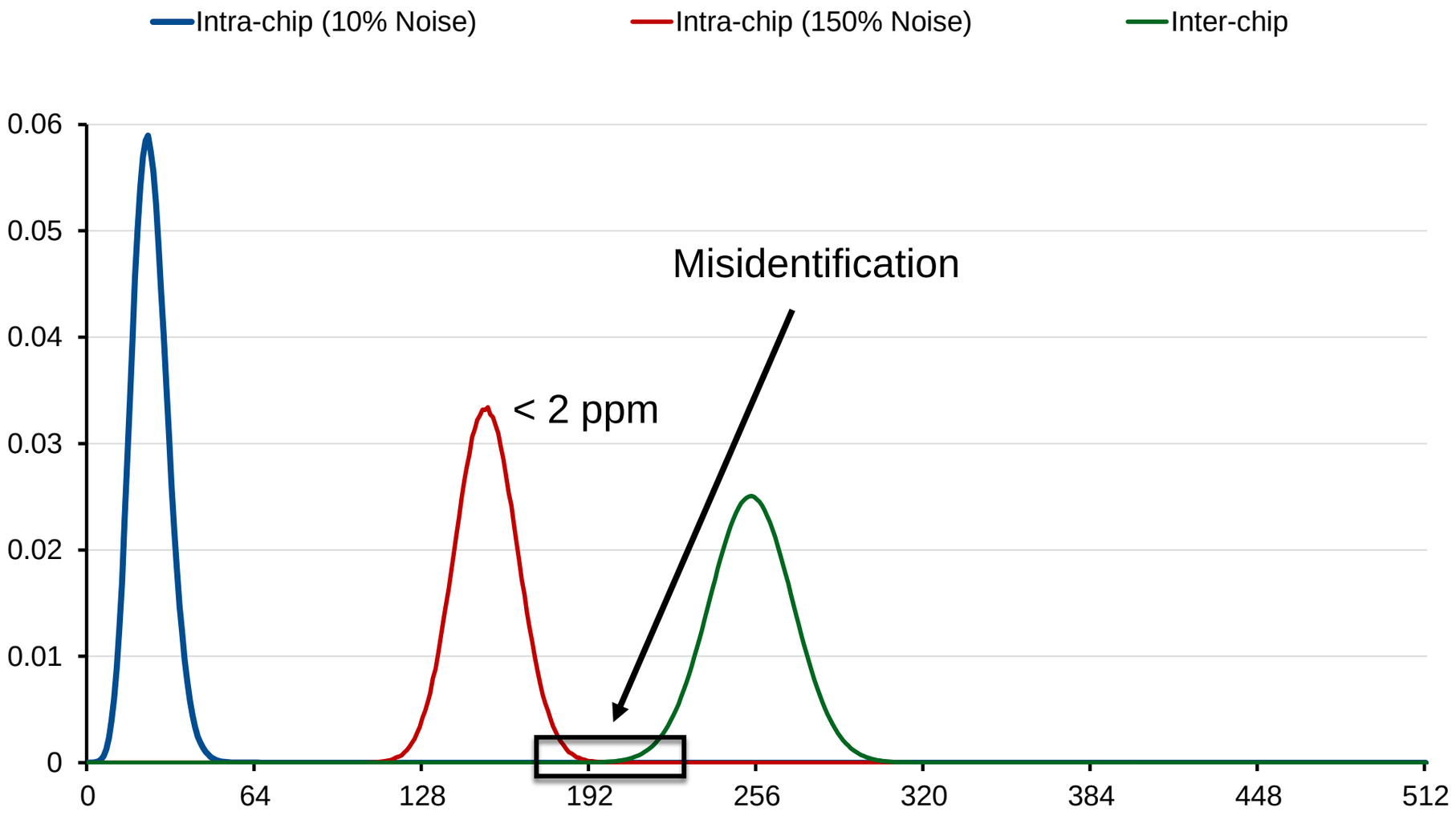
Identification in presence of environmental and measurement noise

Identification and Noise



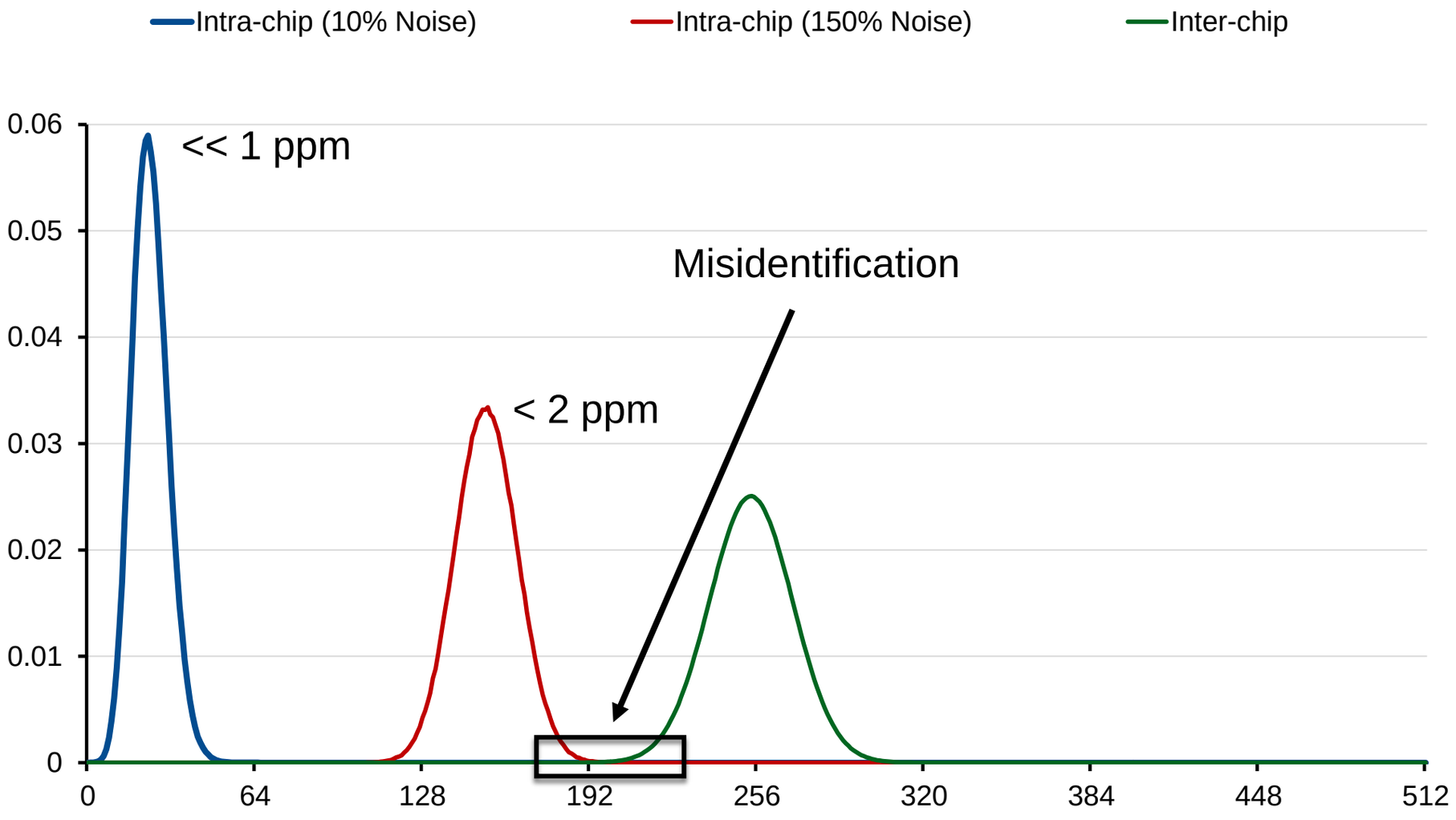
Identification in presence of environmental and measurement noise

