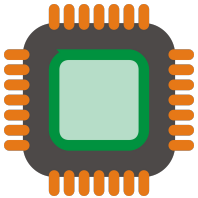
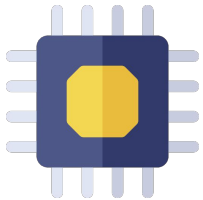
SoK: So, You Think You Know All About Secure Randomized Caches?

USENIX Security 2025 Track 4: Thursday, August 14, 2025

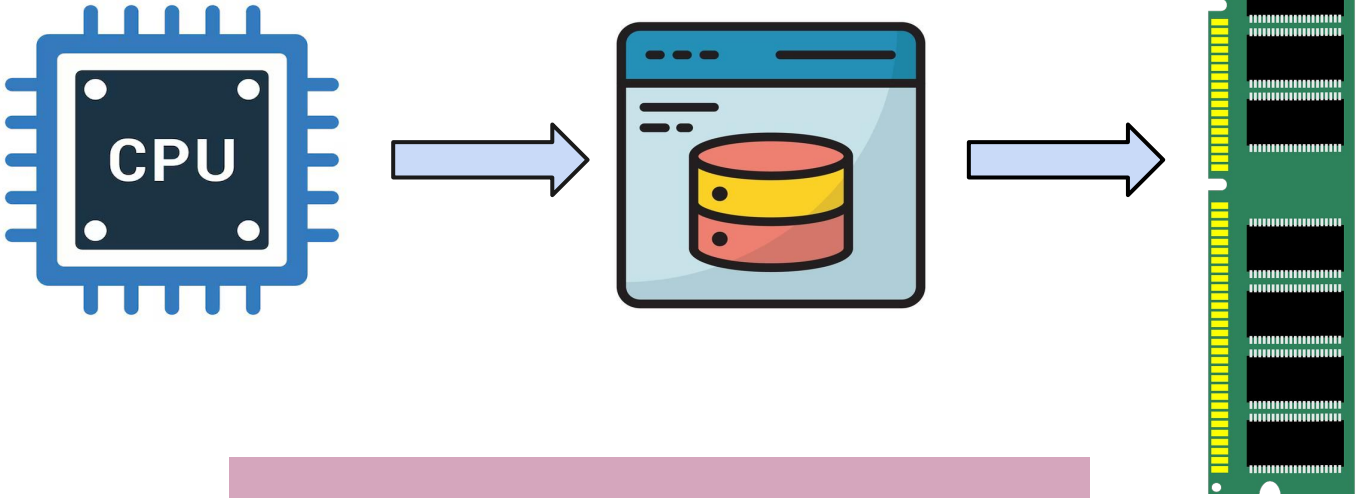
Anubhav Bhatla Hari Rohit Bhavsar
Sayandeep Saha Biswabandan Panda
Indian Institute of Technology (IIT), Bombay





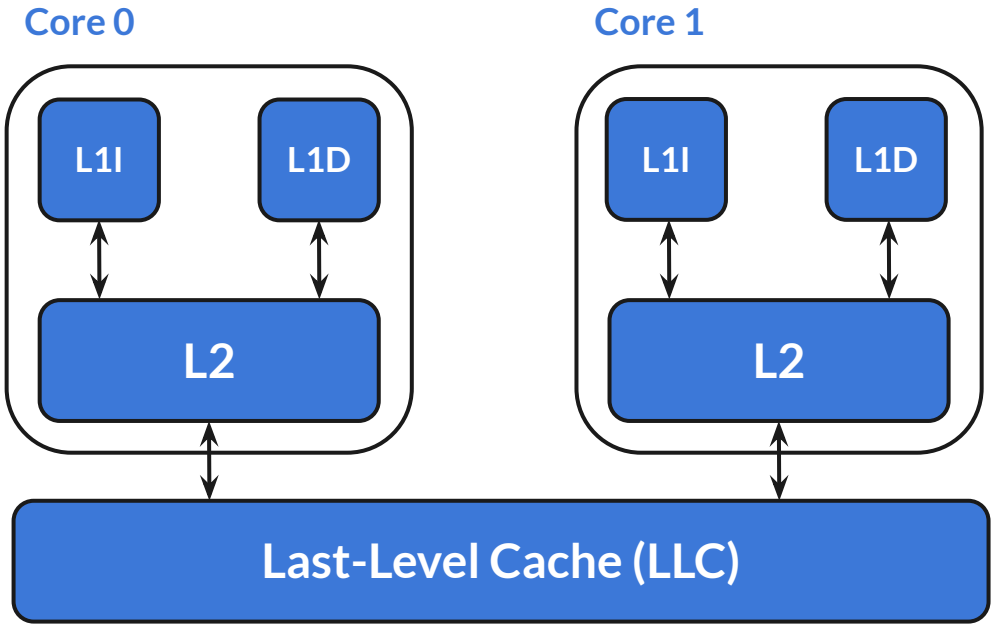
usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Background



Size: Registers < Cache < RAM
Speed: Registers > Cache > RAM

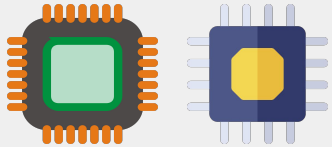
Background



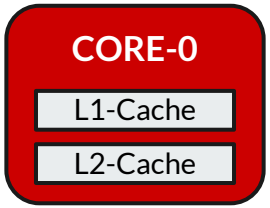
Cache hierarchy in modern processors

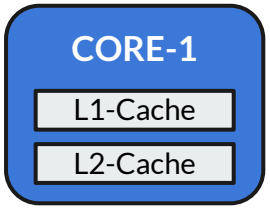
LLC is shared among all processes

Occupancy-based Attacks

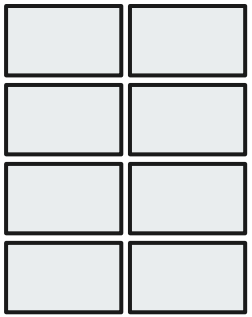


SPY
PROCESS





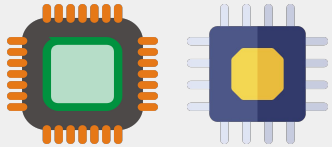
VICTIM
PROCESS

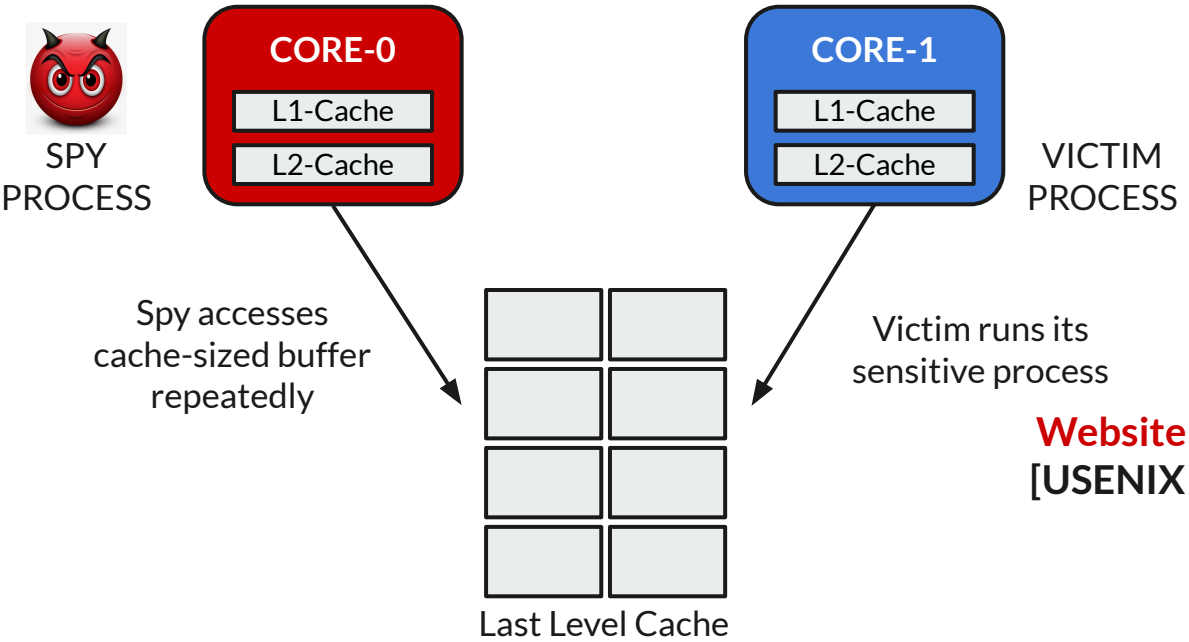


Last Level Cache

Website Fingerprinting
[USENIX SECURITY '19]

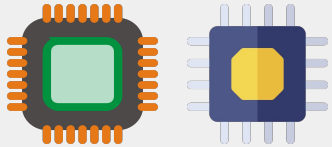
Occupancy-based Attacks

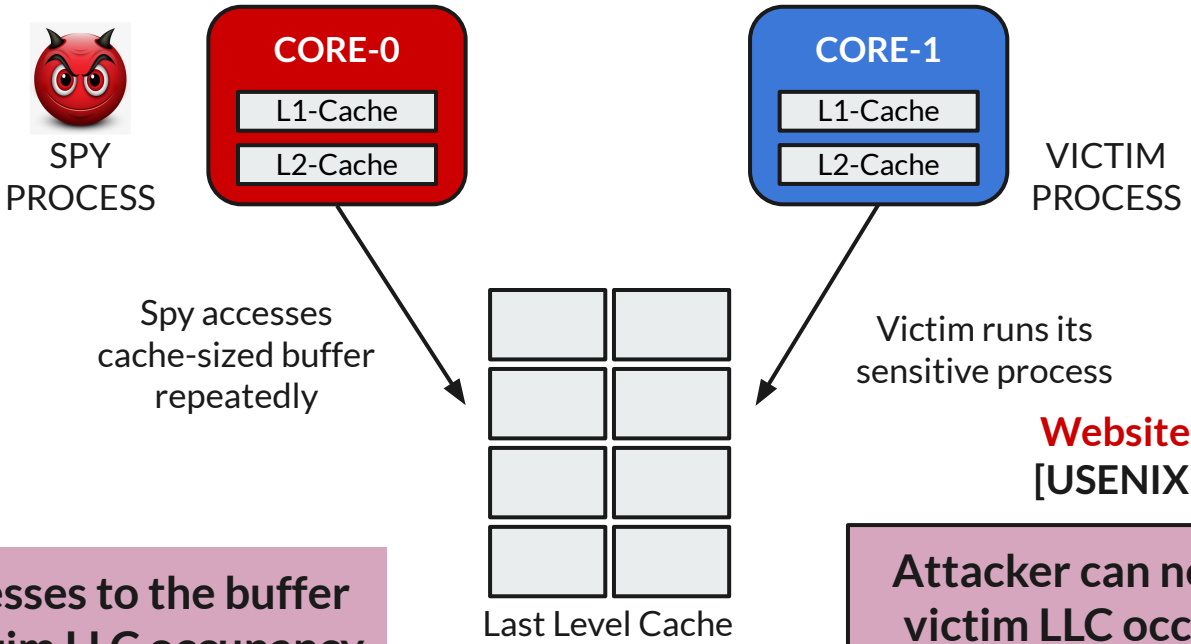




Website Fingerprinting
[USENIX SECURITY '19]

