New Models of Cache Architectures
Characterizing Information Leakage from Cache Side Channels

Tianwei Zhang and Ruby B. Lee
Princeton University
Motivation

Cache side-channel attacks are dangerous

Defenses:
-- Software
-- Hardware

Two questions:
-- Secure or not?
-- Relative vulnerabilities?
Outline

Cache side-channel attacks and defenses

Side-channel Evaluation:
  -- Modeling side-channel leakage using non-interference
  -- Modeling different caches using FSM
  -- Leakage measurement

Cache model validation

Conclusion
Cache Side-channel Attacks

Mechanism:
-- Different cache events (hit or miss)

Root Cause:
-- Interference (external or internal)

Classification:

<table>
<thead>
<tr>
<th></th>
<th>External Interference</th>
<th>Internal Interference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Misses</td>
<td>I. Access-based attack [Percival’s attack]</td>
<td>II. Timing-based attack [Bernstein’s attack]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Hits</td>
<td>III. Access-based attack [Shared library attack]</td>
<td>IV. Timing-based attack [Bonneau’s attack]</td>
</tr>
</tbody>
</table>
Example: Type I Attack

Cache Misses due to External Interference

-- Victim: evict the cache lines holding the attacker’s data
-- Attacker: experiences the cache misses and infers the victim’s memory access
Partitioning-based Cache Defense

Isolating the cache

Static-Partitioning (SP) cache

Partition-Lock (PL) cache

(Partition the cache by ways or sets)

(Locked and cannot be replaced)
Randomization-based Cache Defense

Random noise or random mapping

Random-Eviction (RE) cache

Random-Permutation (RP) cache

NewCache

(Random selected and evicted)

(Dynamic and random mapping between memory address and cache sets)

(Combination of direct-mapping and fully-associative mapping)
Outline

Cache side-channel attacks and defenses

Side-channel Evaluation:
  -- Modeling side-channel leakage using non-interference
  -- Modeling different caches using FSM
  -- Leakage measurement

Cache model validation

Conclusion
Non-Interference in Cache Side channel

Definition:

-- Given a state machine $M$, input $I$ does not affect the output $O$, then $I$ does not interfere with $O$

-- $I$ is non-interfering with $O$: there is no information flow from $I$ to $O$

Quantification:

$$I(I, O) = \sum_{i \in I} \sum_{o \in O} P_{I,O}(i, o) \log\left(\frac{P_{I,O}(i, o)}{P_I(i)P_O(o)}\right)$$

$$= \sum_{i \in I} \sum_{o \in O} P_O(o)P_{I|O}(i|o) \log\left(\frac{P_{I|O}(i|o)}{P_I(i)}\right)$$
Three Conditions for Non-interference

\[ I(I, O) = \sum_{i \in I} \sum_{o \in O} P_o(o)P_{I|O}(i|o) \log \left( \frac{P_{I|O}(i|o)}{P_I(i)} \right) \]

Output Elimination
-- the channel produces no effective output.

Noise Domination:
-- Output is generated due to the inherent noise

Input Ambiguity:
-- for any output, the input is indistinguishable.
Outline

Cache side-channel attacks and defenses

Side-channel Evaluation:

-- Modeling side-channel leakage using non-interference
-- Modeling different caches using FSM
-- Leakage measurement

Cache model validation

Conclusion
Cache State Machine

Three States
-- **V** (occupied by the victim)
-- **A** (occupied by the attacker)
-- **INV** (contain no valid contents)

Five Transition Events
-- **V**\_miss (victim has a cache miss)
-- **V**\_hit (victim has a cache hit)
-- **A**\_miss (attacker has a cache miss)
-- **A**\_hit (attacker has a cache hit)
-- **Invalidate** (clears out the data)
Cache Modeling to Generate Interference Matrix

Event Vector:

\[
T_0 \quad \cdots \quad T_n \quad \text{|--|} \quad \text{Event Vector.}
\]

Information Flow Log:

- \( S'_{0,0} \quad \cdots \quad S'_{0,m-1} \quad \text{|--|} \quad l'_{0,0} \quad \cdots \quad l'_{0,m-1} \quad \text{|--|} \quad \text{Replacement Matrix.}
\]

