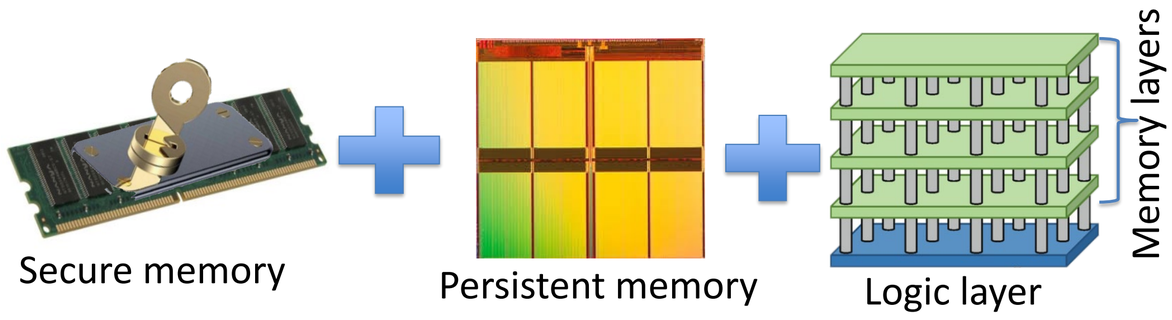
# Towards Smart and Secure Non-Volatile Memory

Huiyang Zhou, North Carolina State University

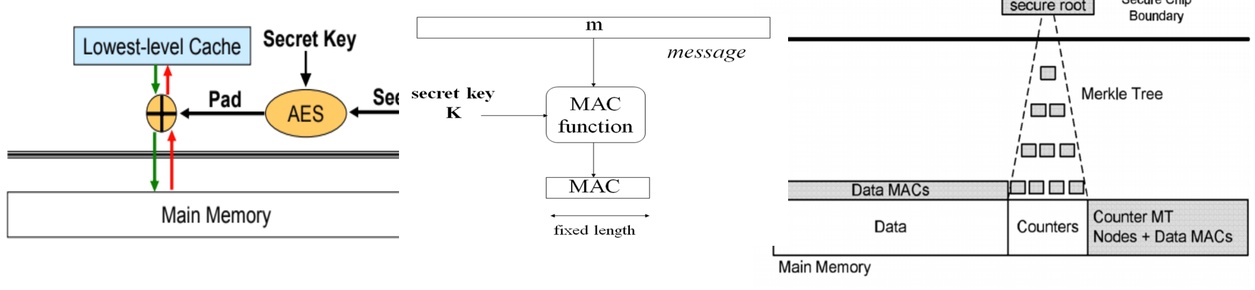
http://people.engr.ncsu.edu/hzhou

## Objective: Smart Secure Persistent Memory

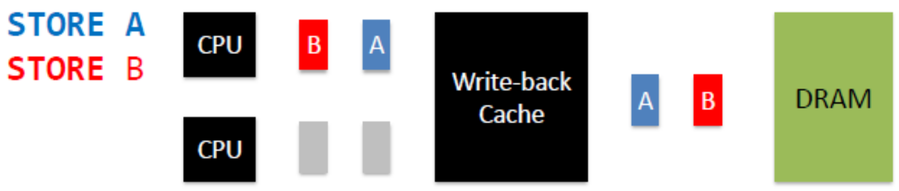



## Key Challenges

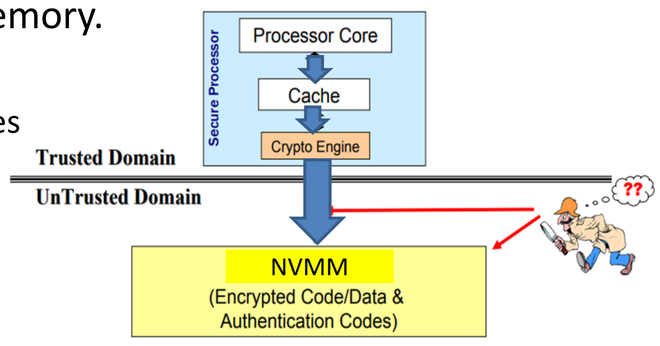
- Secure memory incurs high performance overhead due to security-related meta data, i.e., counters, MAC, and integrity tree.



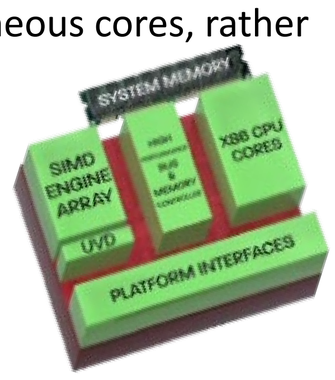

- Memory persistency models
  - Non-volatile memory (NVM) keeps persistent data and expects crash recoverability that requires memory persistency models, which in turn requires store ordering. Store ordering, however, may lead to high memory traffic.




