ConcurORAM: High-Throughput Stateless Parallel Multi-Client ORAM

Anrin Chakraborti
Stony Brook University

Radu Sion
Stony Brook University
Its All About the Clouds!
Protecting Outsourced Data

• Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. Islam et al. NDSS, ‘12
• Connecting the Dots: Privacy Leakage via Write-Access Patterns to the Main Memory. John et al. HOST, ’17
• ...
Oblivious RAM (ORAM)

Observing the physical memory accesses, an adversary cannot learn
1. Which item has been accessed.
2. What operation has been performed.
Path ORAM [Stefanov et al. CCS ‘13]

<table>
<thead>
<tr>
<th>BlockID</th>
<th>LeafID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$v_2$</td>
</tr>
<tr>
<td>1</td>
<td>$v_0$</td>
</tr>
<tr>
<td>2</td>
<td>$v_3$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Position Map

STASH

CLIENT

SERVER

$V_0$

$V_1$

$V_2$

$V_3$
Path ORAM Evictions

Can also evict along pre-determined paths
Multi-Client Scenarios

Inter-Client Privacy

Correctness
Previous Approaches

OPRAM: Inter-Client Comm.
- Rarely suitable for practical scenarios
- Clients need to be aware, online

TaoStore: Trusted Proxy [S&P ’15]
- Violates ORAM trust model
- Proxy is a throughput bottleneck.
- Single point of failure/compromise
ConcurORAM Highlights

- No inter-client communication
- Eliminates proxy, scales better

Parallel Queries

Non-Blocking Evictions

Parallel Evictions
Parallel Queries

**Query Log:**
Transaction log with query addresses

**Result Log:**
Obliviously cache query results
Path ORAM Timeline

Eviction

Queries blocked

Time

Queries

Queries
Non-Blocking Evictions

**Problem:**
Evictions and queries access same data structures

**Insight:**
- Designated trees for queries and evictions
  - Sync periodically
- Multi-Phase Evictions:
  - **Processing** (Expensive): Update Write-only tree
  - **Commits**: Sync data tree
Non-Blocking Eviction Timeline

- **Eviction 1**: Queries stop for commits
- **Eviction 2**: Commit

- **Processing**: Eviction 1
- **Must Wait**
- **Queries**: Time
- **Queries**: Queries stop for commits
Parallel Evictions

**Problem:**
One evictions at a time is not!
• Maintain consistency

**Insight:**
Deterministic eviction path selection
• fix number of evictions in parallel
• Fine-grained locks

“If $k$ consecutive eviction execute in parallel, the eviction paths intersect at only up to $\log(k) + 1$ levels of the tree”
Parallel Eviction Timeline

Only one eviction executes critical section

Eviction 1

Eviction 2

Queries

Queries

Time
Other Challenges

- Ordering eviction commits
- Store and privately query results
- Maintain metadata consistency
- Fault tolerance
- ....
Throughput

Throughput scales better

Throughput (MB/s) (Higher is Better)

Ops/s

# of Clients

1 2 10 15 20 25 30

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM

ConcurORAM TaoStore PD-ORAM
Query Latency

Bounded query access time

Access Time (in ms) (Lower is Better)

Latency vs. # of Clients

- ConcuORAM
- TaoStore
- PD-ORAM
Summary

First Tree-Based Parallel ORAM
  • No inter-client communication
  • No Trusted Third-Parties

Parallelize Queries & Evictions

Scales Well
  • 2x Overall Throughput
What I am working on

Oblivious RAM [NDSS ’19, ‘19]  
Plausible Deniability [PETS ‘17, ‘19]

Integrity-Preserving Block Storage [ApSys ‘17]  
History Independence [TIFS ’15]

Secure CPU Architecture & Secure Virtualization

Query Authentication [TKDE]

I am on the job market!
Thank you!!