A Survey on Hardware Security Techniques Targeting Low-Power SoC Designs

Alan Ehret[1], Dr. Karen Gettings[2],
Bruce R. Jordan Jr.[2], Dr. Michel A. Kinsy[1]
[1] Boston University, Boston, MA
[2] MIT Lincoln Laboratory, Lexington, MA

BOSTON UNIVERSITY
Department of Electrical & Computer Engineering

Introduction

- Embedded systems are integral part of modern life
- More devices are being made “Smart”
- Connected systems provide opportunities for theft, denial of service, etc.

Smart Trashcan

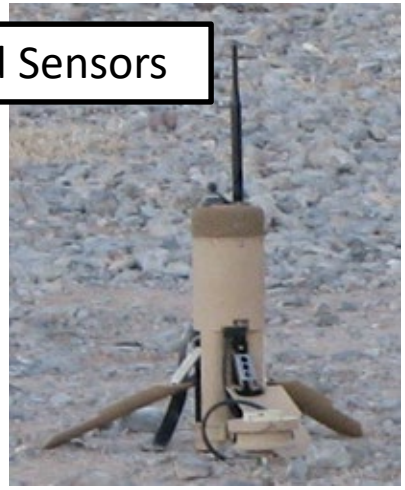
Wireless Camera

Cashless Tolls

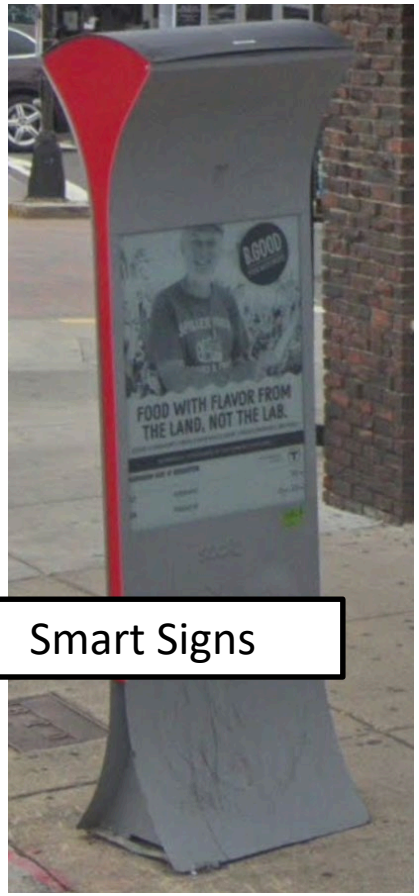
Motion Sensor

Security Challenges

- IoT deployment environment
  - Unattended
  - Physical access available
- Attackers may have unhindered access to devices for long durations
- Tight power/area budgets limit overhead available for security



Ground Sensors

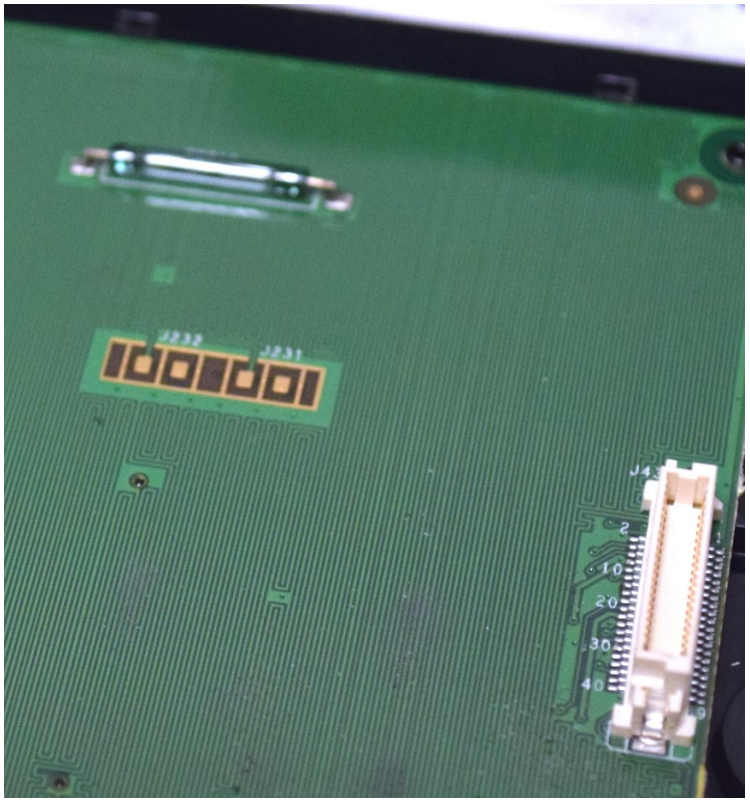


Smart Signs

Parking Meters

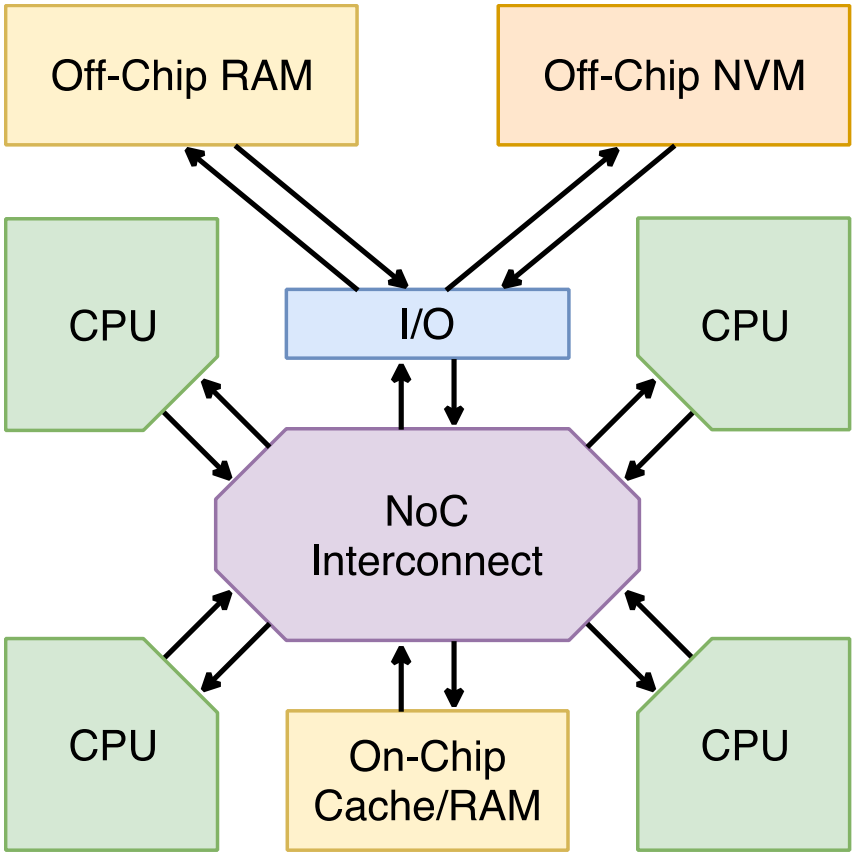
Why Not Anti-Tamper?