Identification and Noise



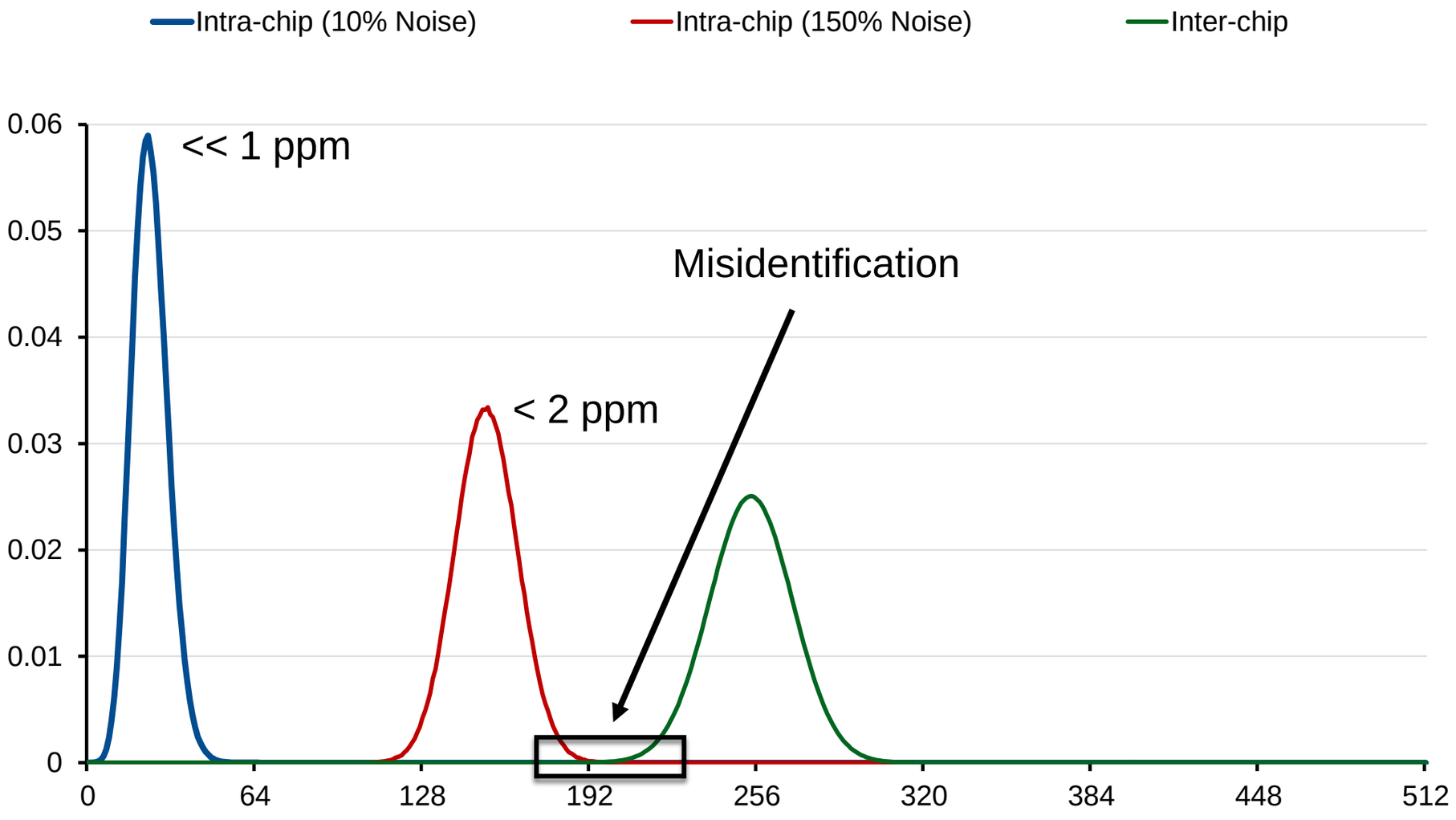
Identification in presence of environmental and measurement noise

Identification and Noise



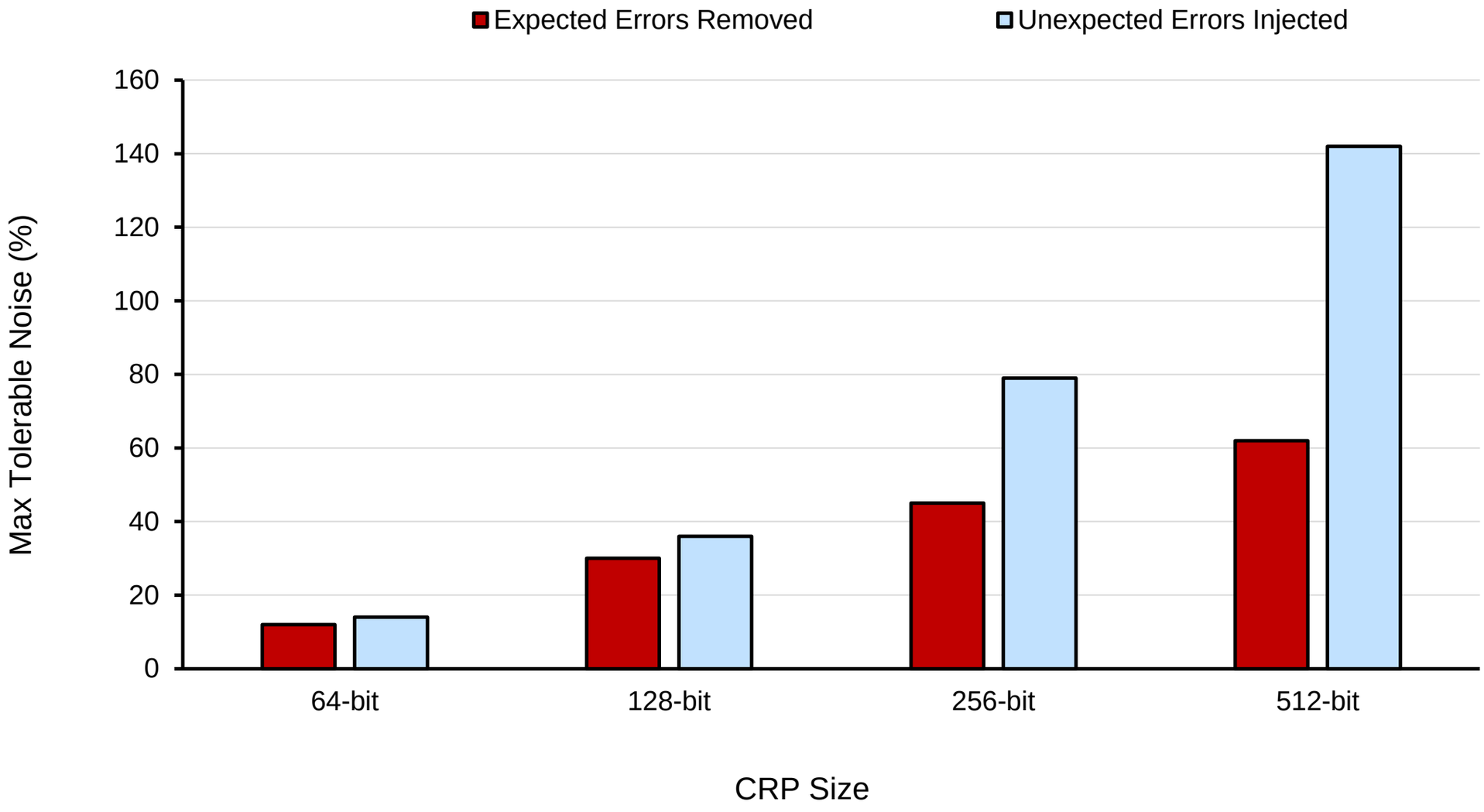
Identification in presence of environmental and measurement noise