Occupancy-based Attacks



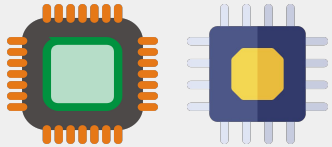


Website Fingerprinting
[USENIX SECURITY '19]

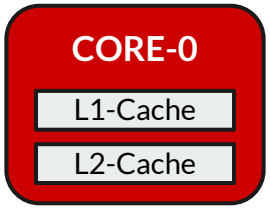
More accesses to the buffer
⇒ Less victim LLC occupancy

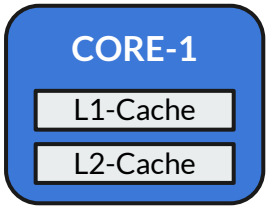
Attacker can now decipher victim LLC occupancy and leak sensitive information

Conflict-based Attacks

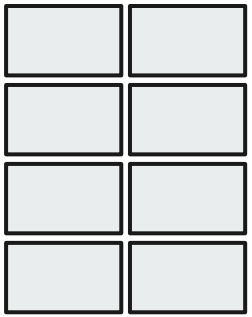


SPY
PROCESS





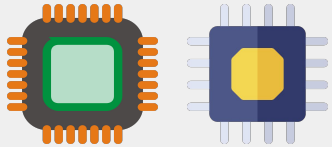
VICTIM
PROCESS

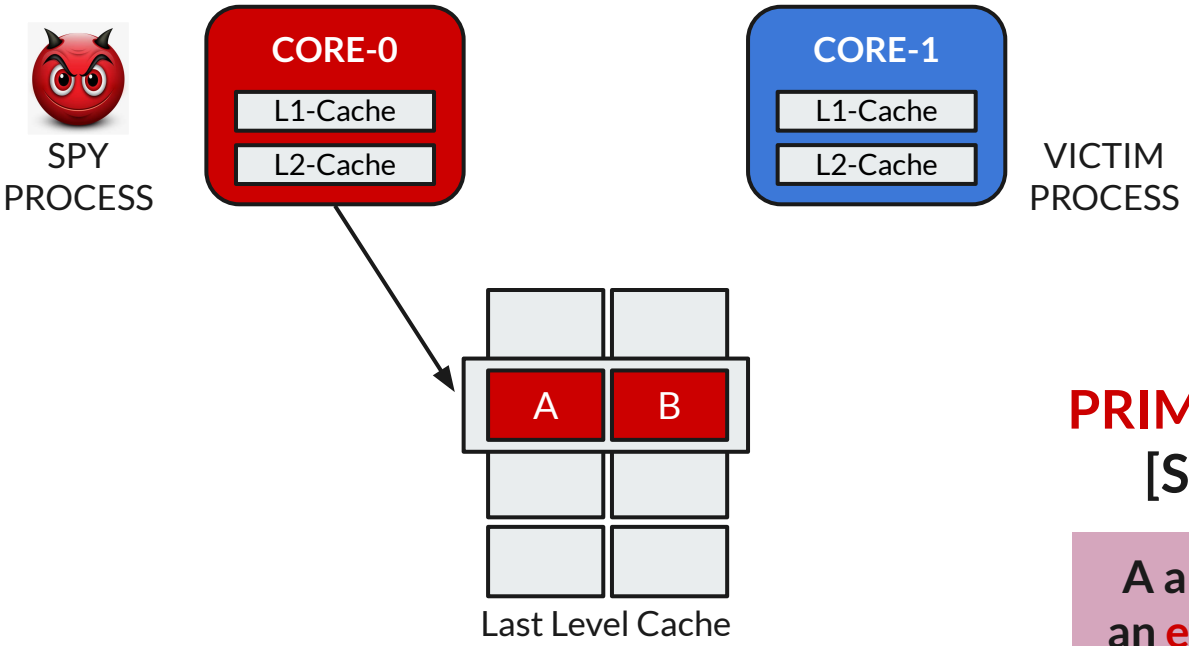


Last Level Cache

PRIME+PROBE
[S&P ‘15]

Conflict-based Attacks

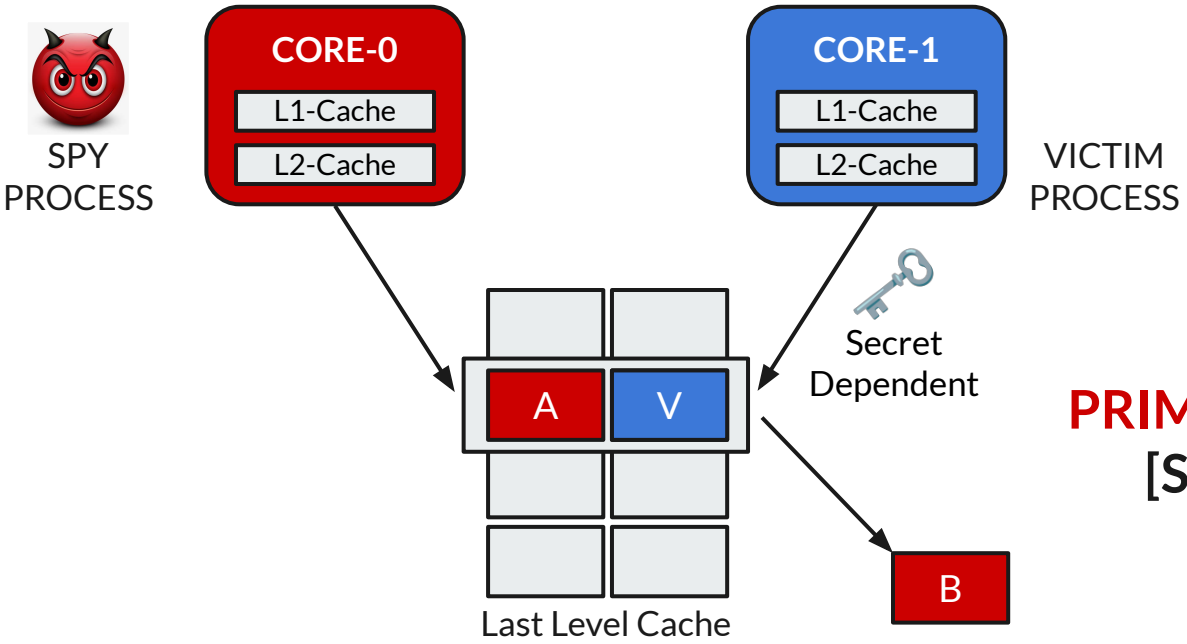




PRIME+PROBE
[S&P '15]

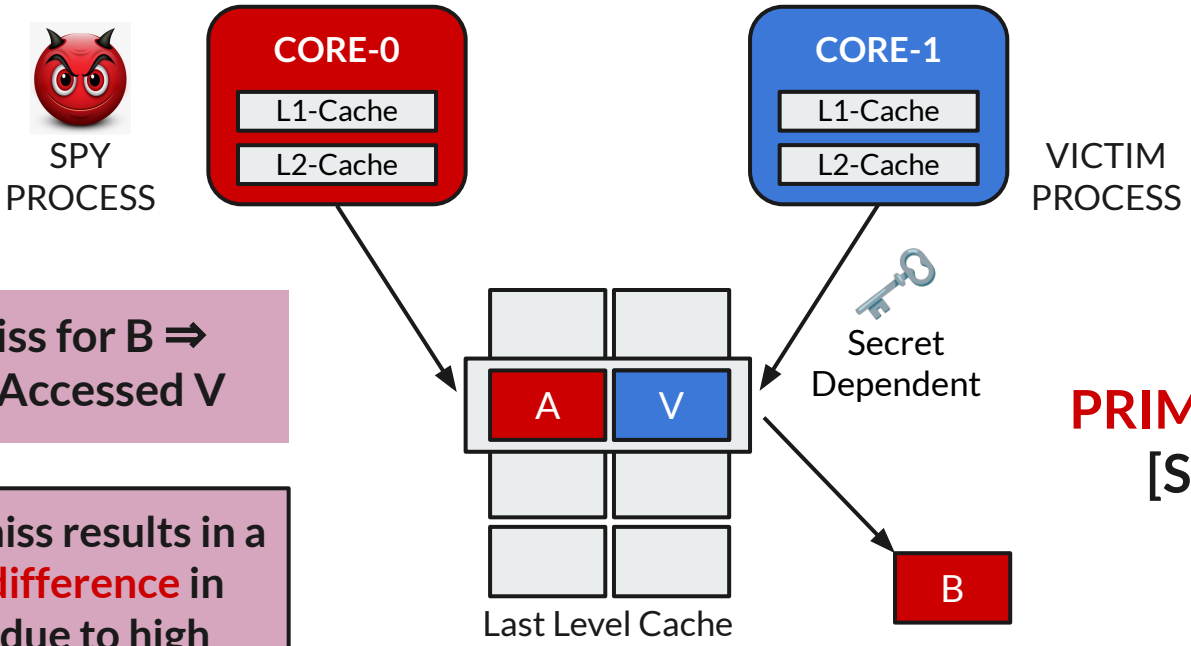
A and B form an eviction set

Conflict-based Attacks



PRIME+PROBE
[S&P '15]

Conflict-based Attacks

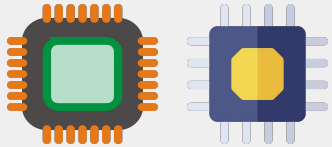


LLC Miss for B ⇒ Victim Accessed V

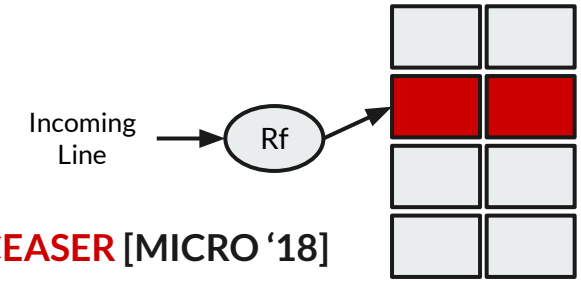
A cache miss results in a timing difference in access due to high DRAM latency

PRIME+PROBE [S&P '15]