- Increases size, weight, power consumption
  - Must always check for intrusions
- Adding security directly to IC hardware is more efficient
  - Circuit level security modules may be shut down with device



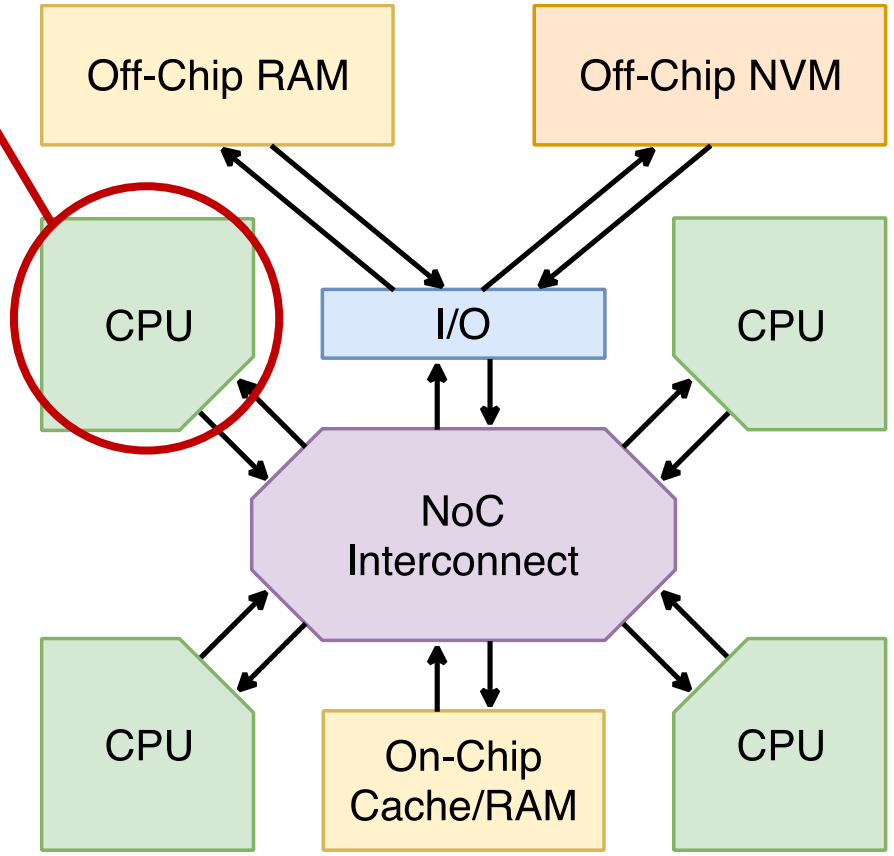
Point of sale device anti-tamper PCB – hackaday.com