- \( S'_{n-1,0} \quad \cdots \quad S'_{n-1,m-1} \quad \text{|--|} \quad l'_{n-1,0} \quad \cdots \quad l'_{n-1,m-1} \quad \text{|--|} \quad \text{Replacement Matrix.}
\]

Current State:

\[
S_{p,q} = \{A, V, INV\} : \text{state matrix}
\]

Replacement Matrix:

\[
l_{p,q} = \{0, 1, \ldots, m-1\} : \text{replacement matrix}
\]

Next State:

\[
S'_{p,q} = \{A, V, INV\} : \text{state matrix}
\]

Informations:

- \( I_{p} \rightarrow O_{q} \) : interference matrix
- \( I_{-1} \rightarrow O_{q} \) : inherent noise
Outline

Cache side-channel attacks and defenses

Side-channel Evaluation:
  -- Modeling side-channel leakage using non-interference
  -- Modeling different caches using FSM
  -- Leakage measurement

Cache model validation

Conclusion
Case Study: Type I Attack

Experiment Setup
-- 3-set, 2-way set-associative cache
-- Use Murphi to go over 10 rounds

Conventional Cache

<table>
<thead>
<tr>
<th>$P_{l,o}(l,o)$</th>
<th>$I_0$</th>
<th>$I_1$</th>
<th>$I_2$</th>
<th>$I_{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_0$</td>
<td>33.3%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>$O_1$</td>
<td>0.0%</td>
<td>33.3%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>$O_2$</td>
<td>0.0%</td>
<td>0.0%</td>
<td>33.3%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
Other Caches Evaluation

Partitioning-based Cache (SP cache):

-- Output Elimination

Randomization-based Cache (RP cache):

-- Large noise domination

-- Input Ambiguity

<table>
<thead>
<tr>
<th>( P_{I,O}(I,O) )</th>
<th>( I_0 )</th>
<th>( I_1 )</th>
<th>( I_2 )</th>
<th>( I_{-1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O_0 )</td>
<td>2.17%</td>
<td>2.19%</td>
<td>2.19%</td>
<td>26.8%</td>
</tr>
<tr>
<td>( O_1 )</td>
<td>2.19%</td>
<td>2.17%</td>
<td>2.19%</td>
<td>26.8%</td>
</tr>
<tr>
<td>( O_2 )</td>
<td>2.19%</td>
<td>2.19%</td>
<td>2.17%</td>
<td>26.8%</td>
</tr>
</tbody>
</table>
Mutual Information Value

\[ I(I, O) = \sum_{0 \leq p' < n} \sum_{0 \leq q' < n} P_{I,O}(I_{p'}, O_{q'}) \log \left( \frac{P_{I,O}(I_{p'}, O_{q'})}{P(I_{p'})P(O_{q'})} \right) \]

<table>
<thead>
<tr>
<th>Cache Architecture</th>
<th>I(I,O) (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Cache</td>
<td>1.585</td>
</tr>
<tr>
<td>SP Cache</td>
<td>0.000</td>
</tr>
<tr>
<td>PL Cache (w/ preload)</td>
<td>0.000</td>
</tr>
<tr>
<td>RE Cache</td>
<td>0.461</td>
</tr>
<tr>
<td>RP Cache</td>
<td>2.586X10^{-6}</td>
</tr>
<tr>
<td>NewCache</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Interference metrics and mutual information show same results

-- Conventional > RE > RP > {PL, SP, New}
Outline

Cache side-channel attacks and defenses

Side-channel Evaluation:

-- Modeling side-channel leakage using non-interference
-- Modeling different caches using FSM
-- Leakage measurement

Cache model validation

Conclusion
An actual attack program

Probability Distribution of Candidate Keys
  -- Correct key has the highest probability in a successful attack

Experiment Setup
  -- Type I attack on AES cipher
  -- gem5 simulator
  -- L1 cache: size = 32KB, linesize = 32B, associativity = 8
Probability Distribution of Candidate Keys

Conventional Cache
Insecure

SP Cache
Secure

PL Cache
Secure

RE Cache
Insecure

RP Cache
Secure

NewCache
Secure
Conclusion

Apply the non-interference principle to the modeling of side-channel information leakage

Exploit mutual information to quantify the side-channel leakage, and identify three non-interference conditions

Build finite-state machines for modeling secure caches to measure the side-channel vulnerability

Verify cache security models with actual attacks
Thank you!