Identification and Noise

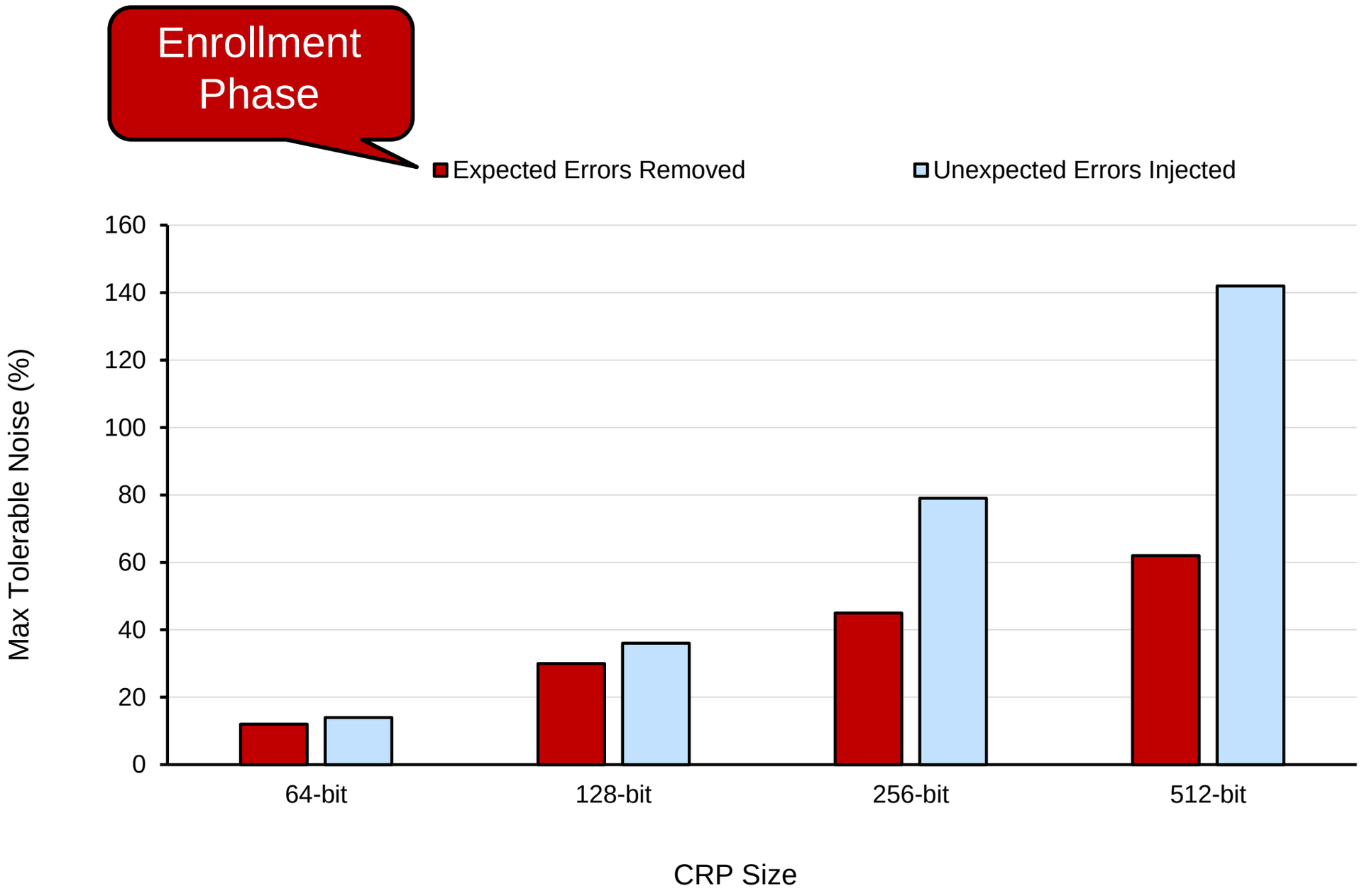


Observe 6% intra-chip variation after +25° C

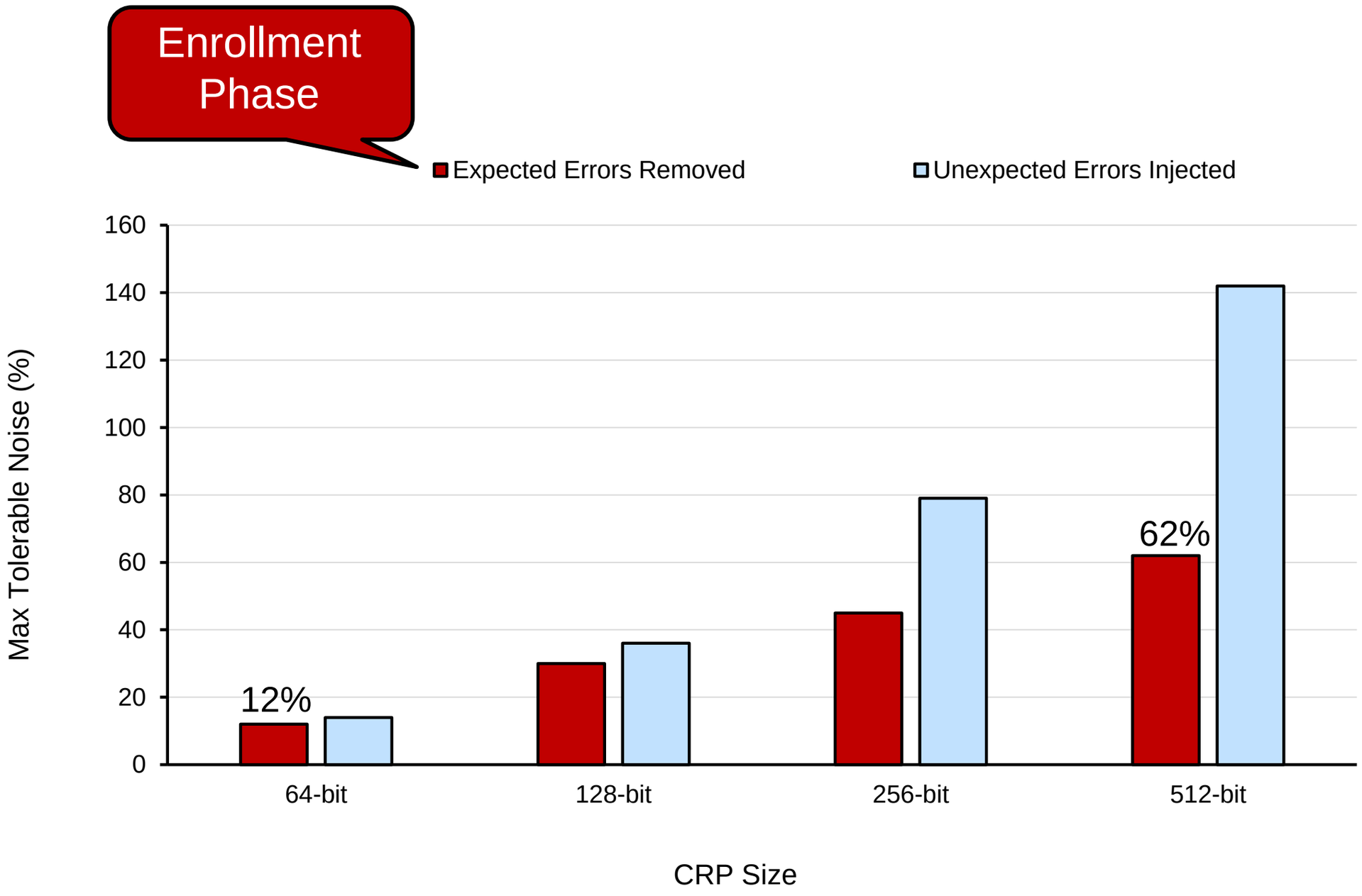
Resiliency to Noise



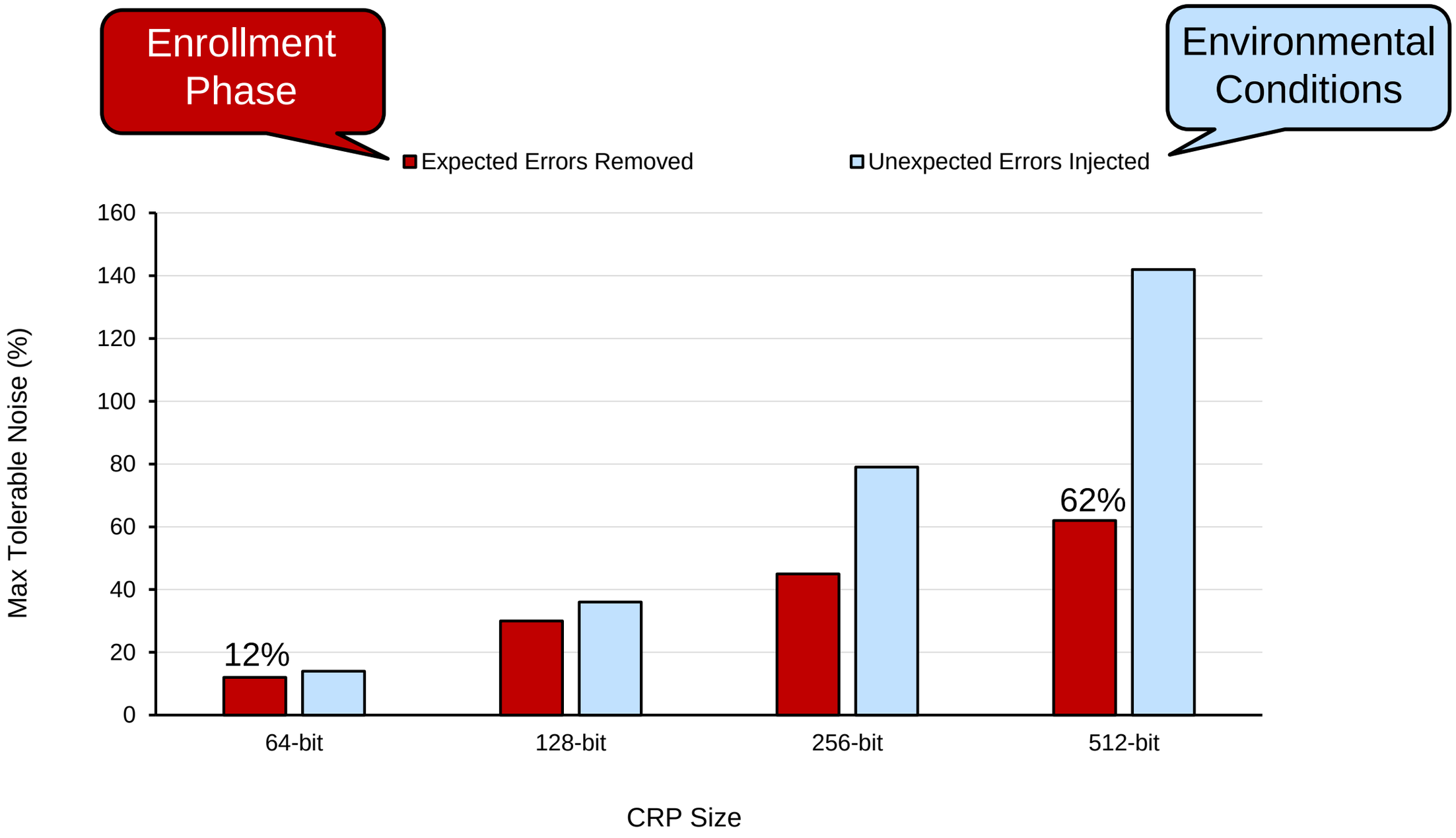
Resiliency to Noise



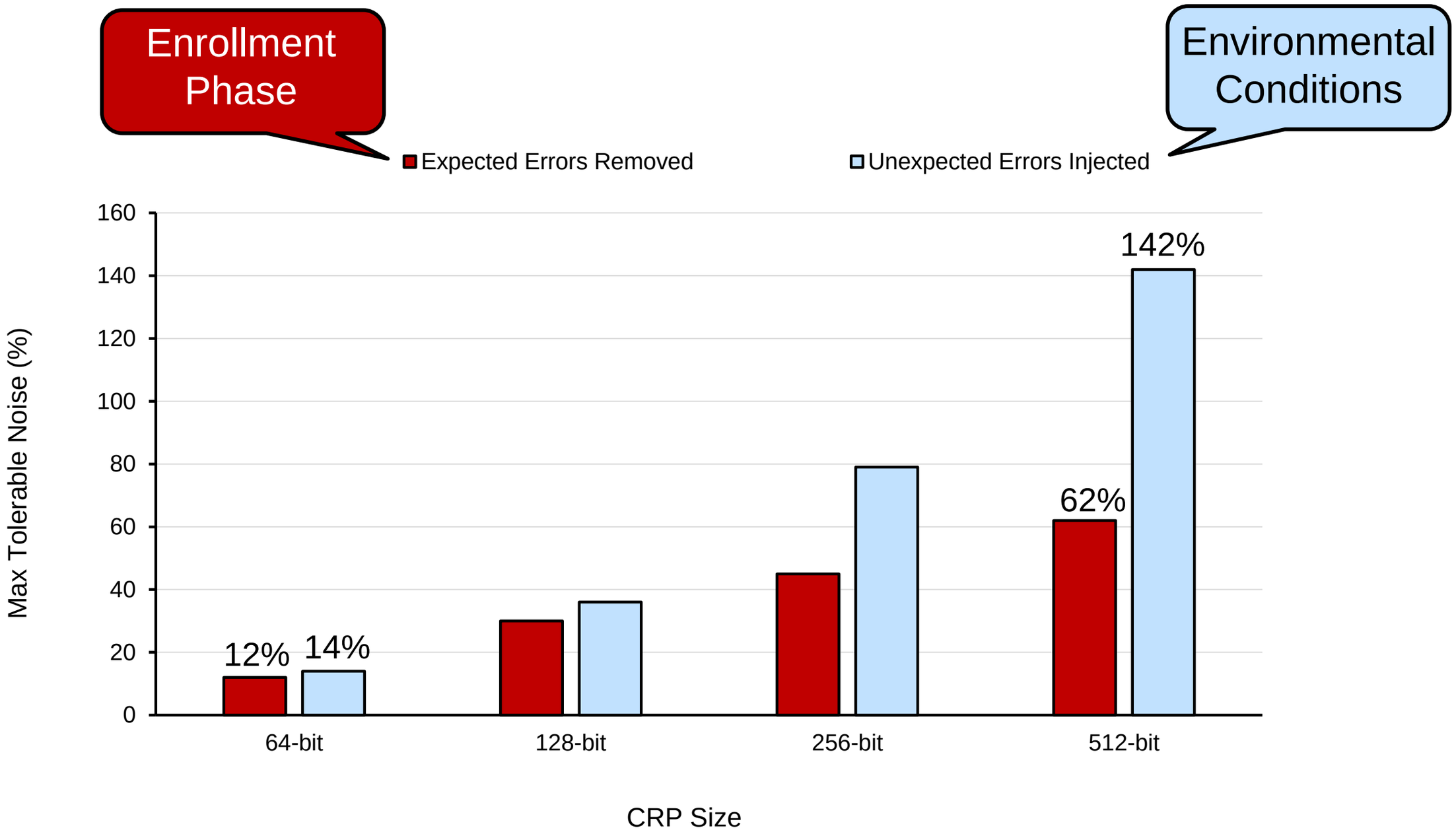
Resiliency to Noise



Resiliency to Noise