Defense Categories



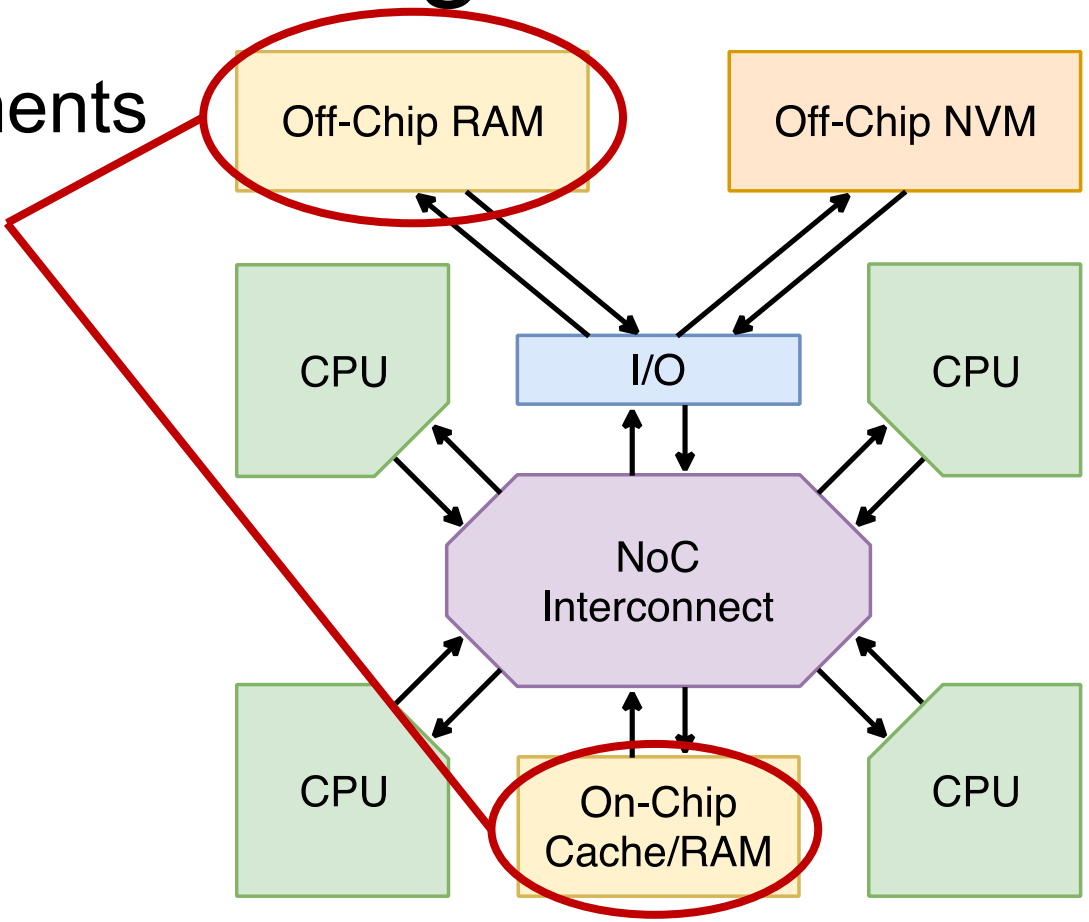
Defense Categories

- Processing Elements



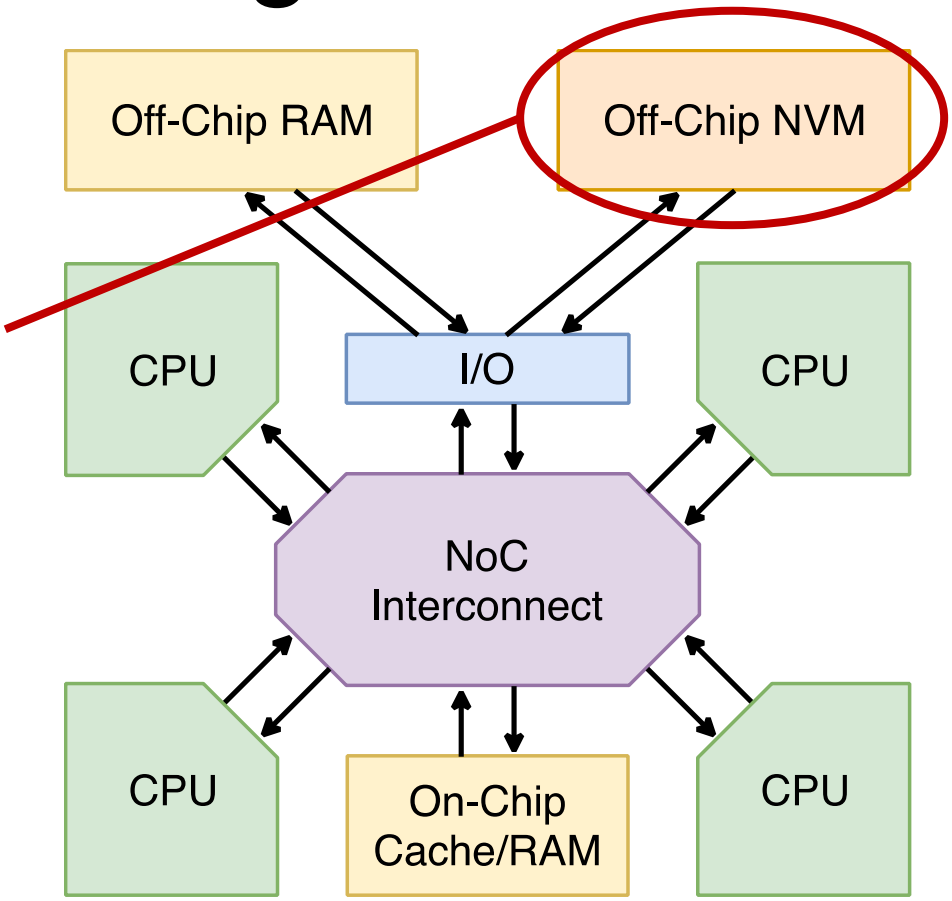
Defense Categories

- Processing Elements
- Volatile Memory



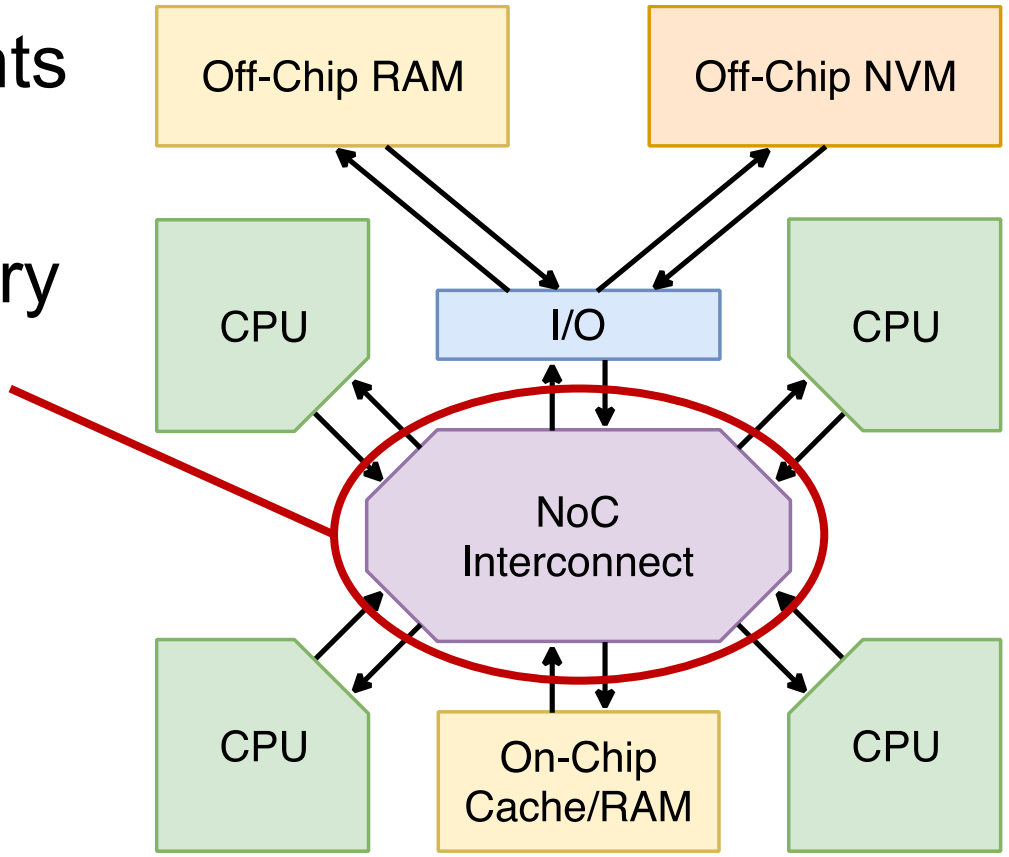
Defense Categories

- Processing Elements
- Volatile Memory
- Non-Volatile Memory

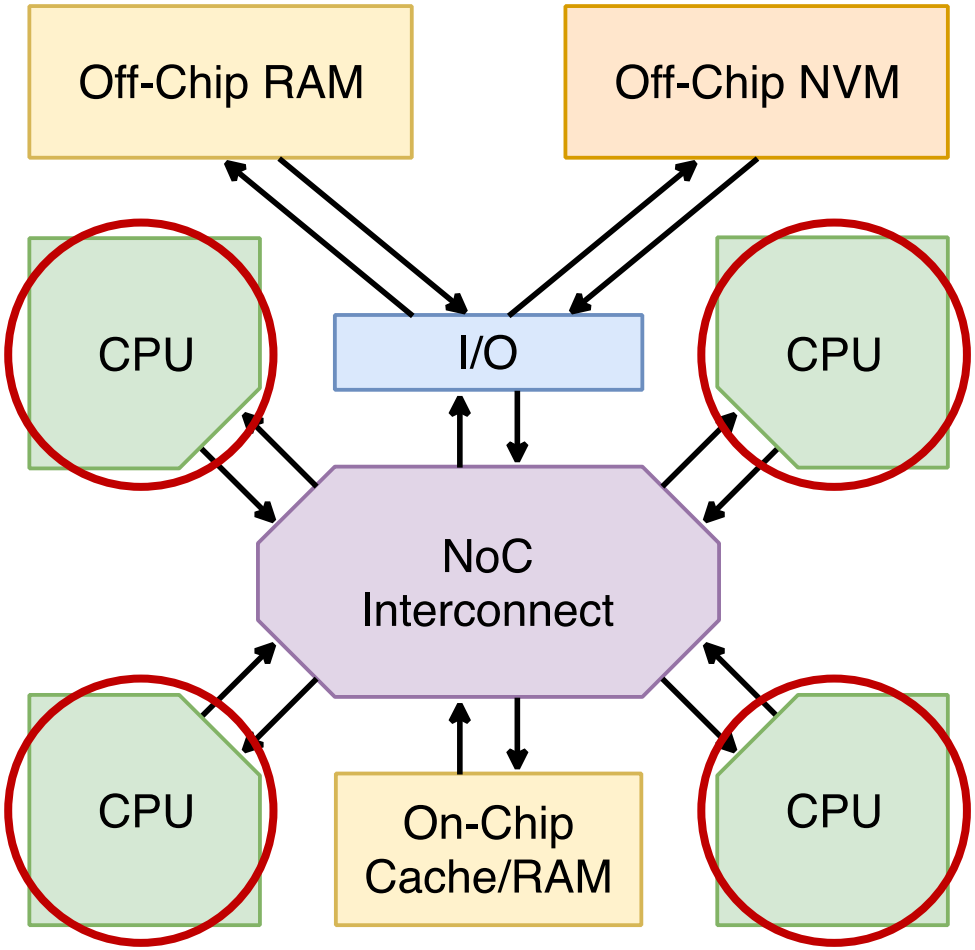


Defense Categories

- Processing Elements
- Volatile Memory
- Non-Volatile Memory
- NoC Interconnects