Popular Secure Randomized Designs

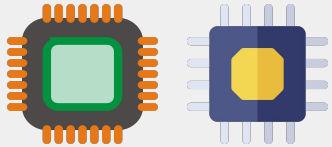


Randomization + Remapping

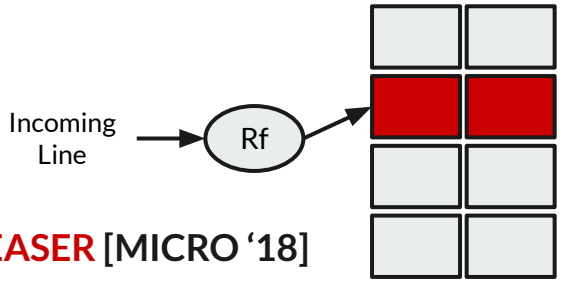


CEASER [MICRO '18]

Popular Secure Randomized Designs

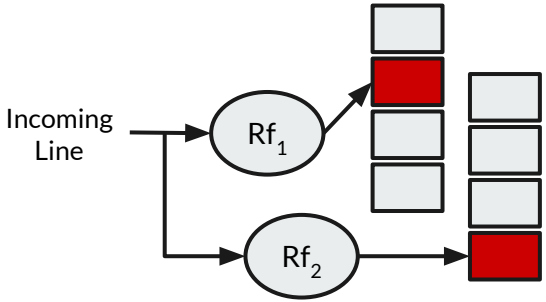


Randomization + Remapping



CEASER [MICRO '18]

Skews + Randomization + Remapping

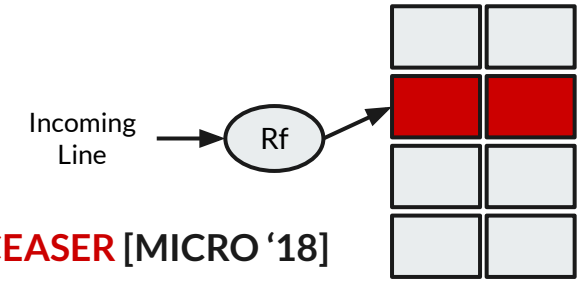


CEASER-S [ISCA '19]

SCATTERCACHE [S&P '19]

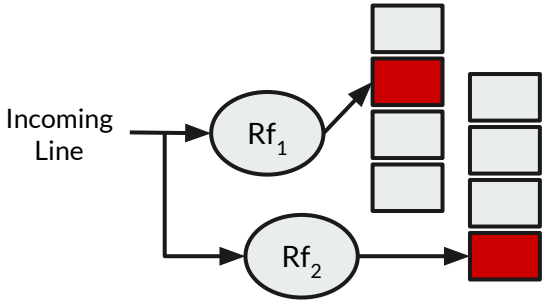
Popular Secure Randomized Designs

Randomization + Remapping

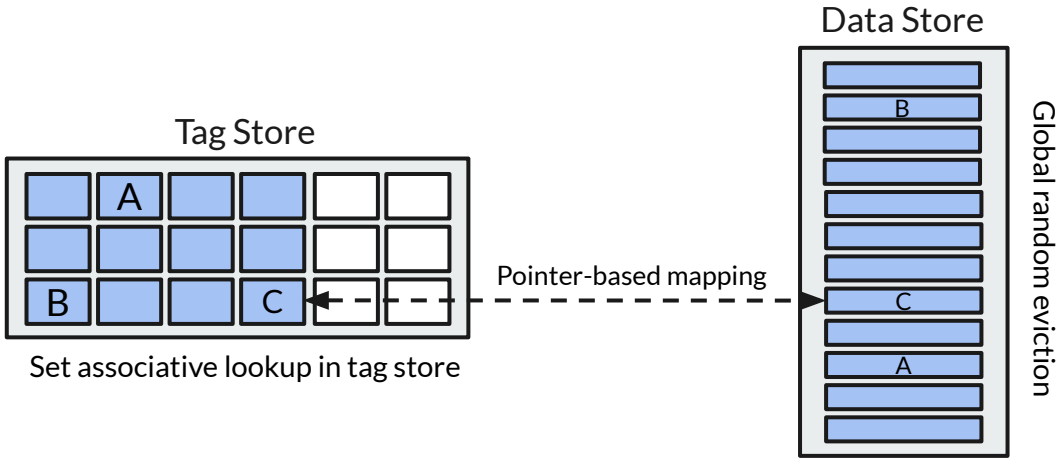


CEASER [MICRO '18]

Skews + Randomization + Remapping



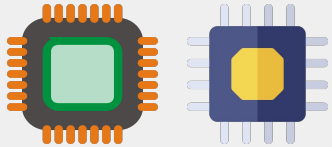
CEASER-S [ISCA '19]
SCATTERCACHE [S&P '19]



Skews (load-aware) + Randomization + Extra Invalid Tags (decoupled) + Global Random Eviction

MIRAGE [USENIX SECURITY '20]
MAYA [ISCA '24]

Overview

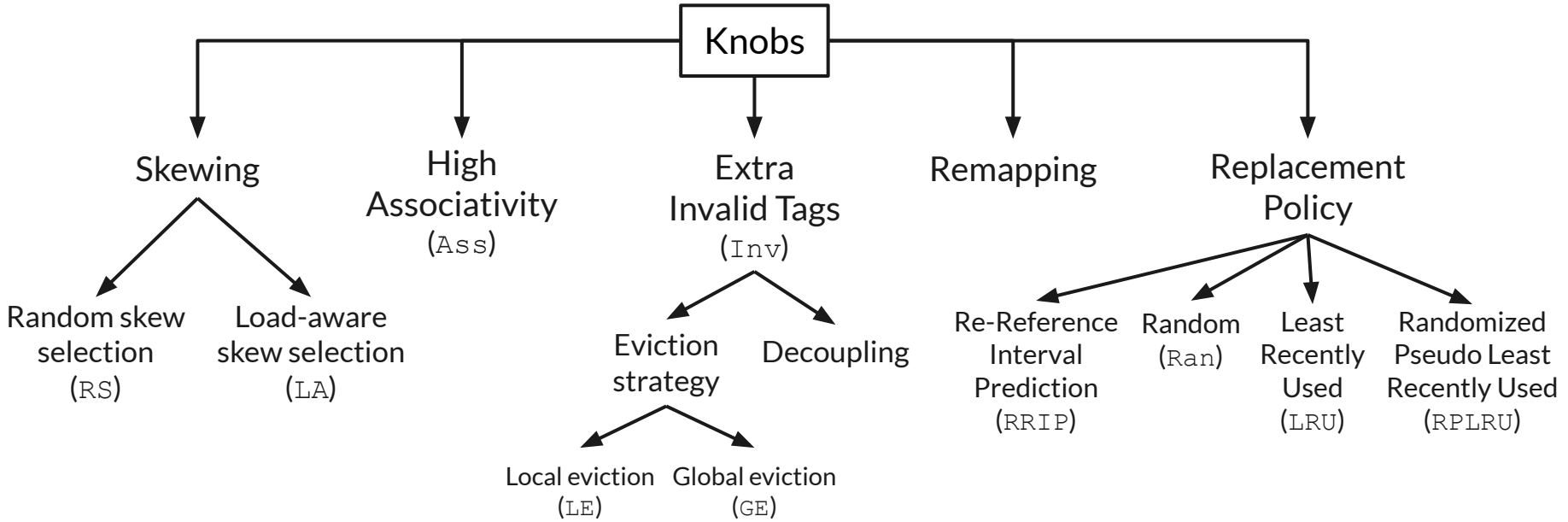


We systematize the design space for secure randomized caches by identifying key security knobs

We perform security analysis of each knob against conflict-based attacks. We also study which combinations of these knobs work

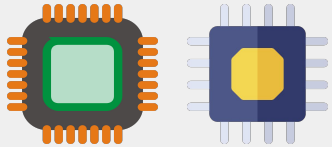
We analyze these knobs against occupancy-based attacks and compare them with partitioning-based designs

Security Knobs



- **Identified various knobs and sub-knobs used in modern secure randomized caches**
- **Randomization using block cipher is assumed by default**

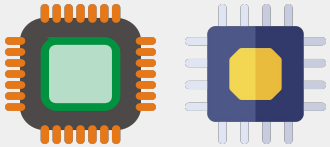
Metrics Used



METRIC I:
Eviction Rate

METRIC II:
Ease of Eviction Set Creation

Metrics Used



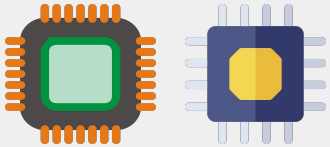
METRIC I:
Eviction Rate

Measures the probability (or rate) of evicting the target address using a perfect eviction set

ScatterCache [USENIX Security '19]
Song et al. [S&P '21]
ClepsydraCache [USENIX Security '23]

METRIC II:
Ease of Eviction Set Creation

Metrics Used



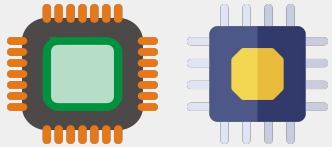
METRIC I:
Eviction Rate

METRIC II:
Ease of Eviction Set Creation

Measures the probability (or rate) of evicting the target address using a perfect eviction set

```
Loop start
1. Select random target x
2. Generate eviction set E for x
3. Access x
4. Access E
5. Check whether x is evicted
end
```

Metrics Used



METRIC I:
Eviction Rate

Measures the probability (or rate) of evicting the target address using a perfect eviction set

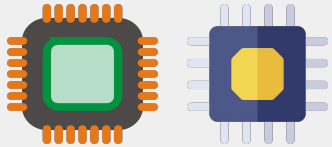
```
Loop start
1. Select random target x
2. Generate eviction set E for x
3. Access x
4. Access E
5. Check whether x is evicted
end
```

METRIC II:
Ease of Eviction Set Creation

Quantify in terms of number of LLC evictions needed to create a fixed-size eviction set

We test popular algorithms such as Conflict Testing and Prime, Prune and Probe (PPP)