- Secure NVM would suffer from much higher performance overhead than volatile memory.
  - High write traffic
  - Frequent meta data updates




- Secure NVM is shared among heterogeneous cores, rather than CPU alone.
  - Shared memory hierarchy
  - Persistency models needed in all cores




## Impact on society

- Security and crash recoverability are key to safety-critical systems such as autonomous driving

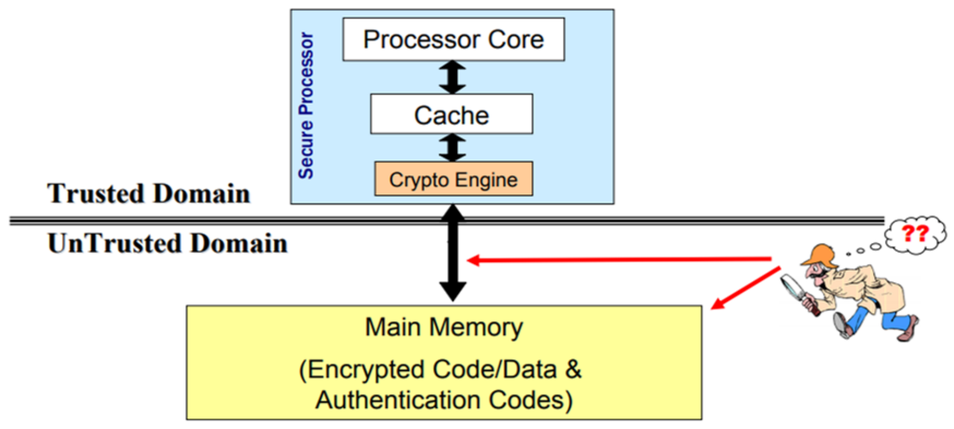
- Also important for cloud servers hosting potentially malicious users

## Education & Outreach

- Graduate student advising and curriculum development
  - A total of 6 graduate students participated in the project so far
  - Secure processor design and memory persistency models become part of a graduate-level course on advanced microarchitecture.
- Recruiting students from under-represented groups

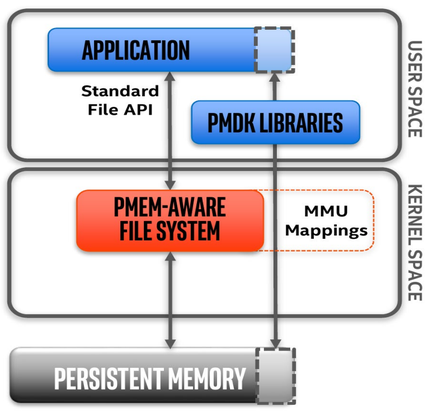
Attack model:

- Adversary has physical access to memory chip
- Processor chip forms trust boundary

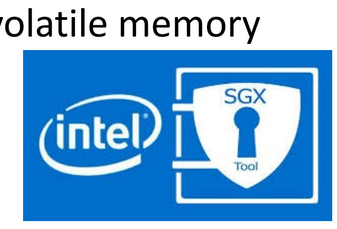



## Scientific Impact

- NVMM is here


- Secure memory is currently for volatile memory




- Secure NVMM is the next step
  - Recoverability => Subtle interactions between security and persistency models
  - Heterogeneous cores => Re-architecting CPU-based persistency models
  - High Performance => Optimizations to overcome bottlenecks

## Work in-progress

- Exploring Memory Persistency Models for GPU (PACT'19)
  - Architectural support for strict & epoch persistency models
  - Programming support for different scopes of epochs
  - Optimizations for durable transactions: reducing memory traffic due to logging

| Persistency Models | Strict Persistency | Relaxed Persistency | | |
|---|---|---|---|---|
| | | Kernel-Level Epoch | CTA-Level Epoch | Loop-Level Epoch |
| Architectural Support | clwb/clflush(opt)/store.wt; sfence; pcommit | l2wb; DeviceSynchronization | clwb/clflush(opt); store.wt; sfence; pcommit; l2wb | clwb/clflush(opt); store.wt; sfence; pcommit; l2wb |
| Suitable Kernel | All | Short-running kernels | Long-running kernels with short-running CTAs | Long-running kernels with long-running CTAs |

- Streamlining Integrity Tree Updates for Secure NVMM
- Scatter-and-Gather Revisited: High-Performance Side-Channel-Resistant AES on GPUs (GPGPU'19)

## Other broader impacts

- Open-source code release: https://github.com/zhenlin36/scatter_gather_aes_cuda