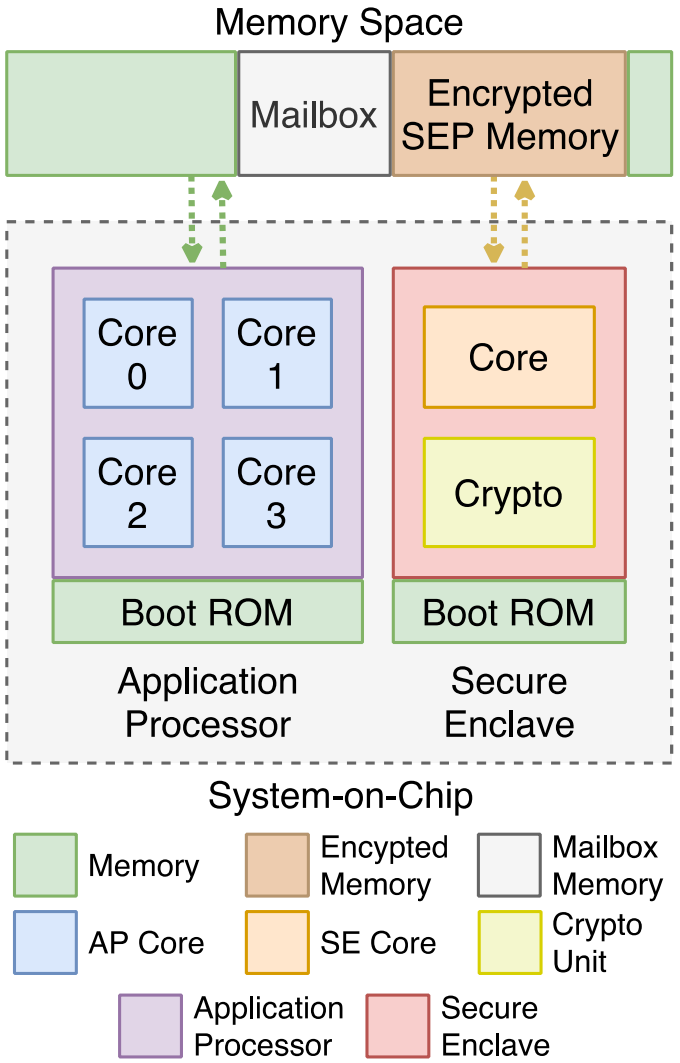


Processing Element Defenses



Secure Enclaves

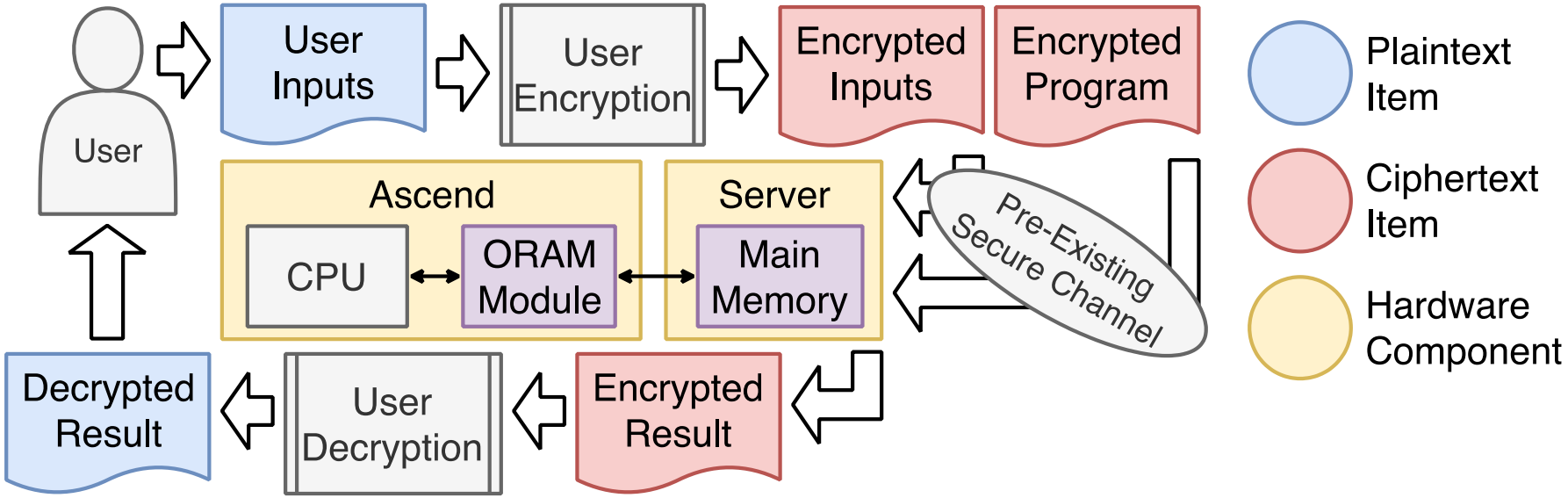
- Dedicated hardware core
  - Isolate sensitive data
- Commercial examples
  - Apple SEP
  - ARM TrustZone
- Mitigates:
  - User and kernel software vulnerabilities
  - Application Processor side-channels



[1] Apple, "Ios security."