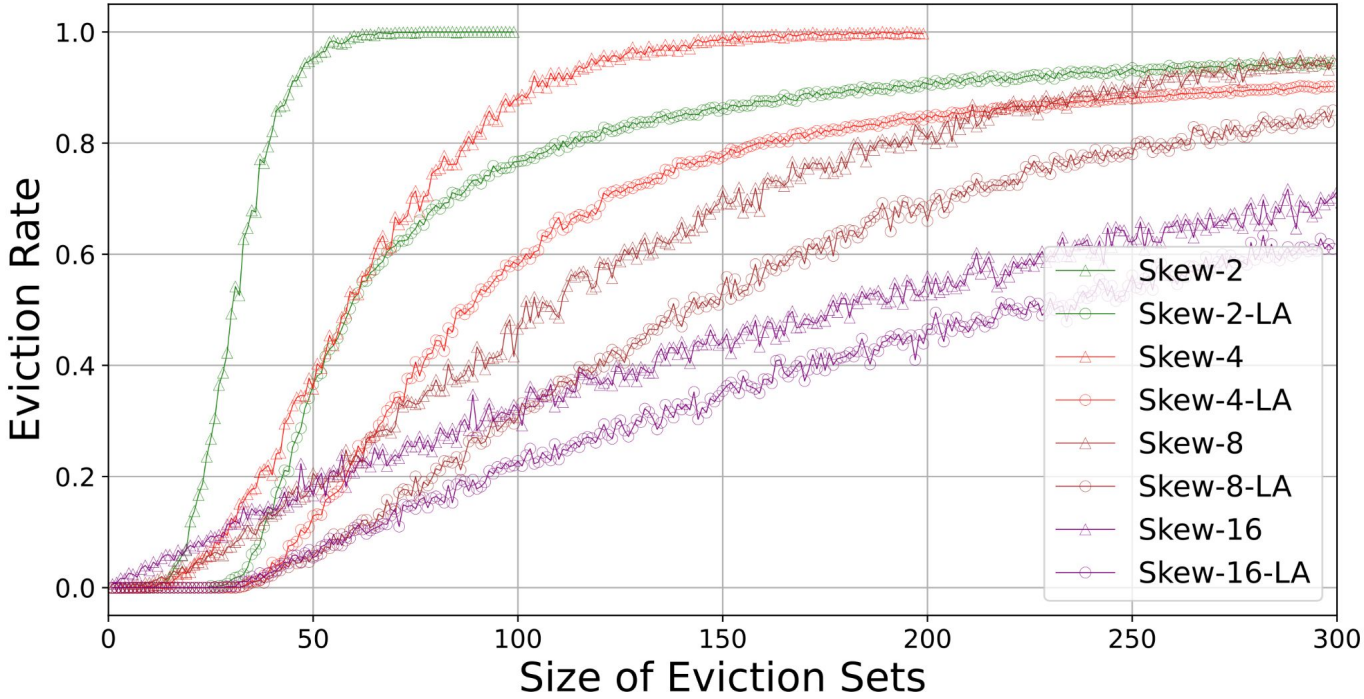
Knob 1: Skewing



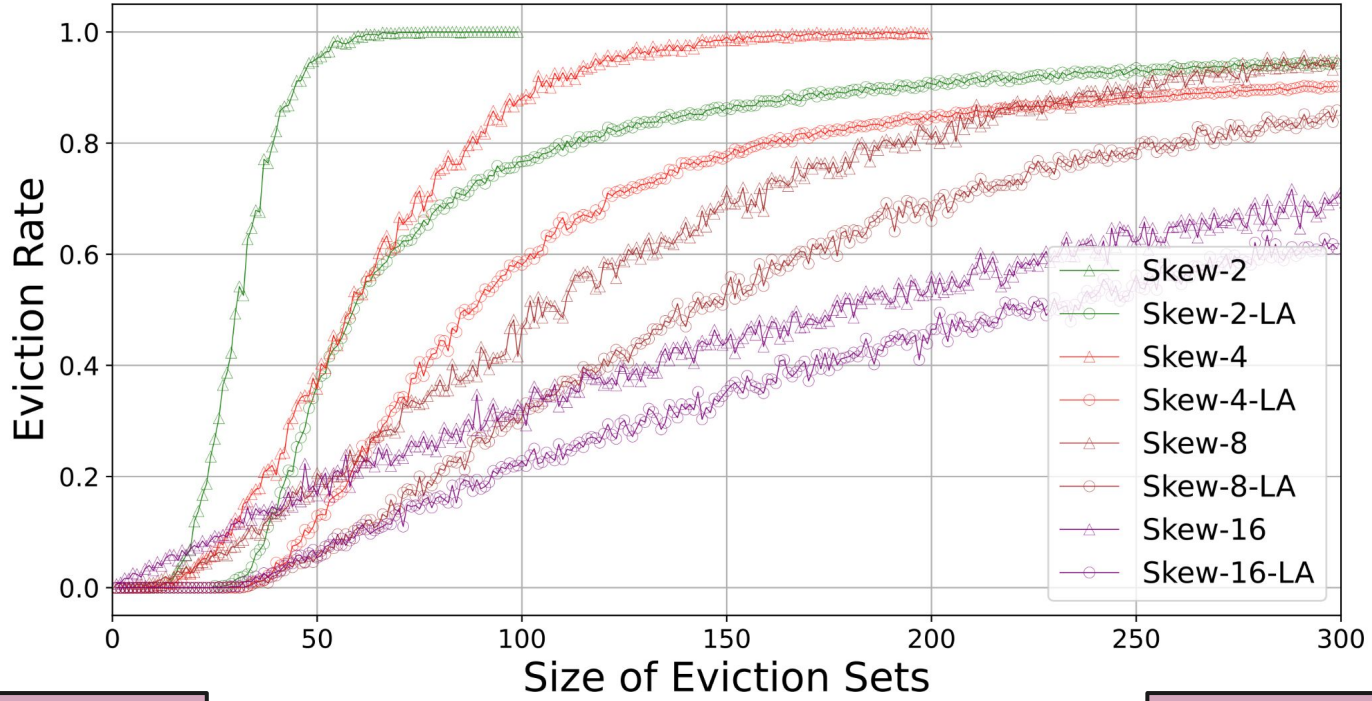
Random
Skew Selection

Load-aware
Skew Selection

Knob 1: Skewing



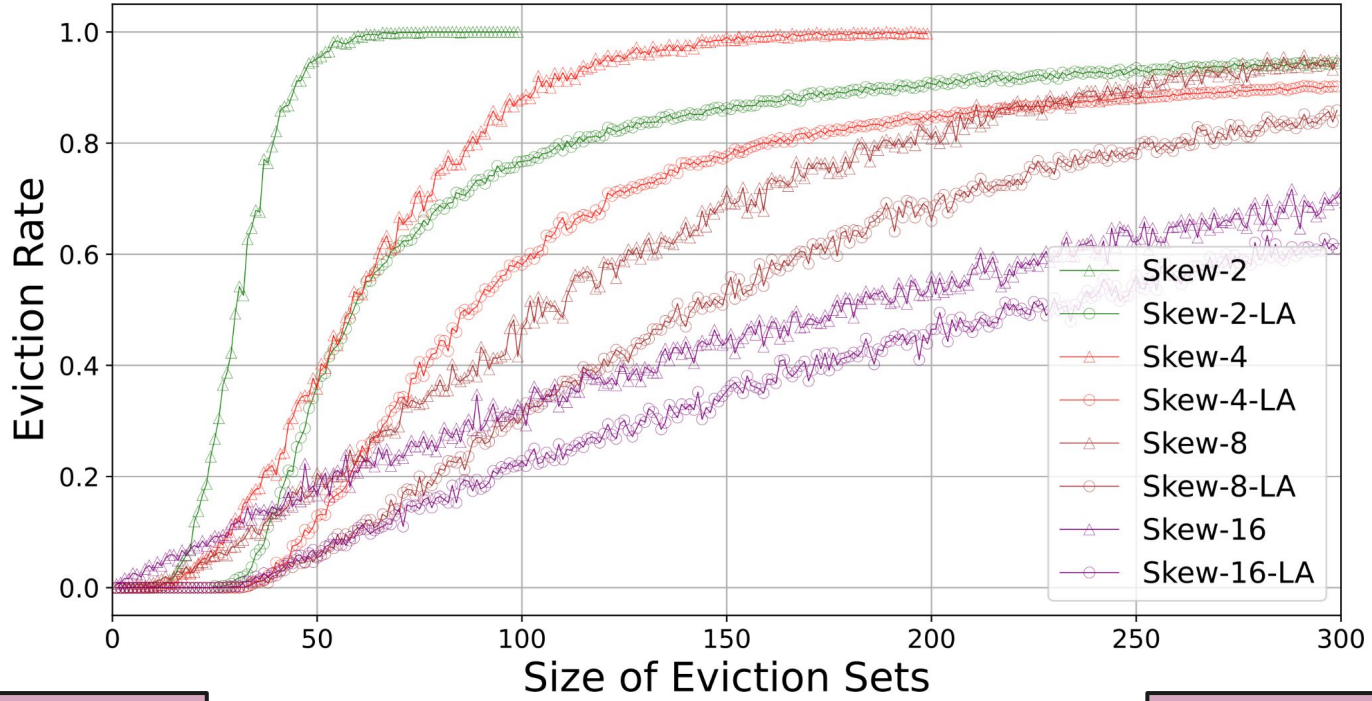
Knob 1: Skewing



Skewing helps improve security

LA does better than Random

Knob 1: Skewing

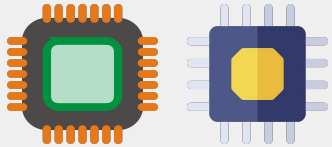


Skewing helps improve security

There are hidden nuances to consider!