Resiliency to Noise



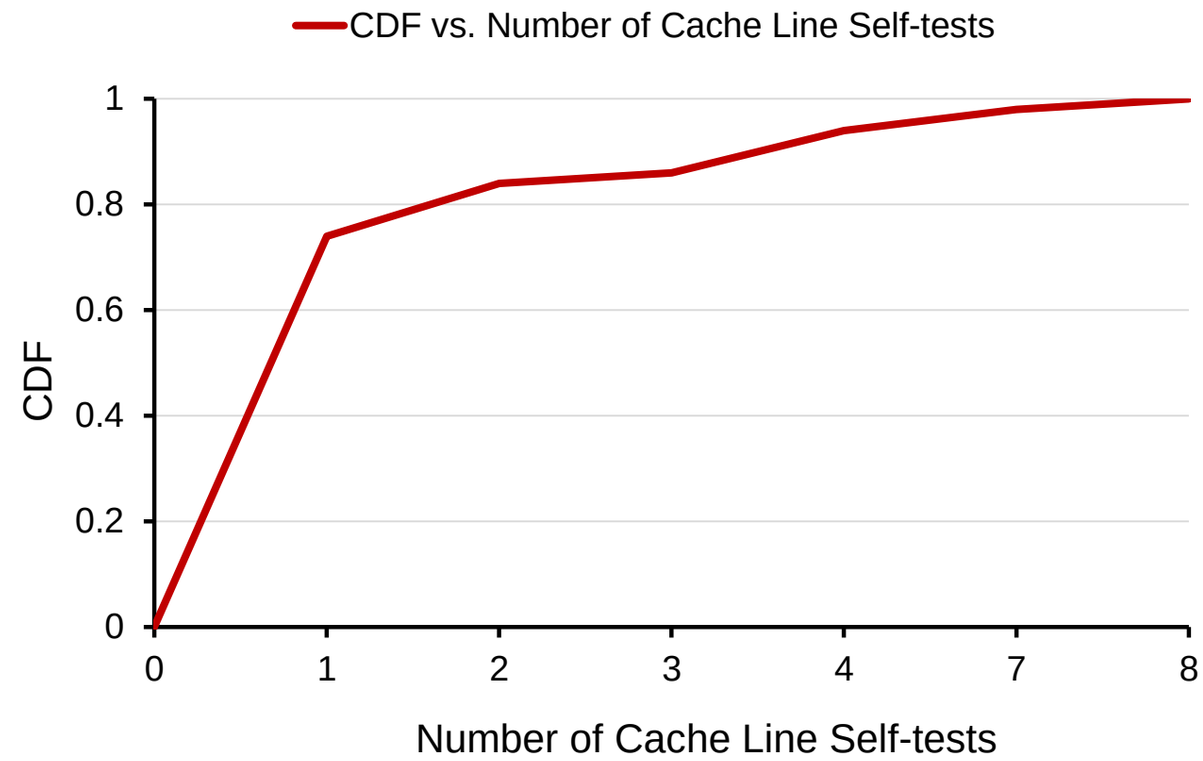
Repeatability and Performance

Repeatability and Performance

- Repeatable cache line errors

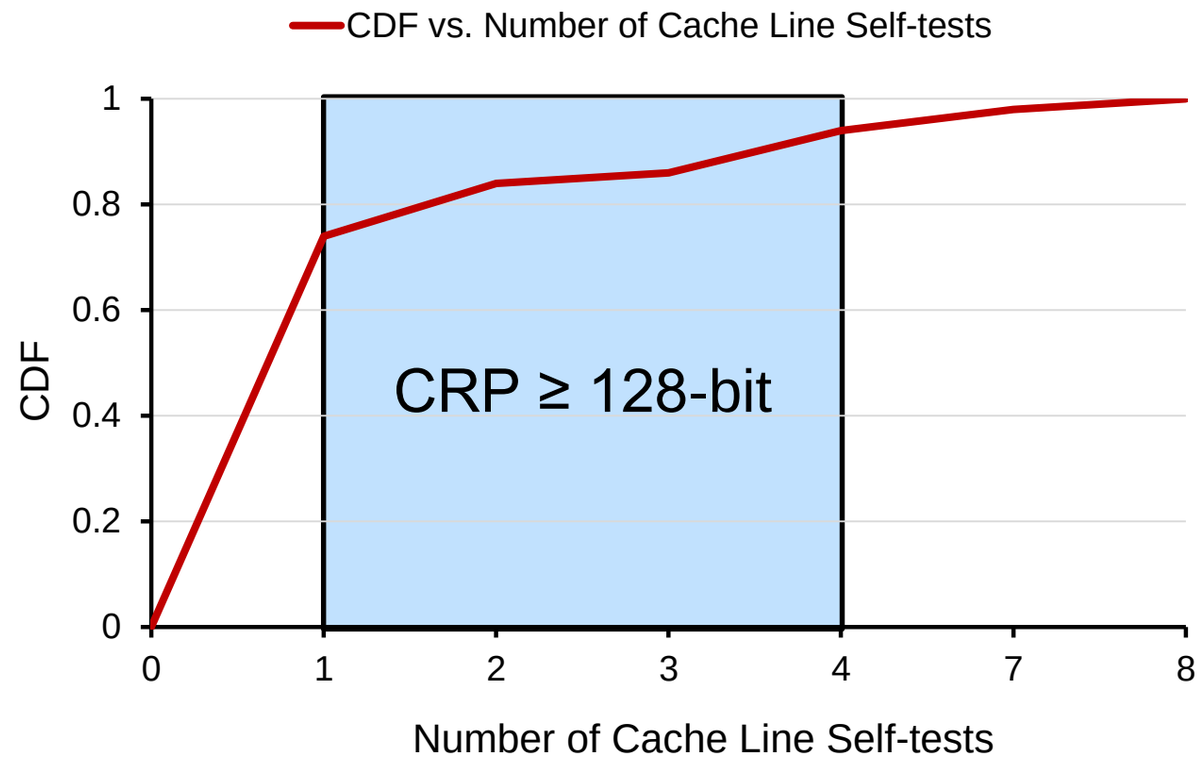
Repeatability and Performance

- Repeatable cache line errors



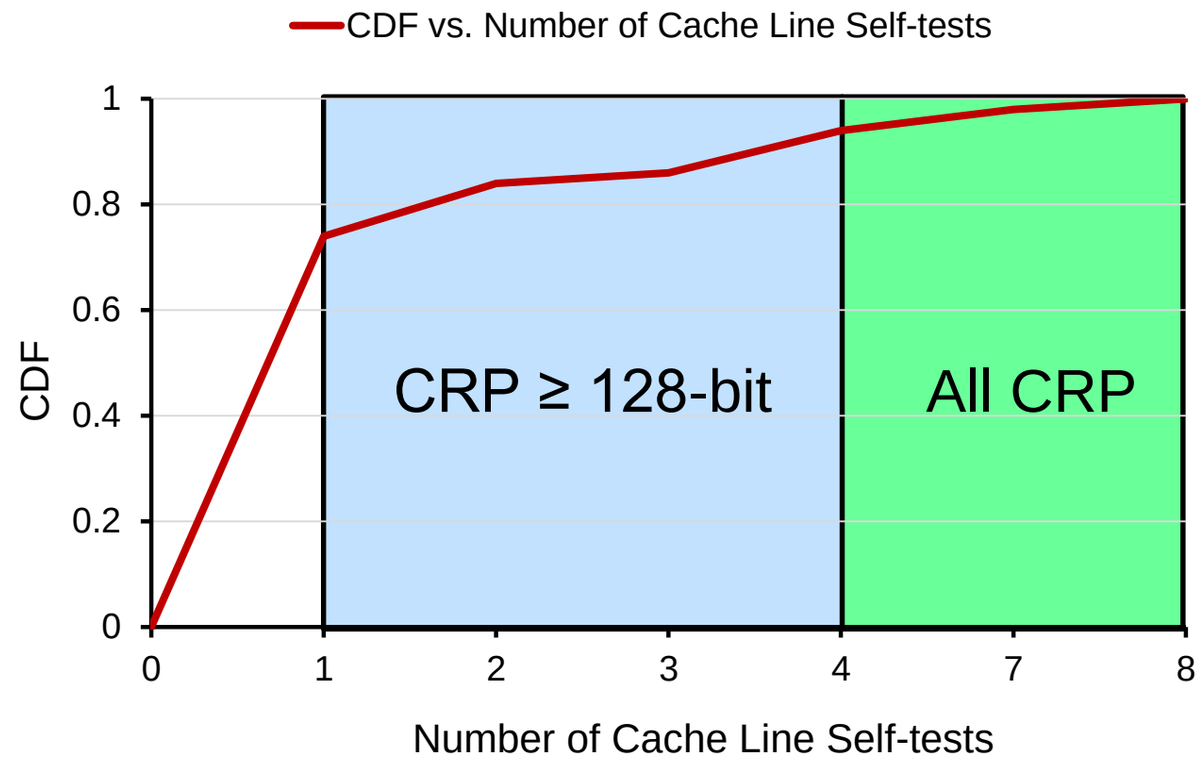
Repeatability and Performance

