“Instant On” Secure Recovery of Non-Volatile Main Memory Systems

Samuel Thomas* — Brown University, samuel_thomas@brown.edu

Tamara Lehman — CU Boulder, tamara.lehman@colorado.edu

R. Iris Bahar — Brown University, iris_bahar@cs.brown.edu

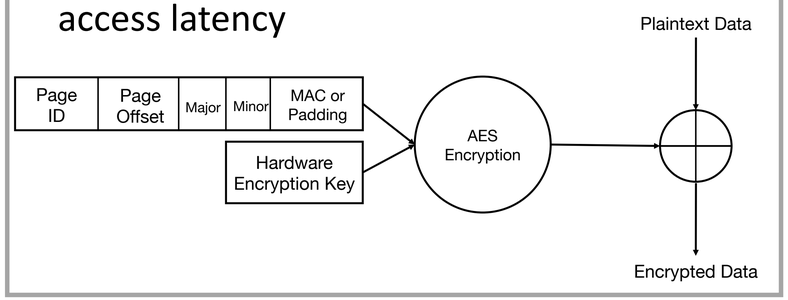
Joseph Izraelevitz — CU Boulder, joseph.izraelevitz@coloardo.edu

BROWN

University of Colorado Boulder

0. Counter Mode Encryption

- **Metadata Counters** compose One-Time Pads used for O(1) encryption
- Encryption key is unique to the **state** in memory
- Hardware encryption done in the memory controller, so latency **hidden** by memory access latency



0. Bonsai Merkle Tree

- **Integrity verification** mechanism
- Root of the tree reflects **overall state** of the entire tree and its contents

With Counter-Mode Encryption

- Leaves made up of metadata counters
- Root reflects state of all counters

For Secure Recovery

- If computed root matches stored root, then metadata is untampered
- Data values checked done by software

1. Motivation

Attack on Non-Volatile Memory

- an attacker could cause a **power failure**
- the data remains **in memory** through power failure
- the attacker can **tamper** with the **memory contents** which can cause application failure

State of the Art Solutions

- tree can be **very** large and…
  - reconstruction of the tree can take **several hours**

or…

- up to **2% normal execution overhead** can be endured

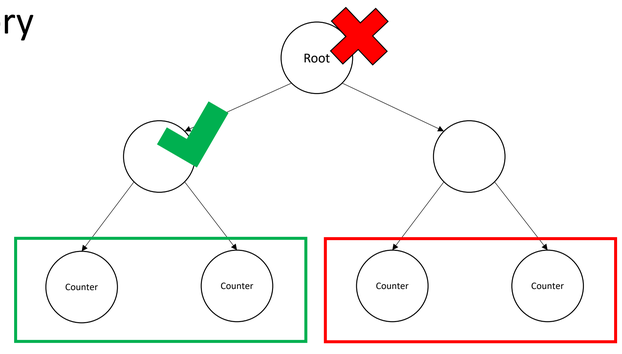
2. Proposed Solution

First, recovery the Bonsai Merkle Subtree from the Root Stored On-Chip

- **if verification succeeds,** then reboot the system with verified memory
- **if verification fails,** then the system is rendered unusable
  - finer knowledge of where an attack might occur
  - can potentially save uncorrupted memory

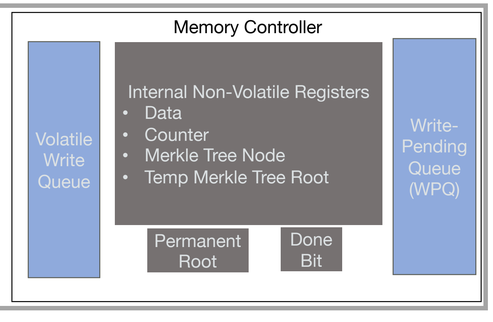
Lazy Verification of System Post-Boot

- takes advantage of **hardware parallelism** while the system is in use
- lower wait time for system verification on power-on after power is endured during normal system execution



0. Assumed Architecture

- memory controller design based on state-of-the-art work
- **Non-volatile registers:** durable storage of trusted values on-chip through crashes
- **Logging:** mechanism in place to ensure transaction of persistent data and metadata writes to memory



3. Hypothesis

- We should see at least **8x** speedup
  - the tree is 8-ary, so recovering an immediate child of the root would be 8x less work
- We will endure **negligible** overhead
  - the solution is contained within the recovery scheme
- The system will be **functional** and its **integrity** will be verified seemingly instantly

4. Future Work

- measuring **how much** data to recover versus **how fast** data recovery can take
- allocating **tree contiguous** memory by process to avoid **incidental complete reconstruction**
- software libraries for **dynamic allocation** of high priority integrity verification memory resources