LA does better than Random

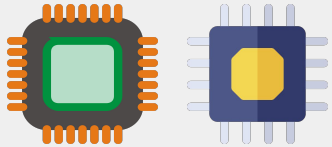
Knob 2: Extra Invalid Tags



Eviction Strategy

Decoupling

Knob 2: Extra Invalid Tags

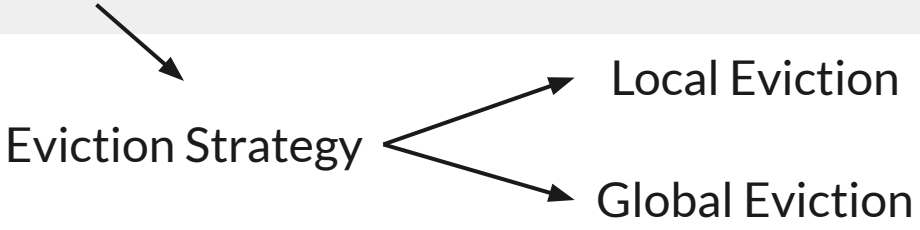


Eviction Strategy

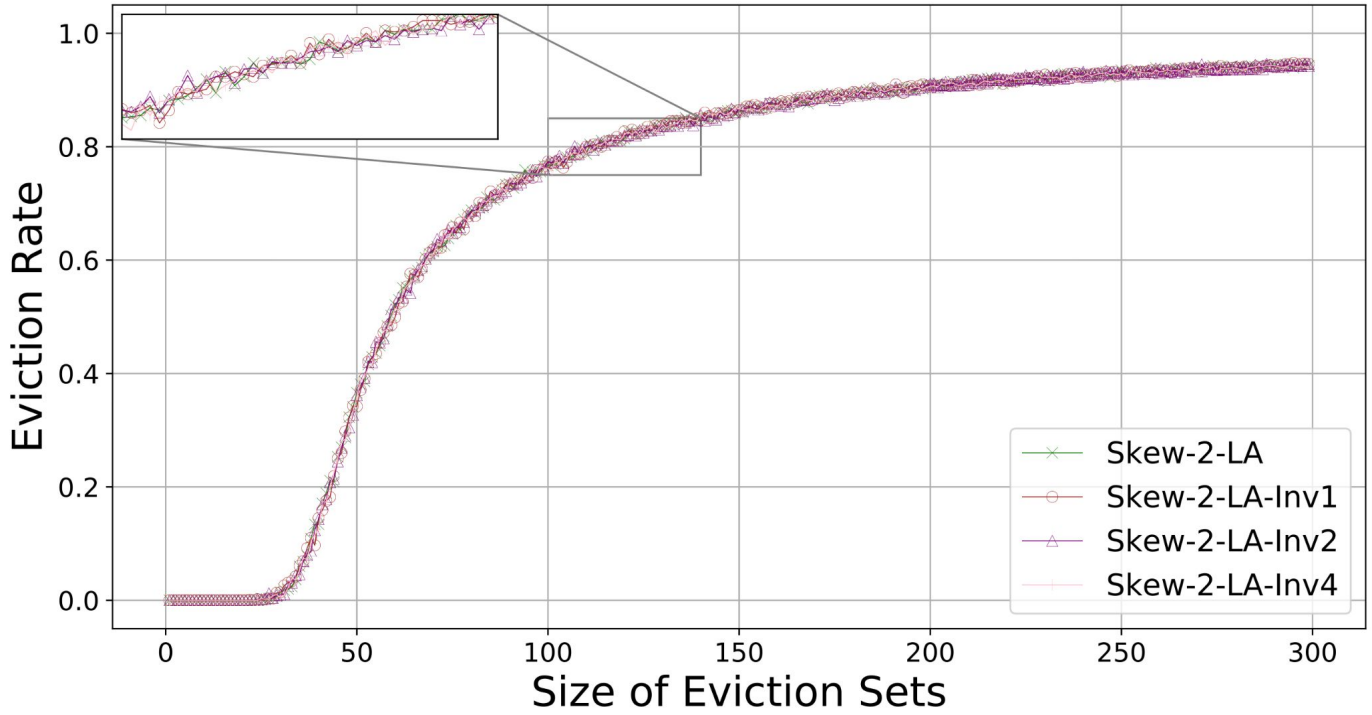
Decoupling

Decoupling has no security impact

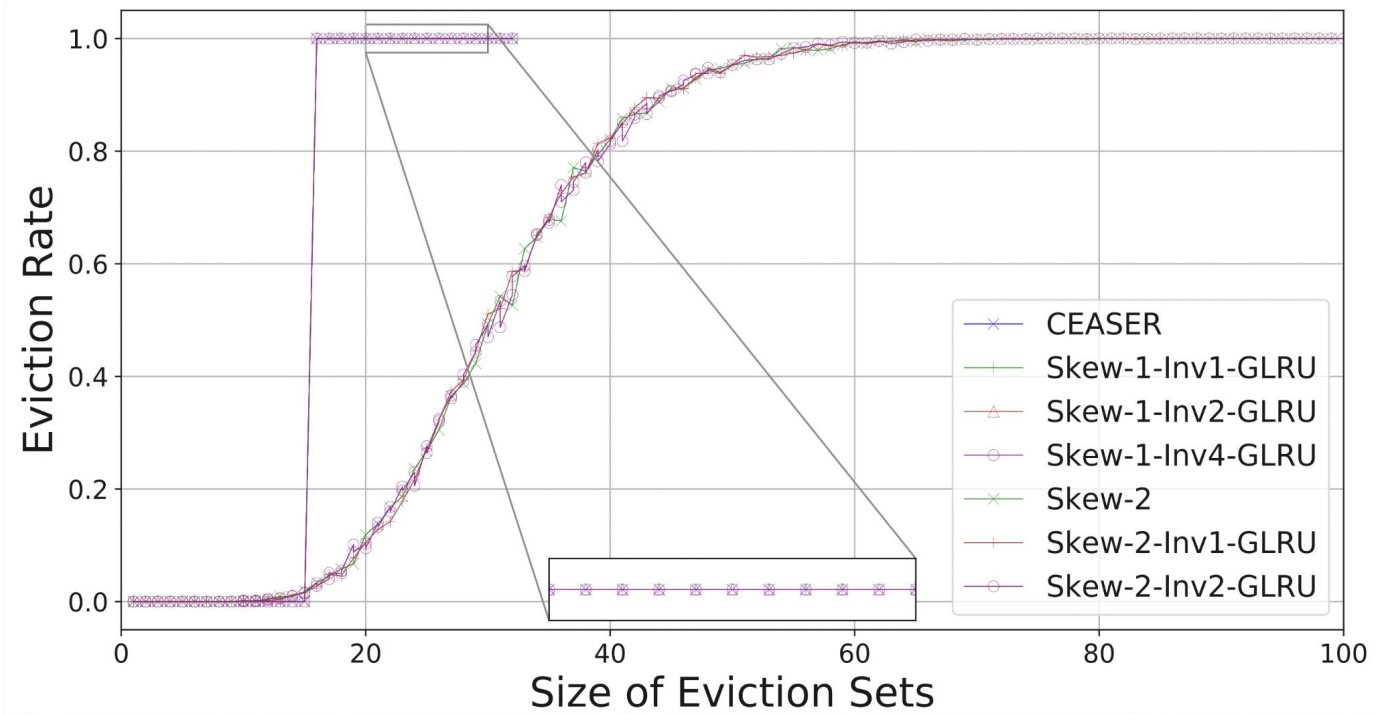
Knob 2: Extra Invalid Tags



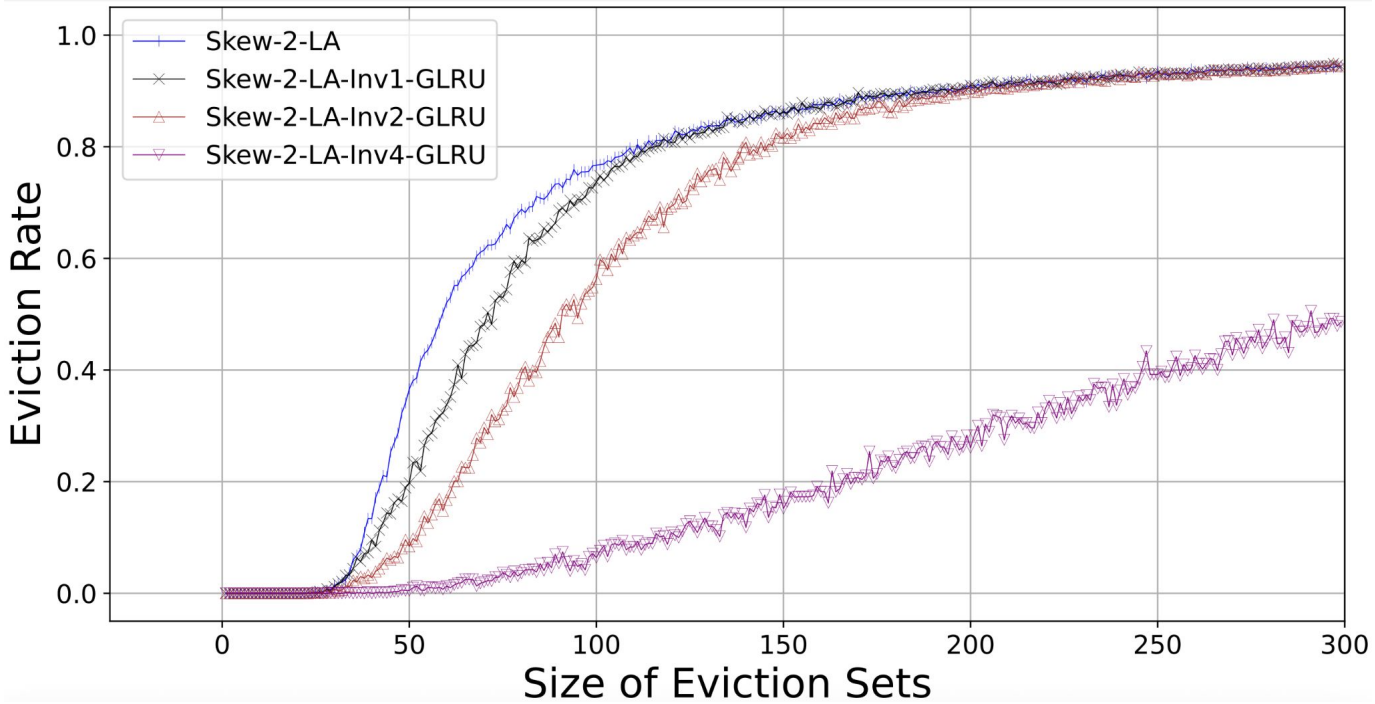
Knob 2: Extra Invalid Tags



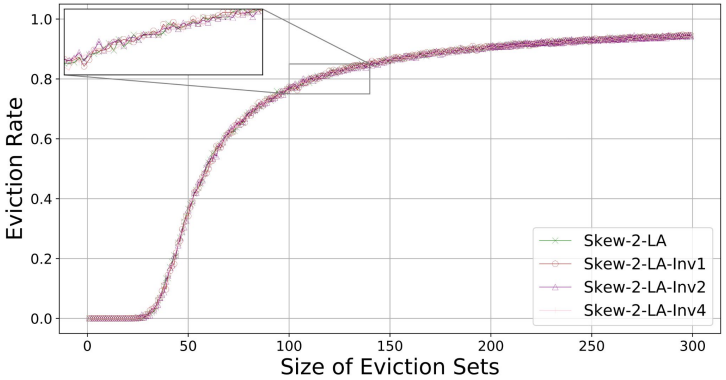
Knob 2: Extra Invalid Tags



Knob 2: Extra Invalid Tags

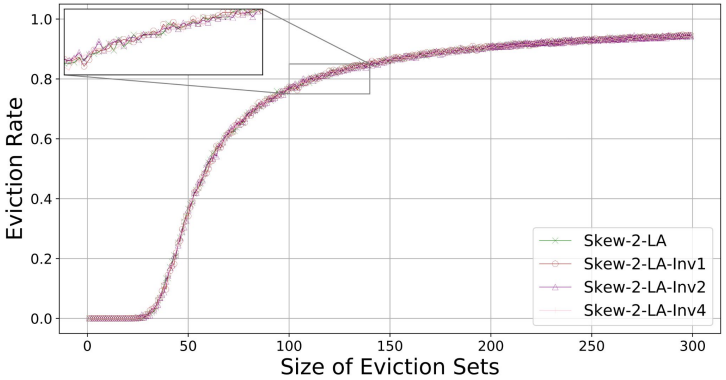


Knob 2: Extra Invalid Tags

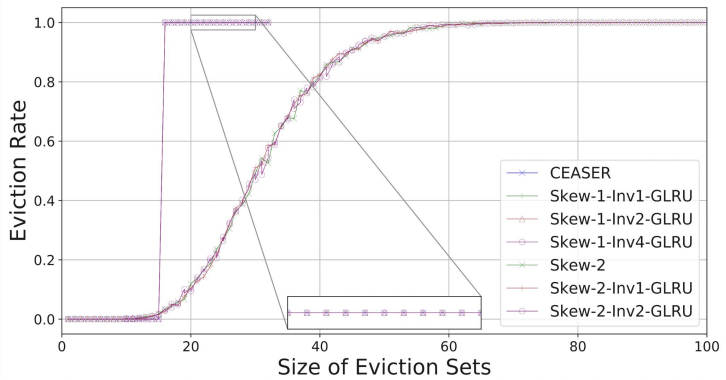


Skew + LA + Inv ❌

Knob 2: Extra Invalid Tags

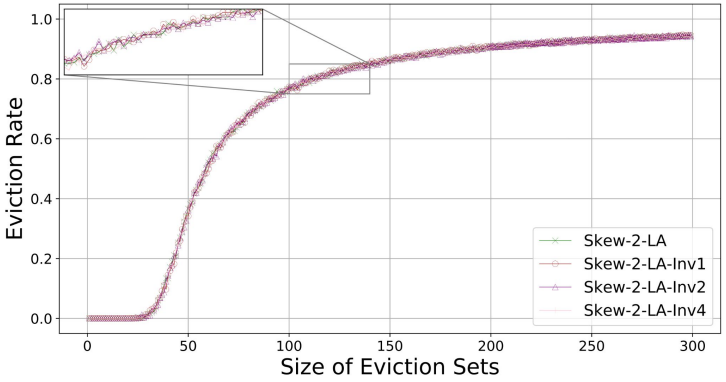