Execution Obfuscation - Ascend

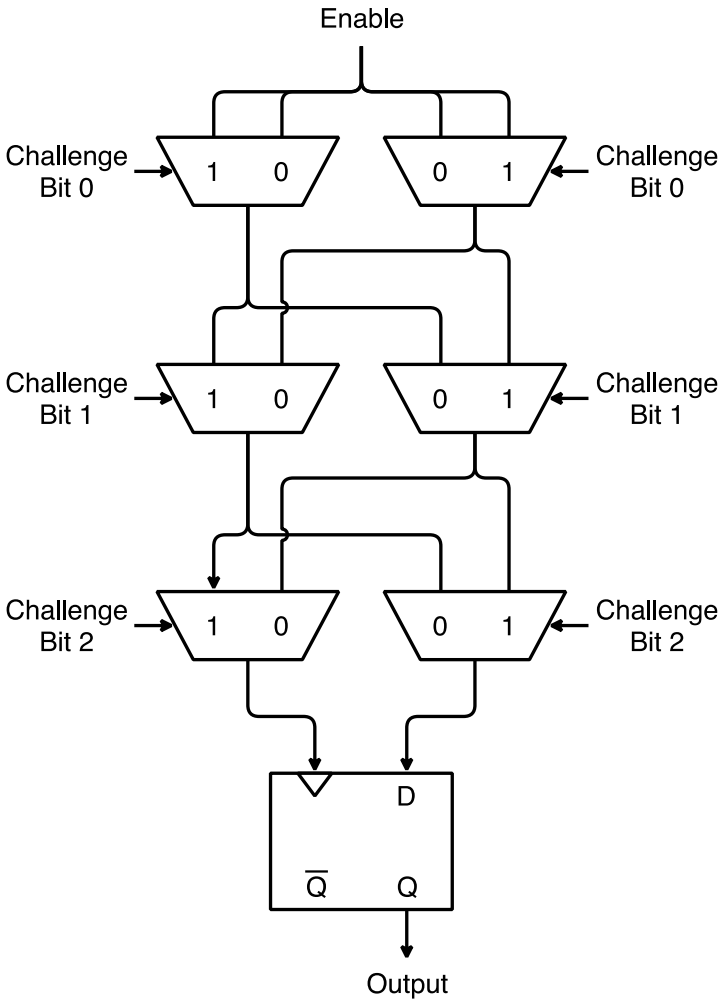
- Mitigate side-channels in cloud environment
  - Power, I/O, Timing
- Honest but curious cloud provider



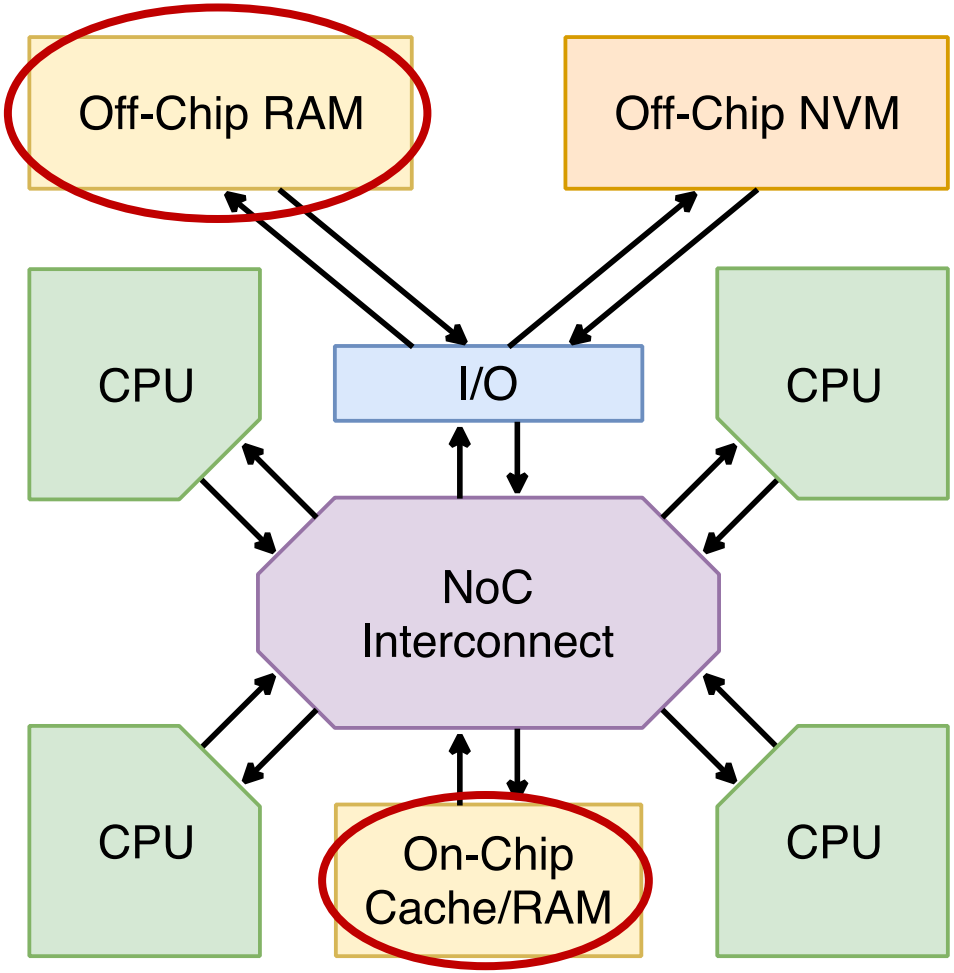
[2] C.W.Fletcher, et al. "A secure processor architecture for encrypted computation on untrusted programs."

Physical Unclonable Functions

- Use variation in manufacturing to uniquely ID a device
- Prevent impersonation of hardware
  - Silicon Fingerprints
- Useful for hardware based:
  - Key generation/storage
  - Device authentication



