CaSA: End-to-end Quantitative Security Analysis of Randomly Mapped Caches

Thomas Bourgeat, Jules Drean, Yuheng Yang, Lillian Tsai, Joel Emer, Mengjia Yan

Published at MICRO’20
Problem: Incomplete Security Analysis

- Cache side channels are a serious security threat
- Promising mitigation: randomly mapped cache
- The security property is not well understood
Key Insights: Telecommunication Analogy

- Contributions:
  - CaSA leverages concepts from telecommunications to enable quantitative analysis
  - An end-to-end communication paradigm to enable comprehensive analysis
  - New findings that refute common beliefs

Cache Channel
Cache Side-channel Attacks

If (secret==1) { foo = ld 0x1000; }

Victim code

Attacker monitor victim’s behavior through cache conflicts.

Attacker code

While(1) { time(ld 0xf000); }
Cache Side-channel Attacks

If (secret==1) {
  foo = ld 0x1000;
}

While(1) {
  time(ld 0xf000);
}

Using black-box mapping function increases the cost to build an Eviction Set to:

$O(N)$ (with $N$ the number of lines in the cache)
Security Metrics

• Community intuition on Security Metrics:
  • “How hard it is to build an eviction set” is a good quantitative notion of security

• State-of-the-art secure cache design approaches:
  • Dynamic mapping
  • Non-deterministic mapping

• Our work:
  • This security metric can be misleading
  • Both design approaches fail to provide security
Dynamic Mappings

• Common belief: attacks can not happen across epochs
• Dynamic remapping incurs performance overhead
Non-Deterministic mapping

• Make conflict relationship between addresses non-deterministic

State of the art:
Non-deterministic + Dynamic mapping = Good security
Traditional Analysis

• Hard-conflict addresses:
  • Guarantee eviction
  • Difficult to obtain

• Soft-conflict addresses:
  • Easy to obtain
  • Need many of such addresses to reliably evict addresses

- Narrowly focus on eviction set construction and lose the bigger picture.
- Only want to create a one-to-one map from micro-architecture events to secret
End-to-end Communication Paradigm

- Leverage the concepts from telecommunication

- Trade-off between calibration and signaling
  - Long time on calibration $\rightarrow$ shorter time needed for signaling
  - Short time on calibration $\rightarrow$ longer time needed for signaling

---

Calibration (i.e., Eviction set construction) \[ \text{Eviction Addresses} \] \[ \text{Signaling (e.g., Prime+Probe)} \] \[ \text{Decode} \] \[ \text{Signal Samples} \]
New Security Metric

"How difficult to construct an eviction set"

End-to-end communication cost in Calibration + Signaling
Statistical Representation of Signals

• Signal: a random variable “X”
  • Describes the number of misses observed by the attacker
  • Follow a probability distribution

• Example:

<table>
<thead>
<tr>
<th></th>
<th>Prob observing 0 miss</th>
<th>Prob observing 1 miss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim accesses</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>No victim accesses</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

• Successfully covert the problem to a statistical analysis problem
  • How many samples are needed to distinguish the two distributions?
Two Insightful Findings

• Cross-epoch communication is possible

• Spending maximum resources on calibration is not the best strategy
Cross-epoch Communication

• In each epoch:

![Diagram showing the timeline of epochs with rounds of calibration and signal transfer]

- One round of calibration
- One round of signal transfer
- Beginning/end of an epoch
Cross-epoch Communication

Signals across epochs when attacking the RSA square-and-multiple function.

Cache configuration: 16 hash-groups / 1 way per hash-group/ 16k cache lines
Cross-epoch Communication

Cache configuration: 16 hash-groups / 1 way per hash-group/ 16k cache lines
Epoch size = 100*16K accesses
Trade-off between Calibration and Signaling

Spending the maximum amount of resources in calibration is not always the best strategy.

Cache configuration: 16 hash-groups / 1 way per hash-group/ 16k cache lines
Epoch size = 100*16K accesses
Conclusion & Long-term Impact

• Comprehensive security analysis for micro-architecture side channels should focus on end-to-end communication.

• CaSA formalize the analysis of micro-architecture behavior to the analysis of random variables.

- Cache Occupancy Attack
- Memory Controller
- Front End
- Execution Engine
- ......