Skew + LA + Inv ❌

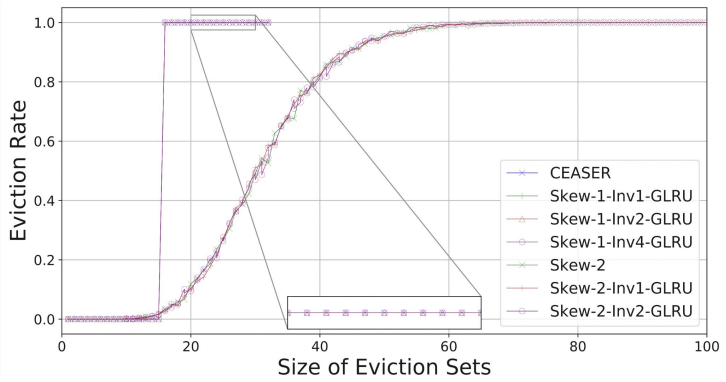


Skew + Inv + GE ❌

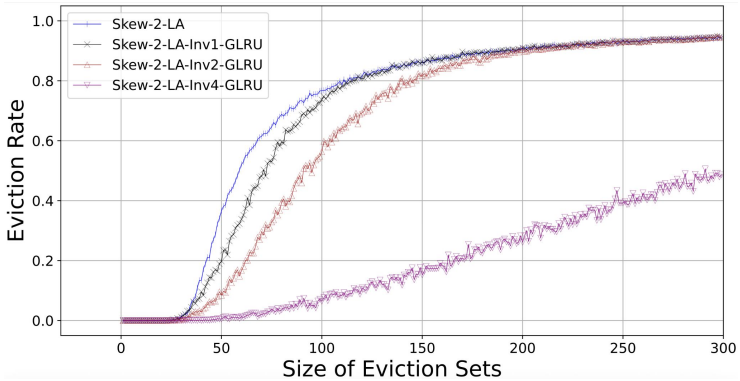
Knob 2: Extra Invalid Tags



Skew + LA + Inv ❌

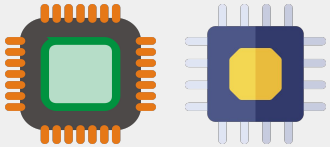


Skew + Inv + GE ❌

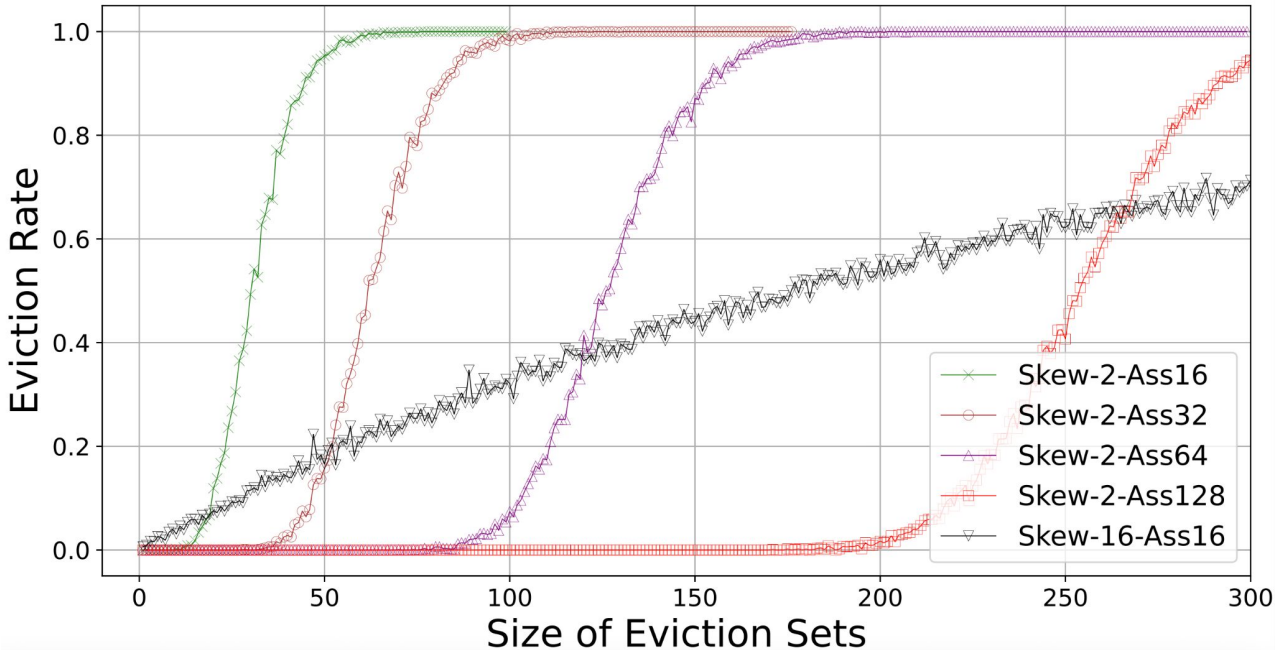


Extra invalid tags can improve security, but they need to be coupled with appropriate knobs and sub knobs

Knob 3: High Associativity

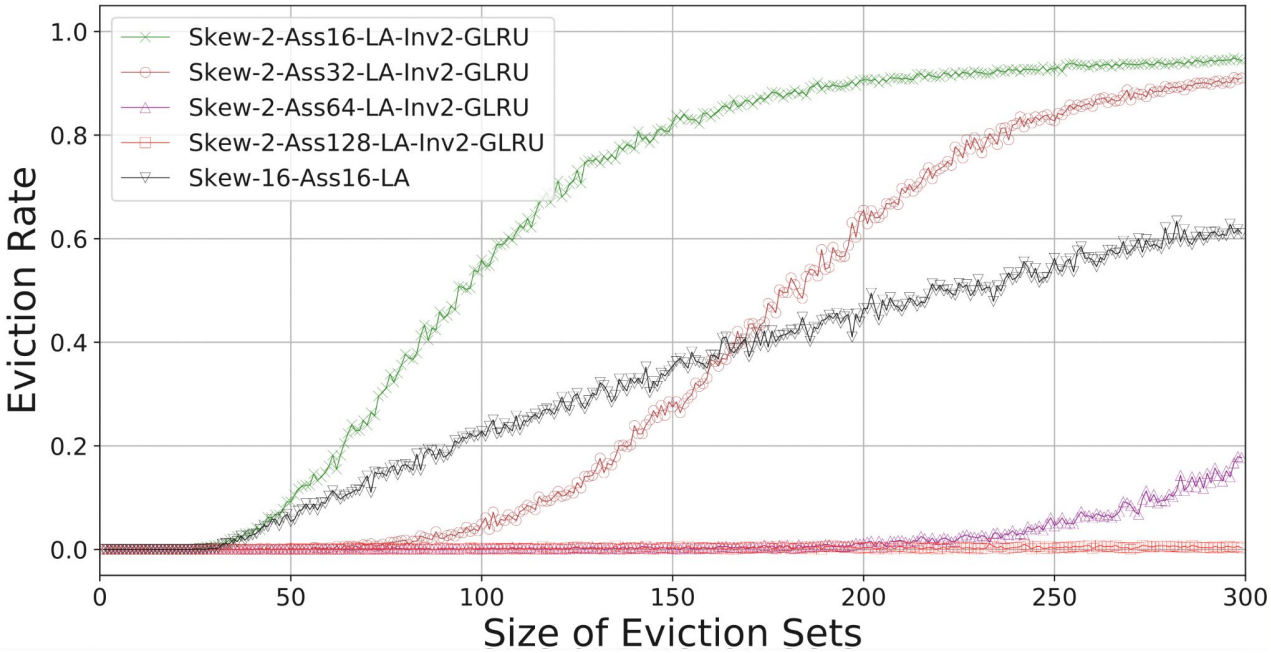


Knob 3: High Associativity



High associativity provides significant security gains, even with just two skews.

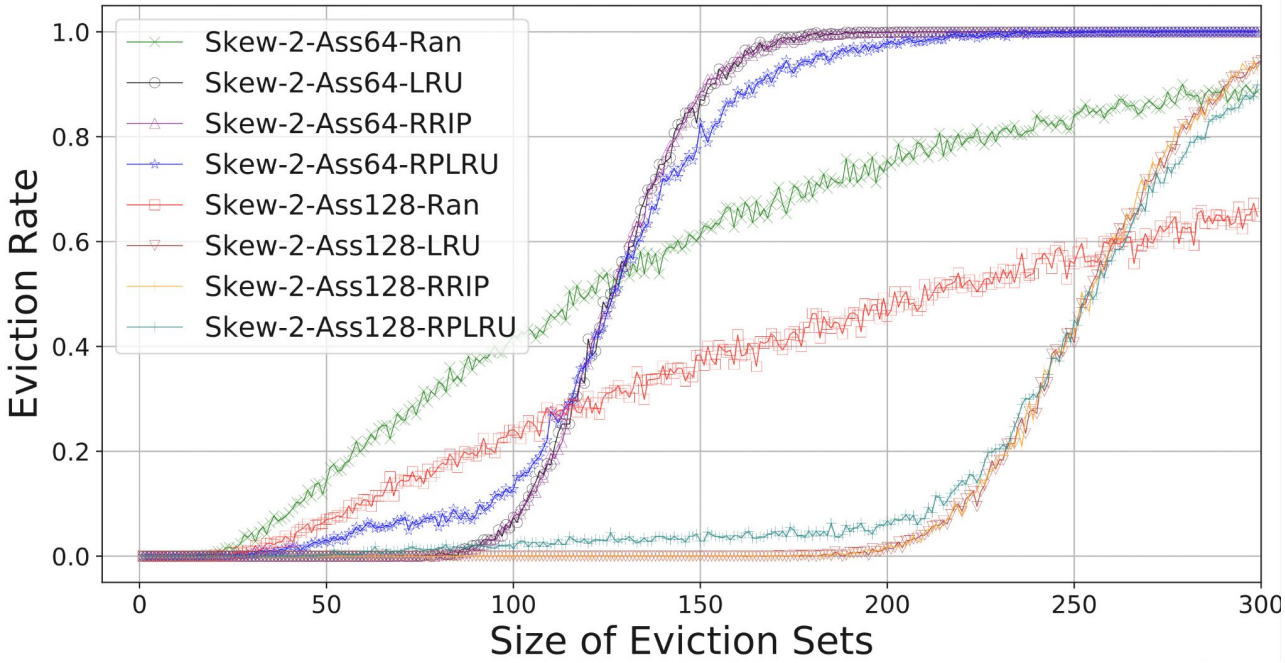
Knob 3: High Associativity



High associativity provides significant security gains, even with just two skews.