Volatile Memory Defenses



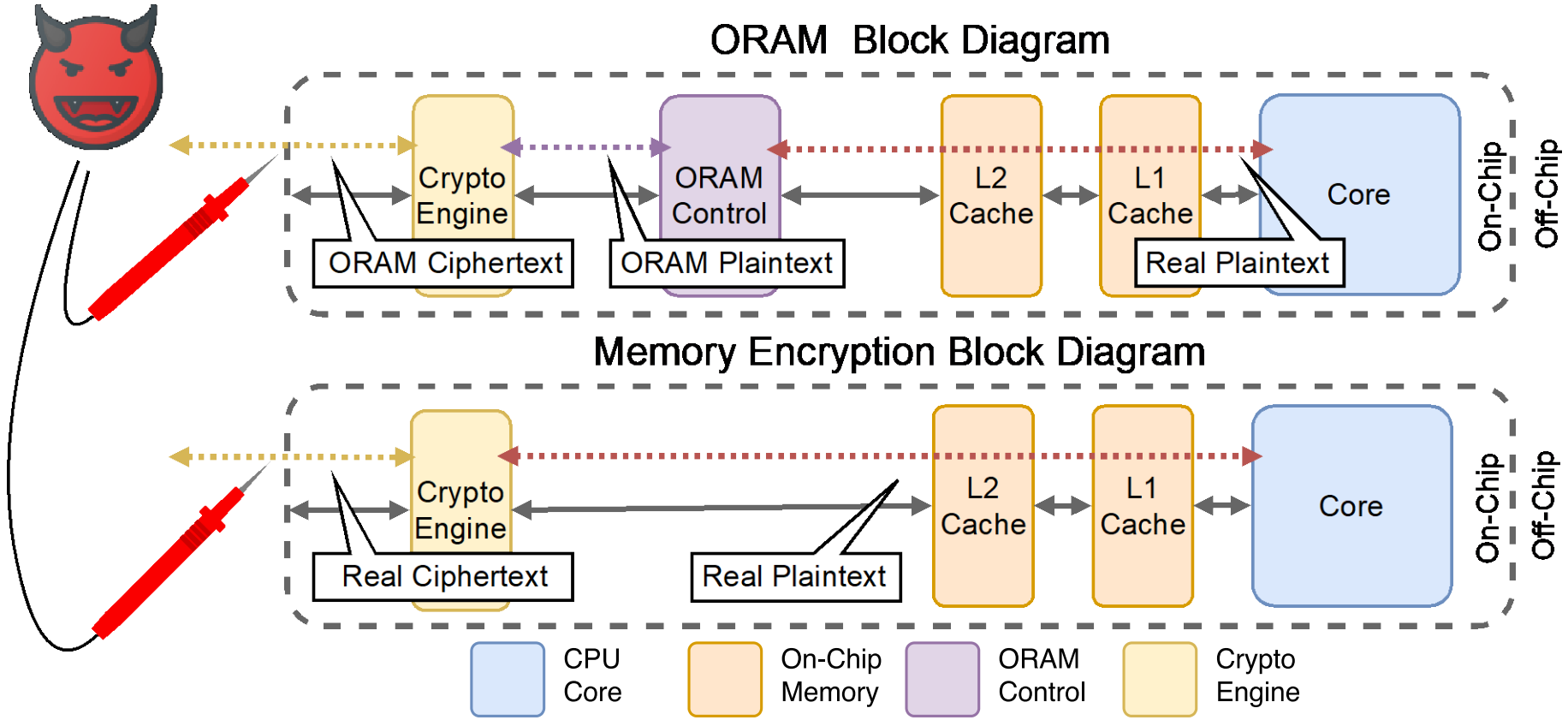
Memory Encryption & ORAM

- Prevent main memory leaks
- ORAM has greater overhead than memory encryption

	Obfuscate Contents	Obfuscate Addresses	Obfuscate Timing*
Memory Encryption	✓		
Oblivious RAM	✓	✓	✓

*ORAM does not necessarily obfuscate timing but some implementations (see Ascend in [9]) add this functionality to obfuscate all aspects of the memory.

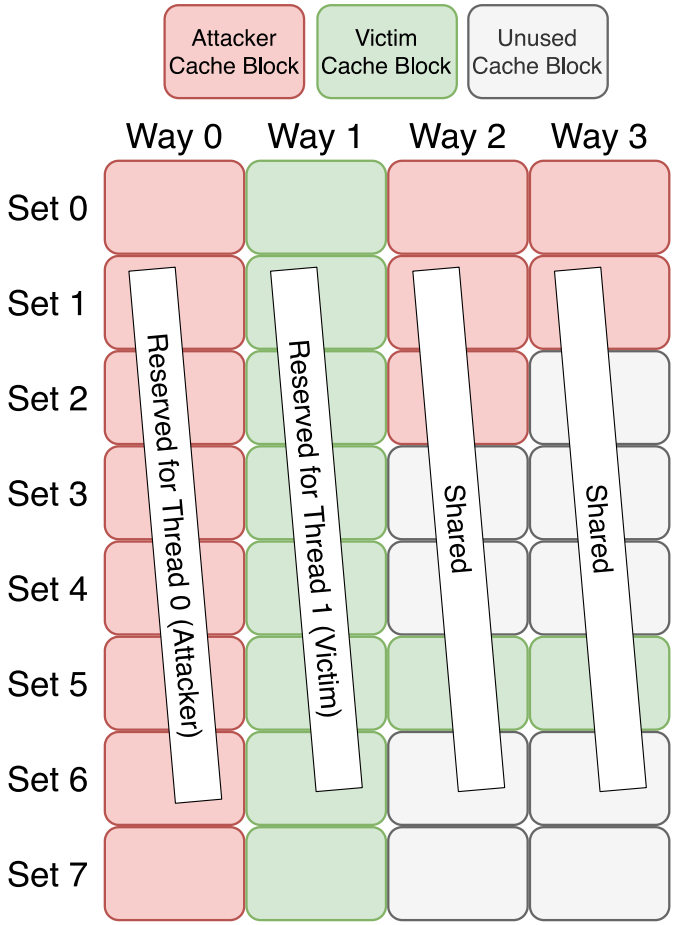
Memory Encryption & ORAM



[3] E. Stefanov, et al. "Path oram: an extremely simple oblivious ram protocol,"
[4] D. Kaplan, et al. "Amd memory encryption,"

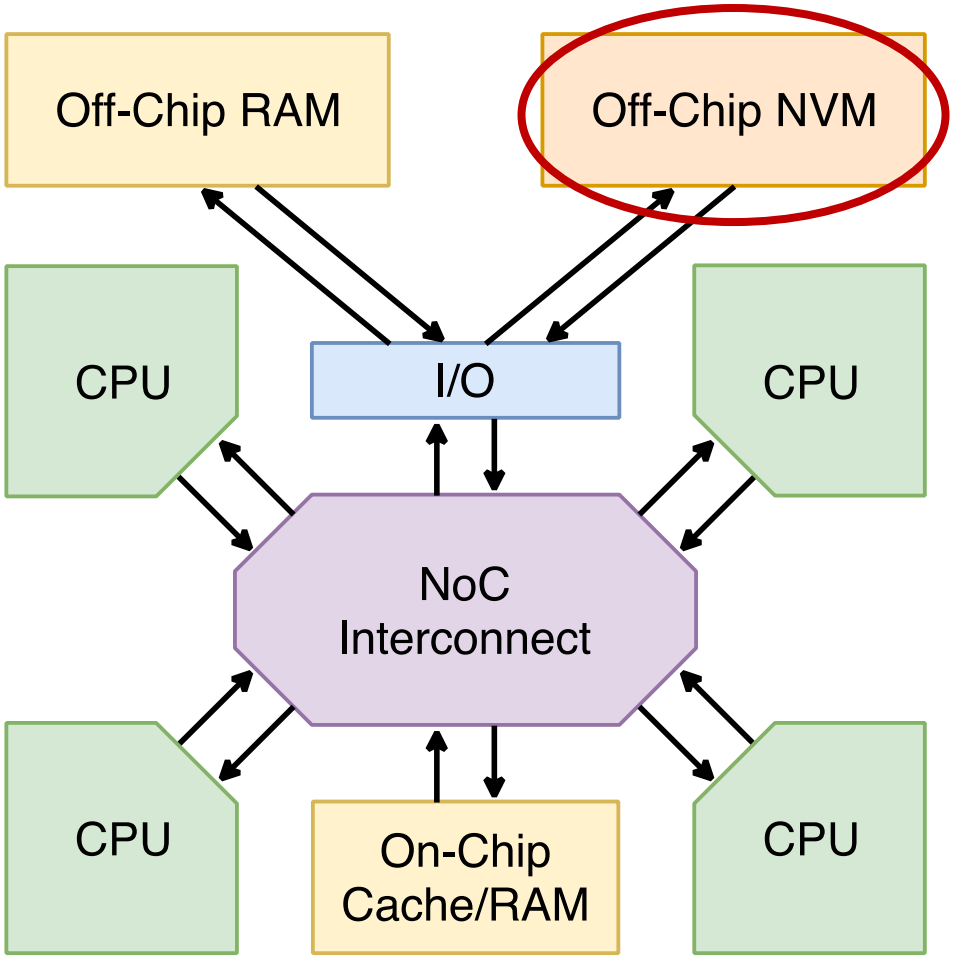
Non-Monopolizable Caches

- Reserve cache ways for threads of execution
  - Low hardware overhead
  - Moderate performance overhead
- Mitigate cache timing side-channels
  - Prevent threads from evicting each other's data