- Repeatable cache line errors



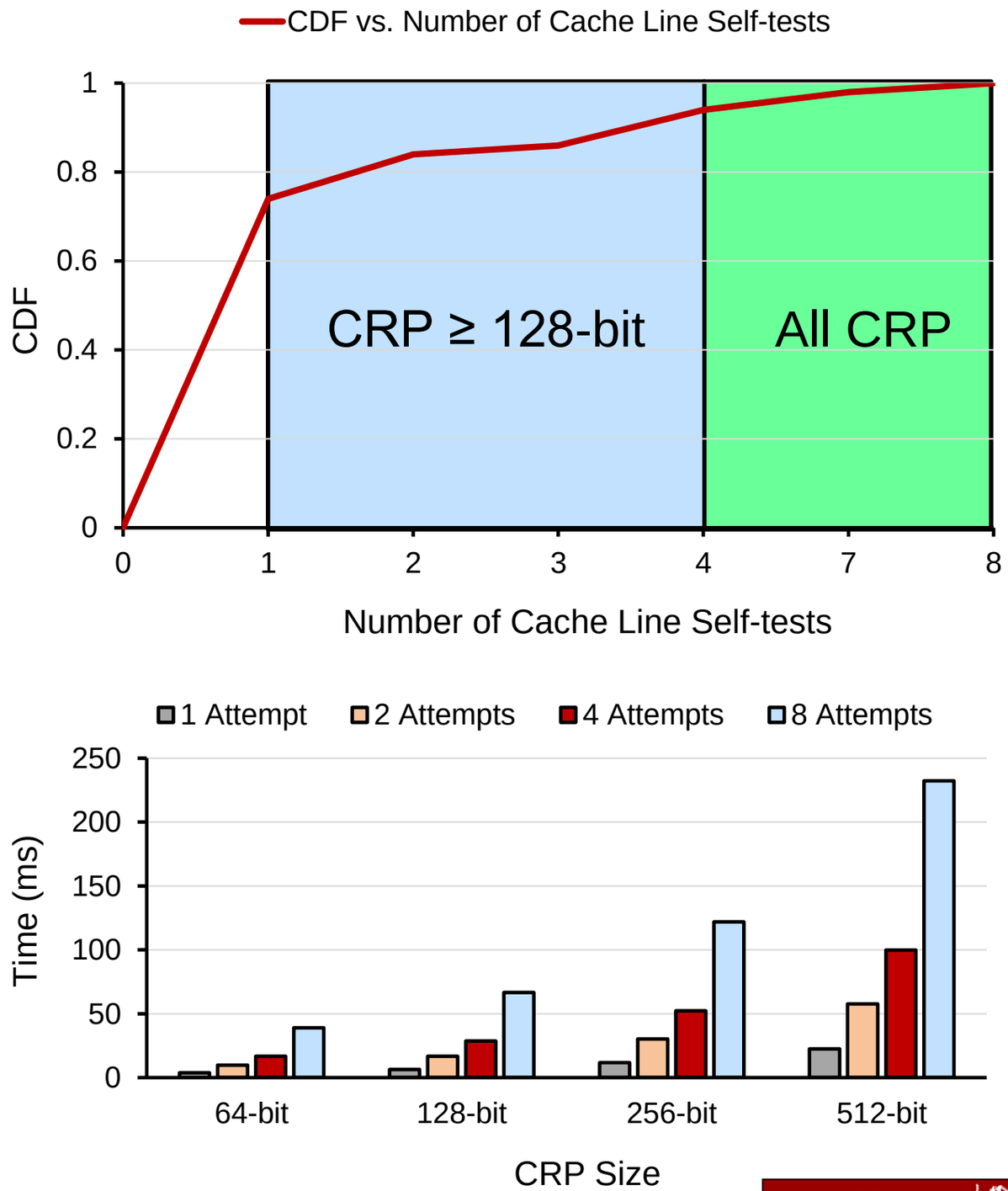
Repeatability and Performance

- Repeatable cache line errors



Repeatability and Performance

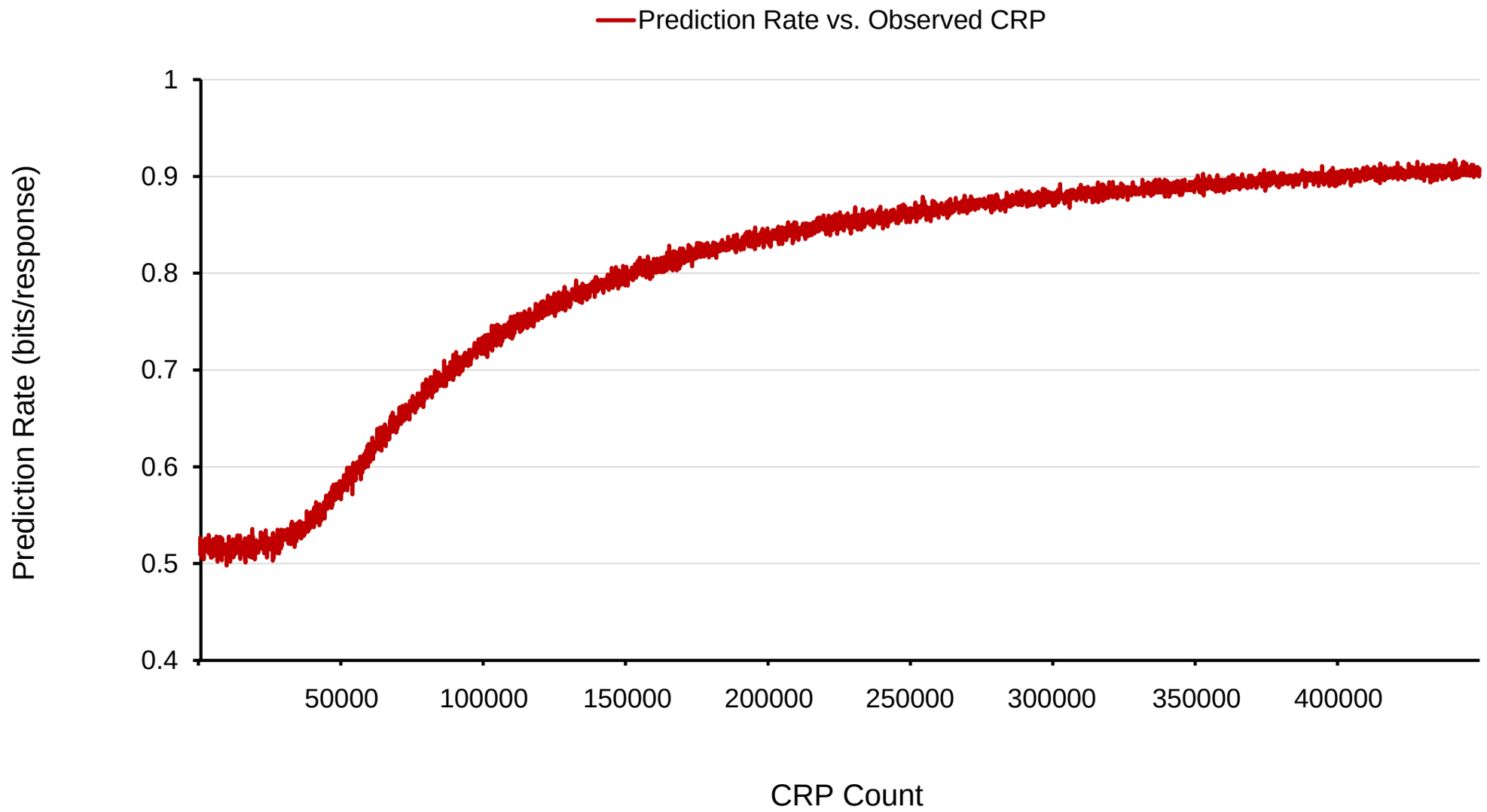
- Repeatable cache line errors



- Linear increase in runtime as a function of self-test attempts