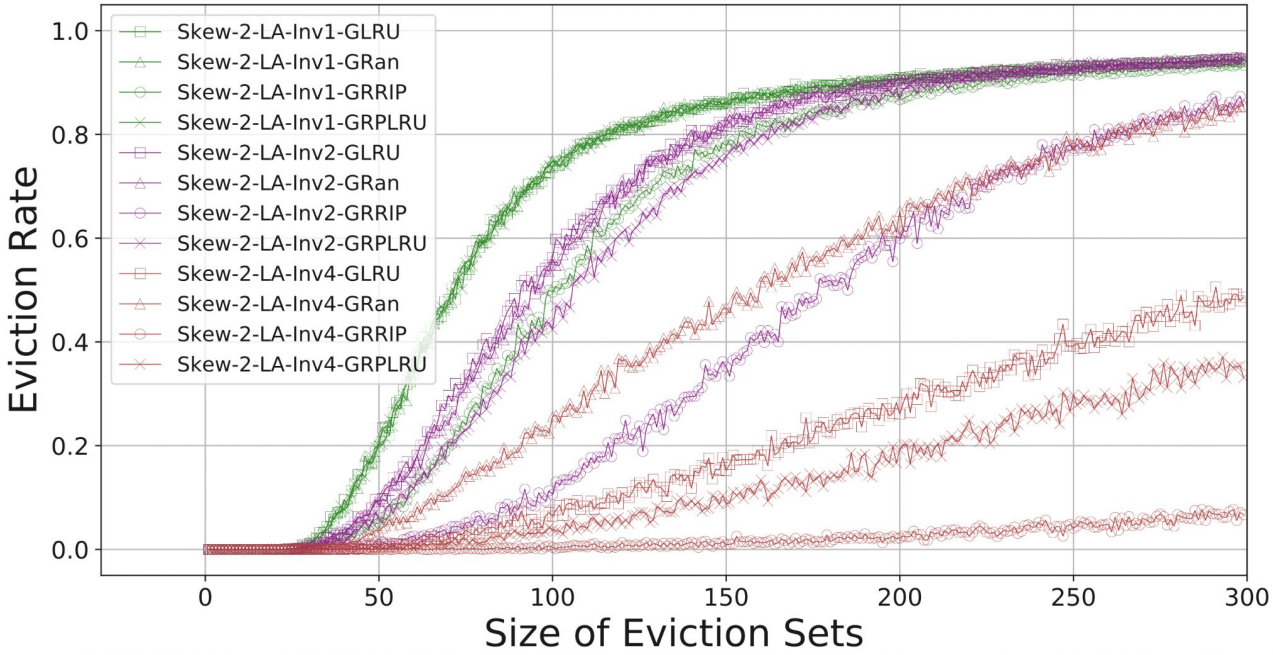
Knob 4: Replacement Policy

Knob 4: Replacement Policy



Random performs worse than LRU and RRIP

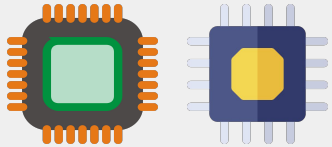
Knob 4: Replacement Policy



Random performs worse than LRU and RRIP

GRPLRU performs similarly to GLRU, while being more practical

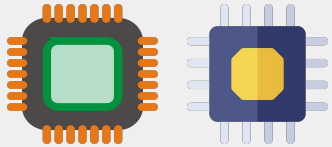
Knob 5: Remapping



Number of LLC Evictions

- Conflict Testing
- Prime, Prune, and Probe

Knob 5: Remapping

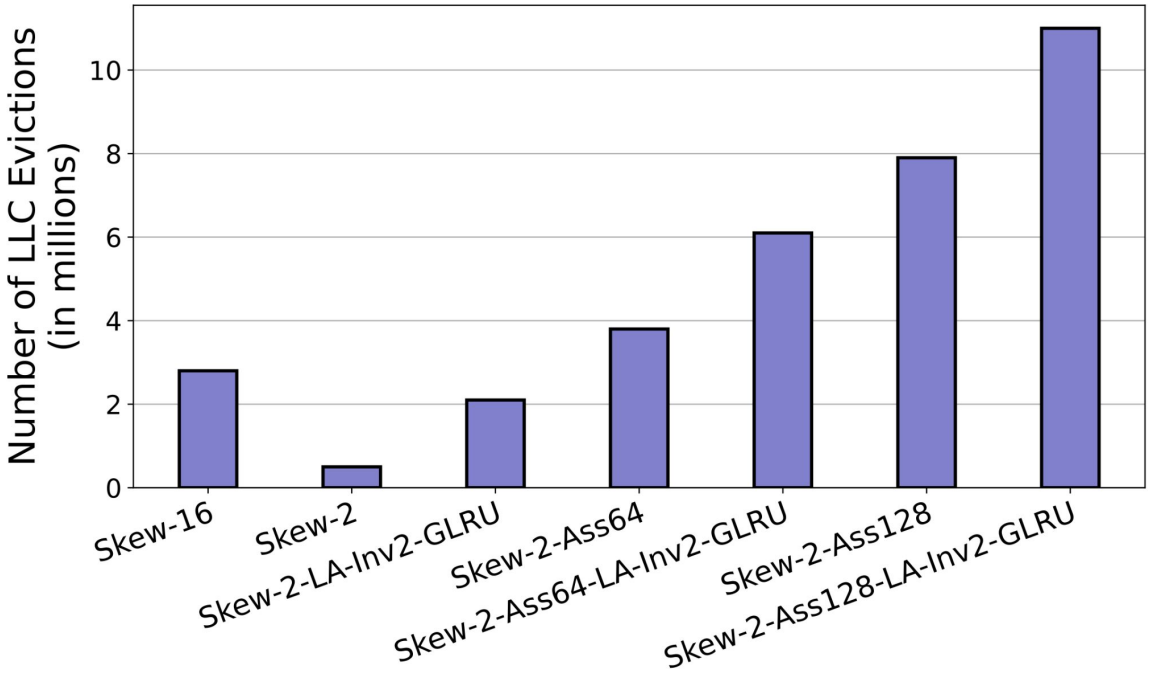


Number of LLC Evictions

- Conflict Testing
- Prime, Prune, and Probe

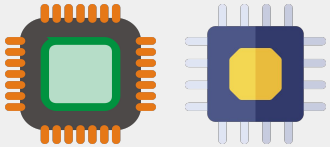
Conflict Testing is faster than PPP

Knob 5: Remapping

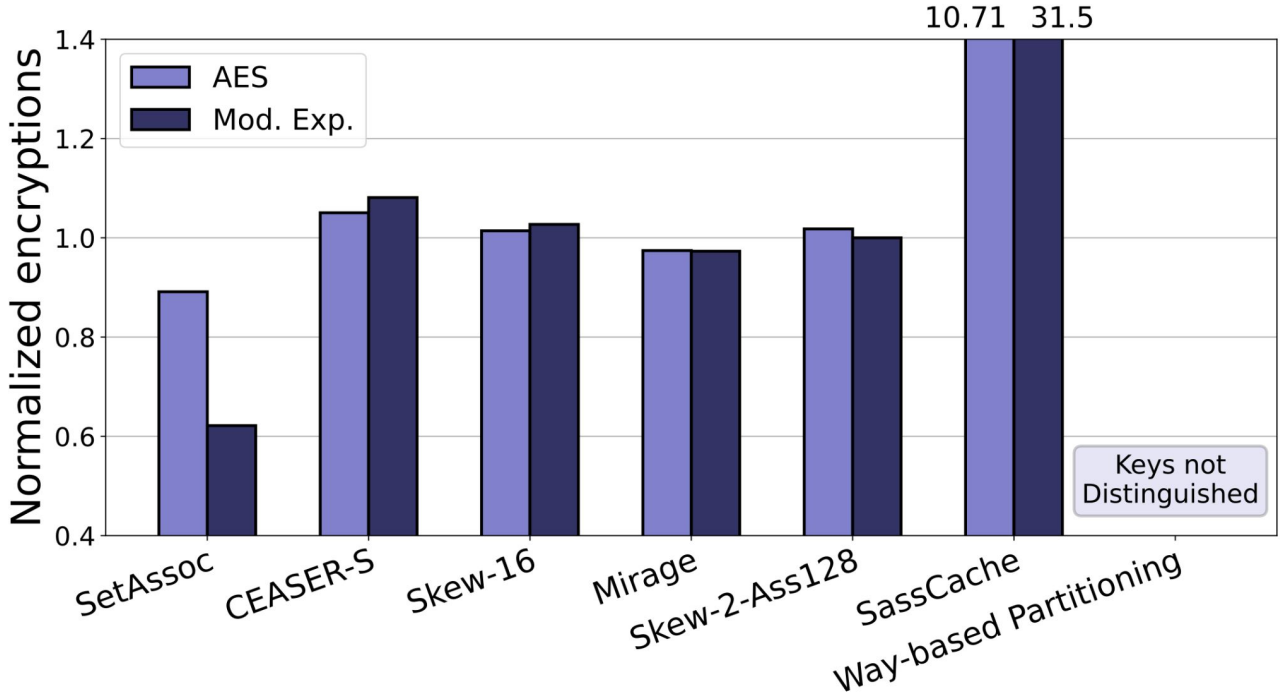


High associativity designs have higher remapping periods compared to CEASER-S and Skew-16. Inv + LA + GE improves this further.