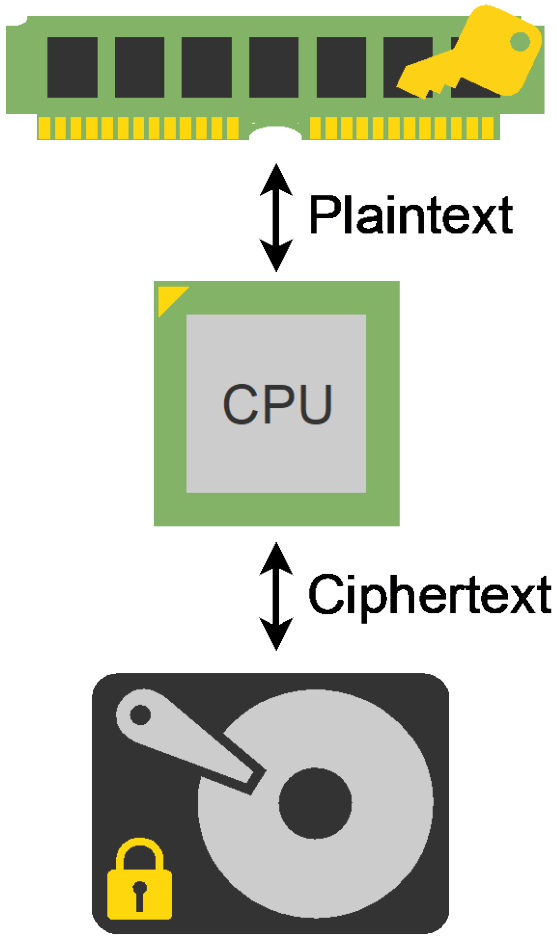
[5] L. Domnitser, et al. "Non-monopolizable caches: Low-complexity mitigation of cache side channel attacks."

Non-Volatile Memory Defenses



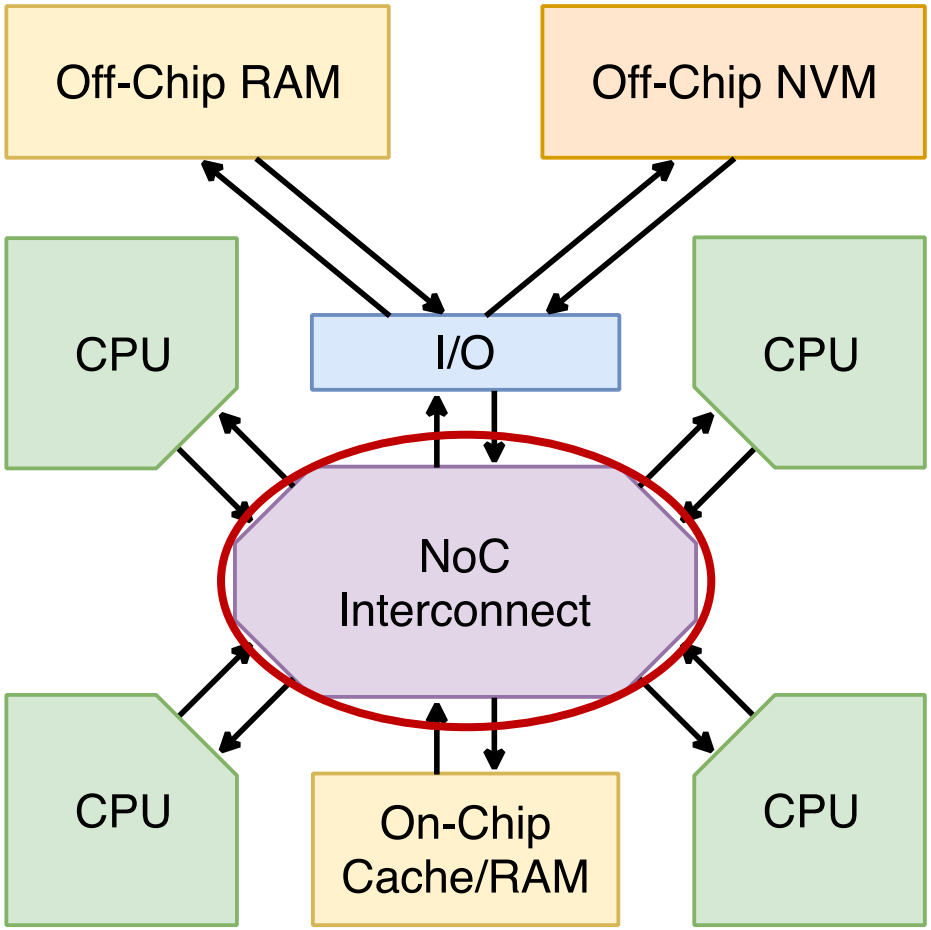
Full Disk Encryption

- Protects Data-at-Rest
- Encryption key stored in RAM
- Data on disk protected while machine is off
- Key could be leaked by side channel
- Self Encrypting Drives (SED) vulnerable to Hot-Swap attacks
  - Requires physical access



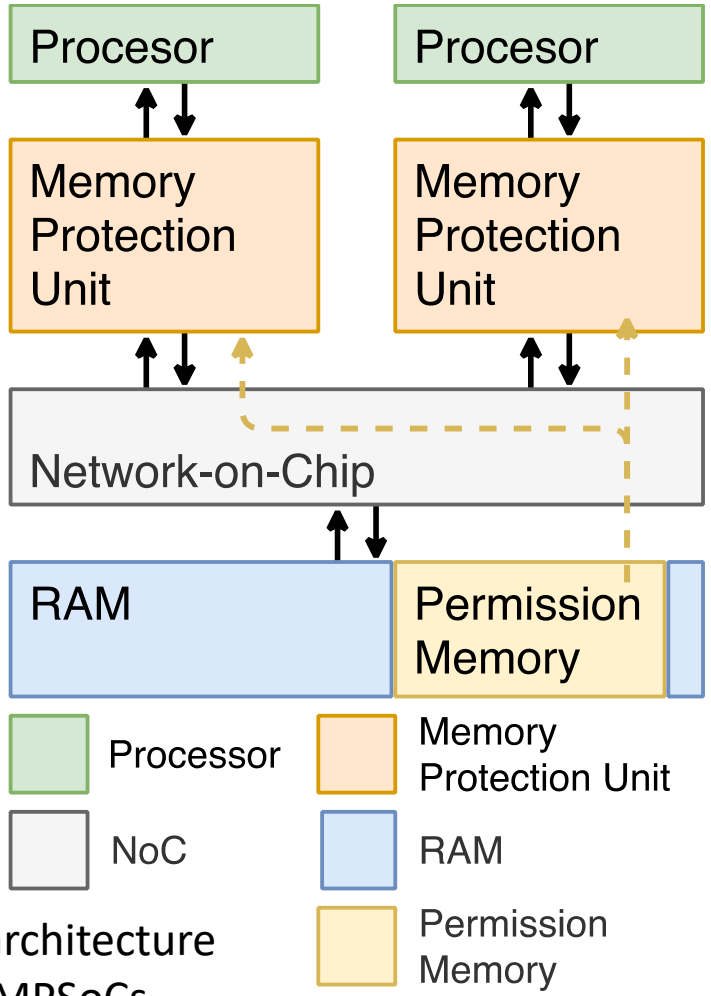
[6] T. Muller, et al., "Self-encrypting disks pose self- decrypting risks,"