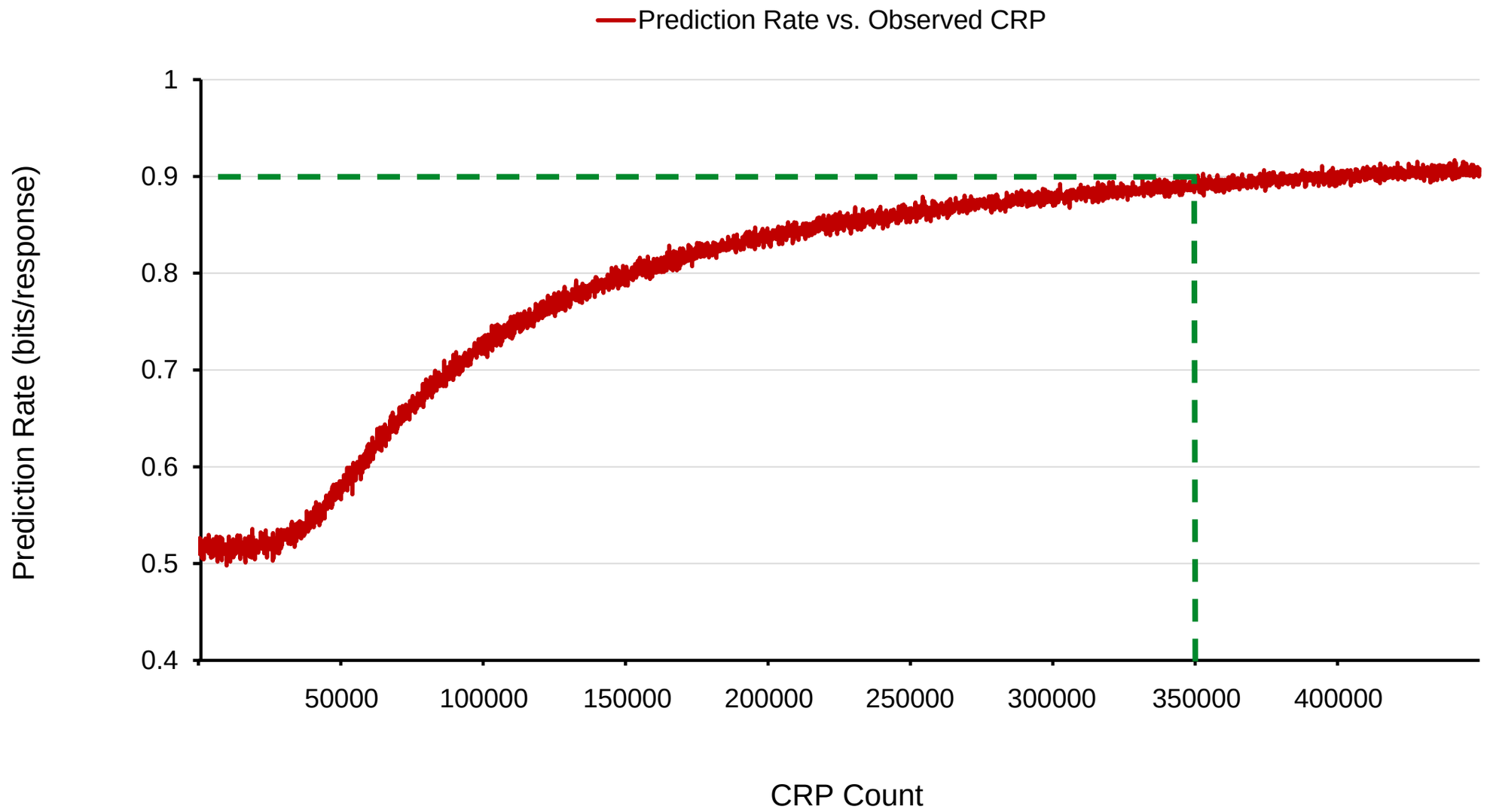
Model Building Attack Case Study

Model Building Attack Case Study



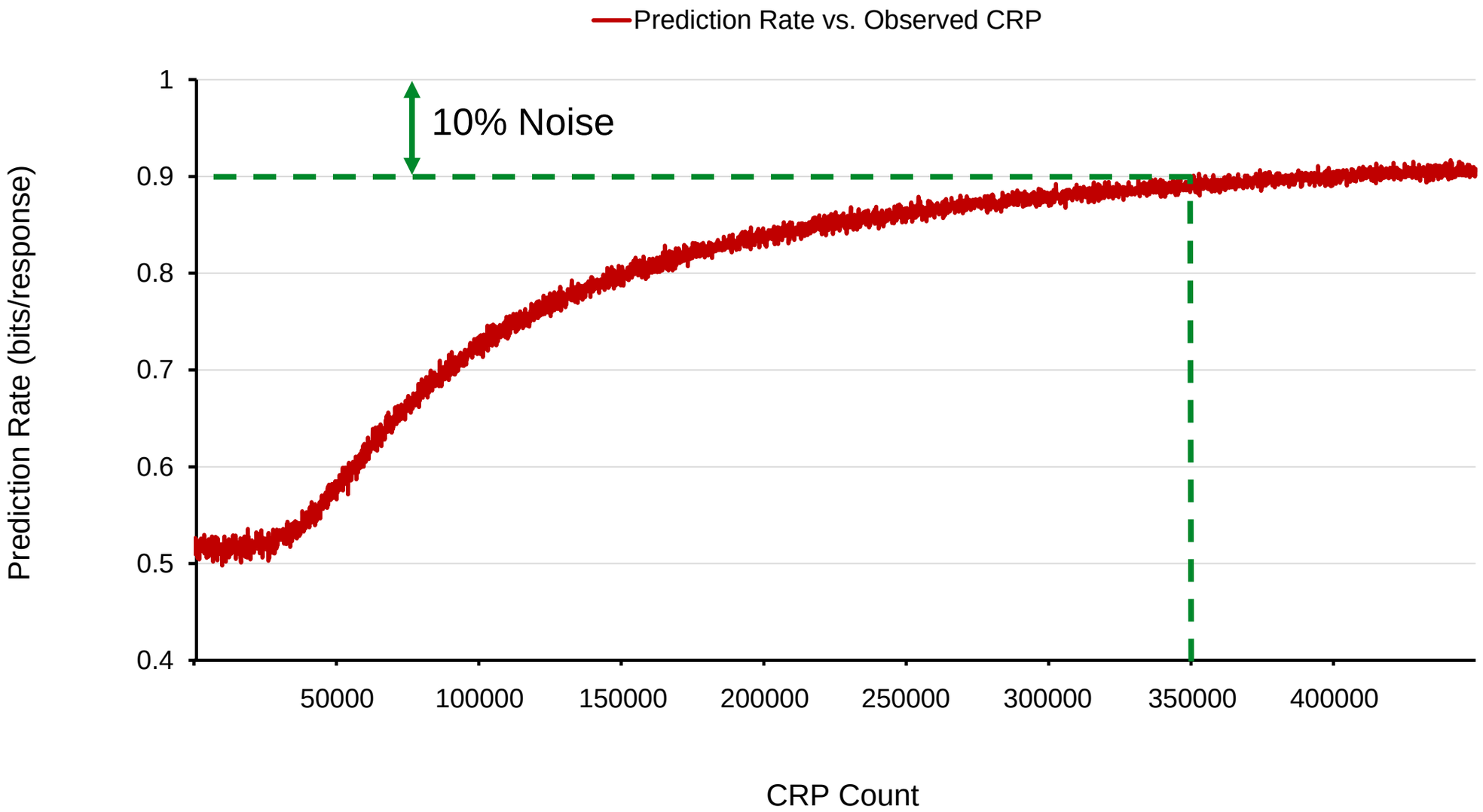
COMPUTER ARCHITECTURE RESEARCH LAB

Model Building Attack Case Study

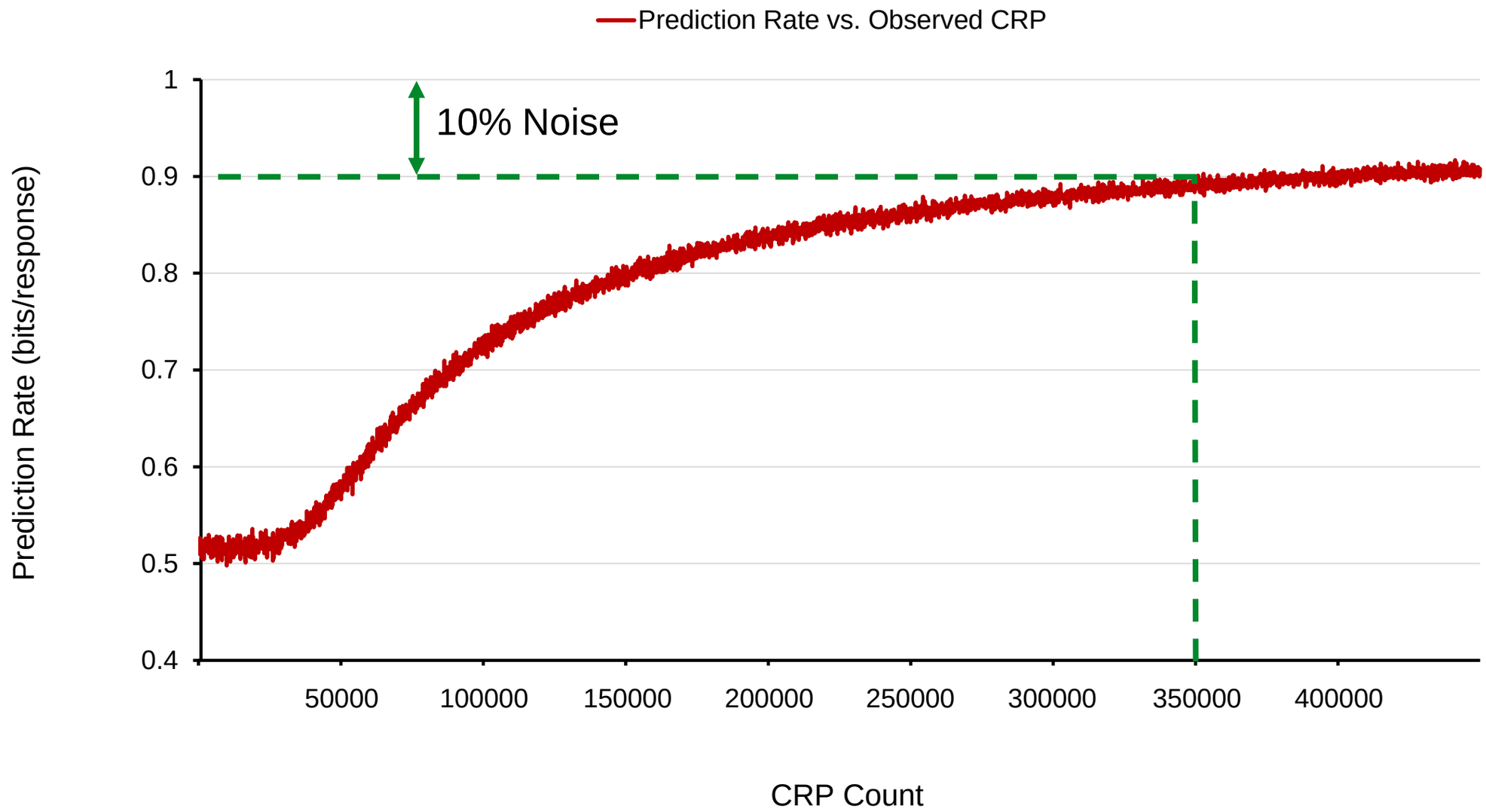


COMPUTER ARCHITECTURE RESEARCH LAB

Model Building Attack Case Study



Model Building Attack Case Study



Regenerate logical error map

Conclusion

- Observe that correctable errors in caches can be used as silicon fingerprints
- Introduce a challenge-response design that can sustain large number of authentications (10 year lifetime)
- Demonstrate robustness of technique to noise (up to 142%)
- Realize a proof-of-concept to show system is practical

Thank you!

Questions?

Authenticache: Harnessing Cache ECC for System Authentication

Anys Bacha and Radu Teodorescu

Department of Computer Science and Engineering

The Ohio State University

http://arch.cse.ohio-state.edu

THE OHIO STATE UNIVERSITY

COMPUTER ARCHITECTURE RESEARCH LAB