Network-on-Chip Defenses



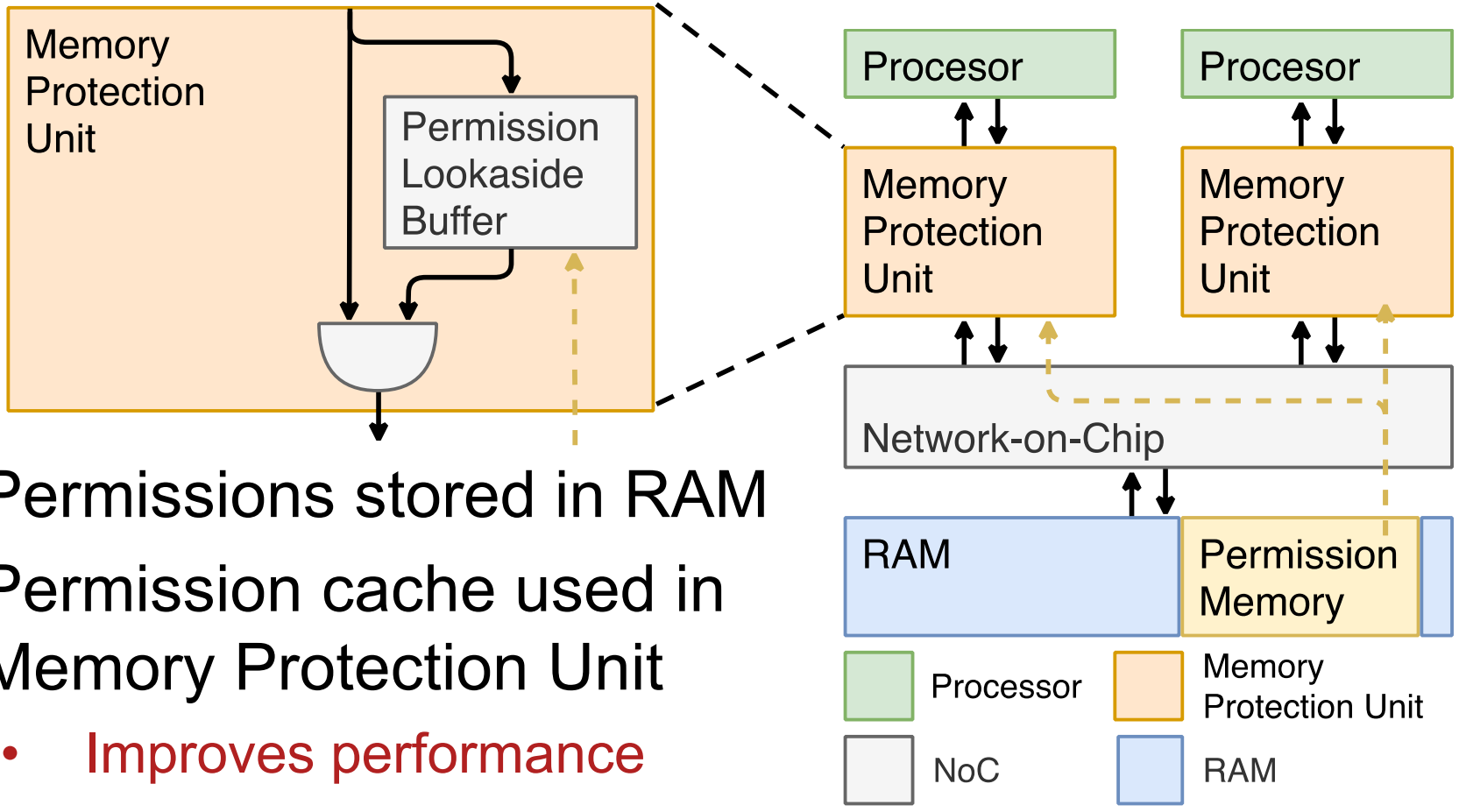
NoC Interconnect

- Use IP from different sources
- Run multiple applications with different trust levels
  - Must prevent misuse of hardware resources
- NoC performs permission checks on traffic
  - Support virtual isolation of software stacks



[7] Porquet J., et al. "NoC-MPU: A secure architecture for flexible co-hosting on shared memory MPSoCs.

NoC Interconnect



- Permissions stored in RAM
- Permission cache used in Memory Protection Unit
  - Improves performance

[7] Porquet J., et al. "NoC-MPU: A secure architecture for flexible co-hosting on shared memory MPSoCs.

Defense Comparisons

	Secure Enclave	Execution Obfuscation	Memory Encryption	Limited Use Memory	Disk Encryption	Cache Arch.	ORAM	PUF	NoC Isolation
Access-Based Cache Side-Channel	✓	✓				✓	✓		
Power-Based Cache Side-Channel		✓				✓	✓		
Binary Reverse Engineering		✓							
In-Memory Data Theft	✓		✓				✓		✓
On-Chip Data Theft	✓					✓			✓
Data at Rest Theft				✓	✓				
Counterfeiting								✓	
On-Chip DoS									✓

Conclusion

- Low-power connected systems vulnerable to a variety of attacks
  - Increased potential for theft, denial of service
- No one-size-fits-all solution
  - Must analyze threat model for each system
  - Consider size, weight, power budgets

References

[1] Apple, “Ios security,” apple.com/business/docs/iOS Security Guide.pdf.

[2] C.W.Fletcher, M.v.Dijk, and S.Devadas, “A secure processor architecture for encrypted computation on untrusted programs,” in Proceedings of the seventh ACM workshop on Scalable trusted computing. ACM, 2012, pp. 3–8.

[3] E. Stefanov, M. Van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas, “Path oram: an extremely simple oblivious ram protocol,” in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013, pp. 299–310.

[4] D. Kaplan, J. Powell, and T. Woller, “Amd memory encryption,” White paper, 2016.

[5] L. Domnitser, A. Jaleel, J. Loew, N. Abu-Ghazaleh, and D. Ponomarev, “Non-monopolizable caches: Low-complexity mitigation of cache side channel attacks,” ACM Transactions on Architecture and Code Opti- mization (TACO), vol. 8, no. 4, p. 35, 2012.

[6] T. Muller, T. Latzo, and F. C. Freiling, “Self-encrypting disks pose self- decrypting risks,” in the 29th Chaos Communinication Congress, 2012, pp. 1–10.

[7] Porquet, J.; Greiner, A.; Schwarz, C. NoC-MPU: A secure architecture for flexible co-hosting on shared memory MPSoCs. In Proceedings of the 2011 Design, Automation & Test in Europe, Grenoble, France, 14–18 March 2011; pp. 1–4.