Evaluation against Occupancy Attacks

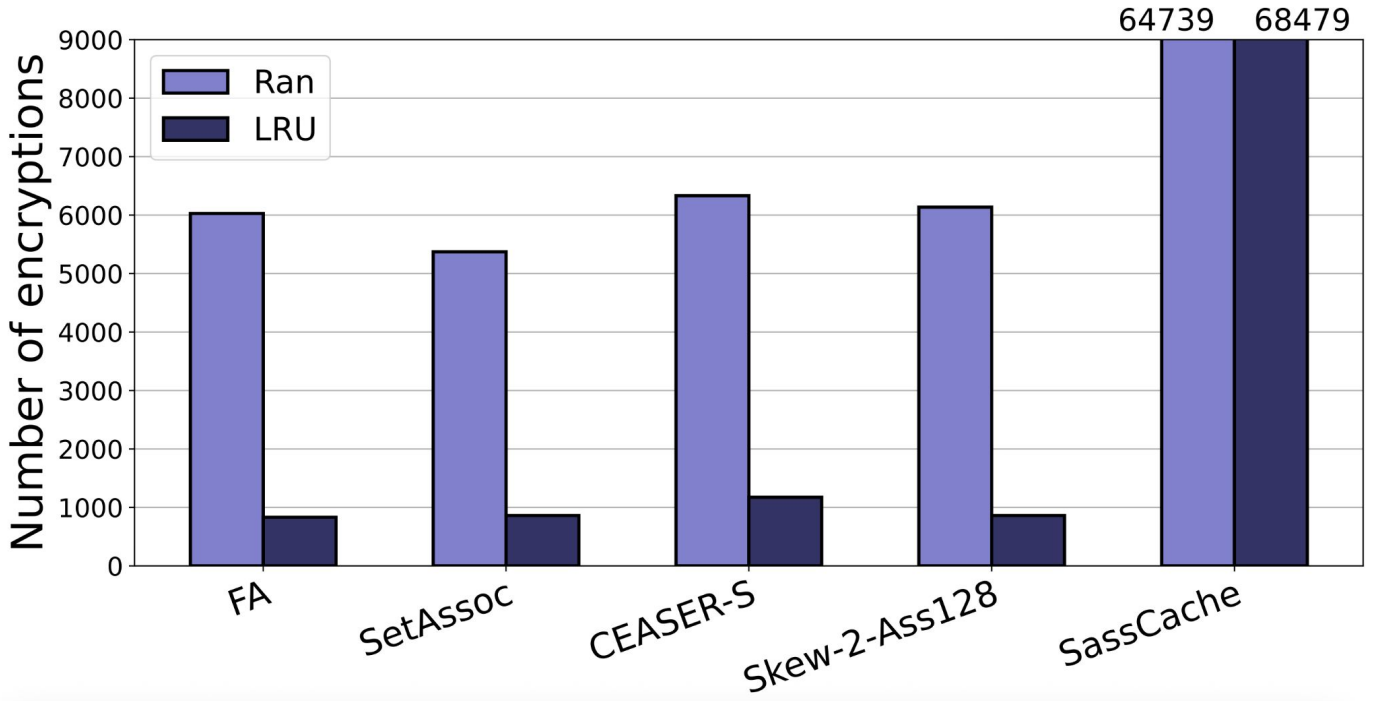


Evaluation against Occupancy Attacks



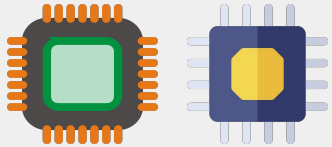
Only partitioning-based mitigations are effective

Evaluation against Occupancy Attacks



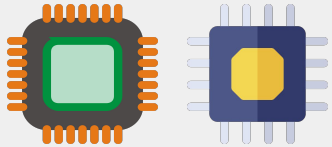
Random policy > Deterministic policy; Local eviction ~ Global eviction

Impact of Warm-up State



```
Loop start
1. Select random target x
2. Generate eviction set E for x
3. Access x
4. Access E
5. Check whether x is evicted
end
```

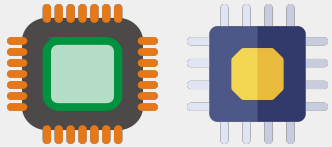
Impact of Warm-up State



Cache initialized as empty

```
Loop start
1. Select random target x
2. Generate eviction set E for x
3. Access x
4. Access E
5. Check whether x is evicted
end
```

Impact of Warm-up State

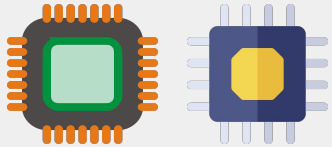


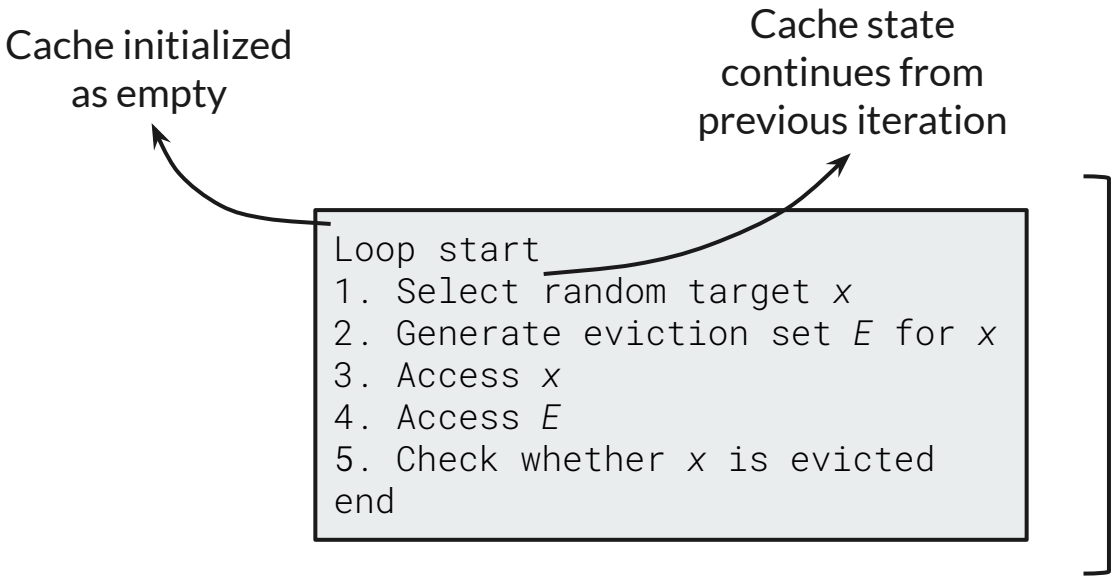
Cache initialized as empty

Cache state continues from previous iteration

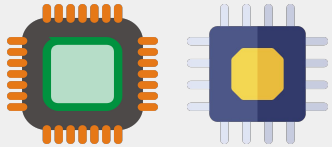
```
Loop start
1. Select random target x
2. Generate eviction set E for x
3. Access x
4. Access E
5. Check whether x is evicted
end
```

Impact of Warm-up State



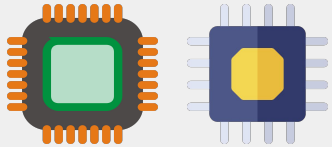


Impact of Warm-up State



```
Loop start
1. Reset cache warm-up state
2. Select random target x
3. Generate eviction set E for x
4. Access x
5. Access E
6. Check whether x is evicted
end
```

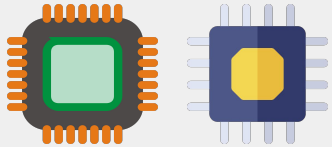
Impact of Warm-up State



```
Loop start
1. Reset cache warm-up state
2. Select random target x
3. Generate eviction set E for x
4. Access x
5. Access E
6. Check whether x is evicted
end
```

Why not use this technique by default?

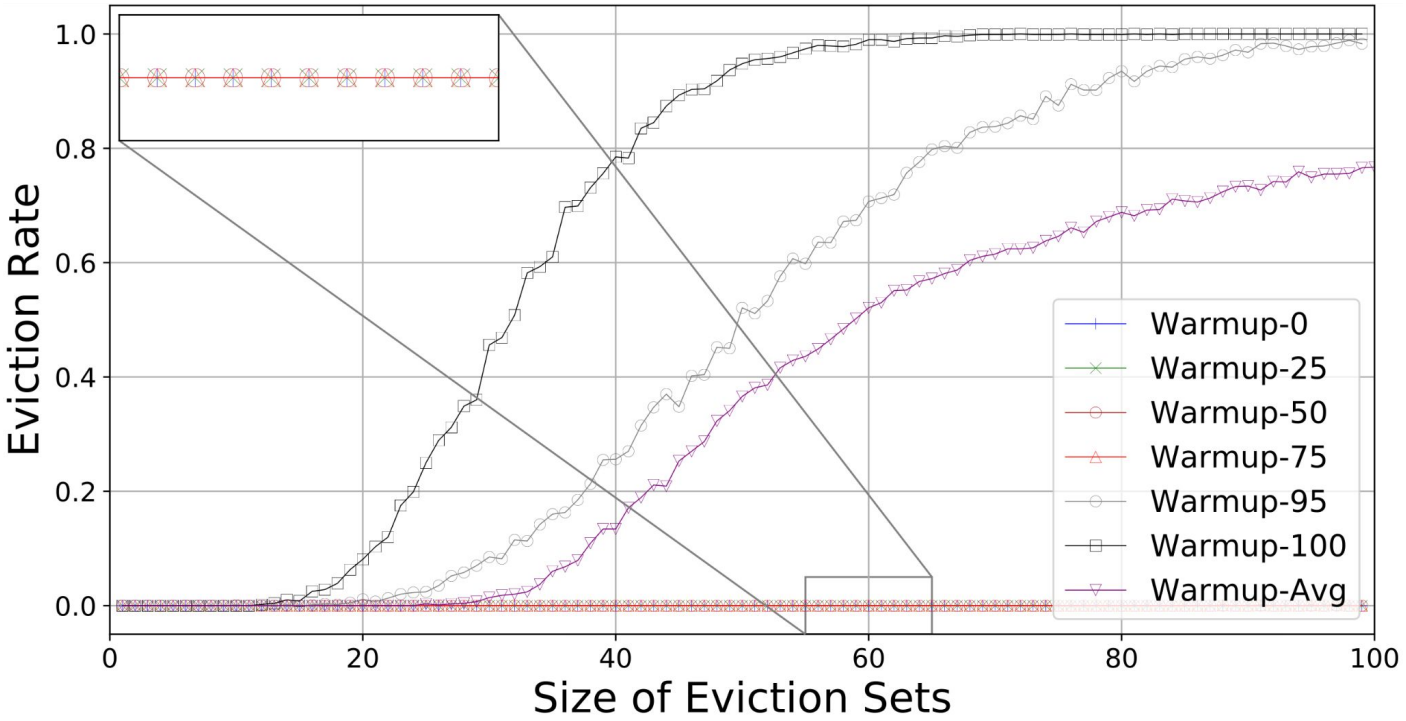
Impact of Warm-up State



```
Loop start
1. Reset cache warm-up state
2. Select random target x
3. Generate eviction set E for x
4. Access x
5. Access E
6. Check whether x is evicted
end
```

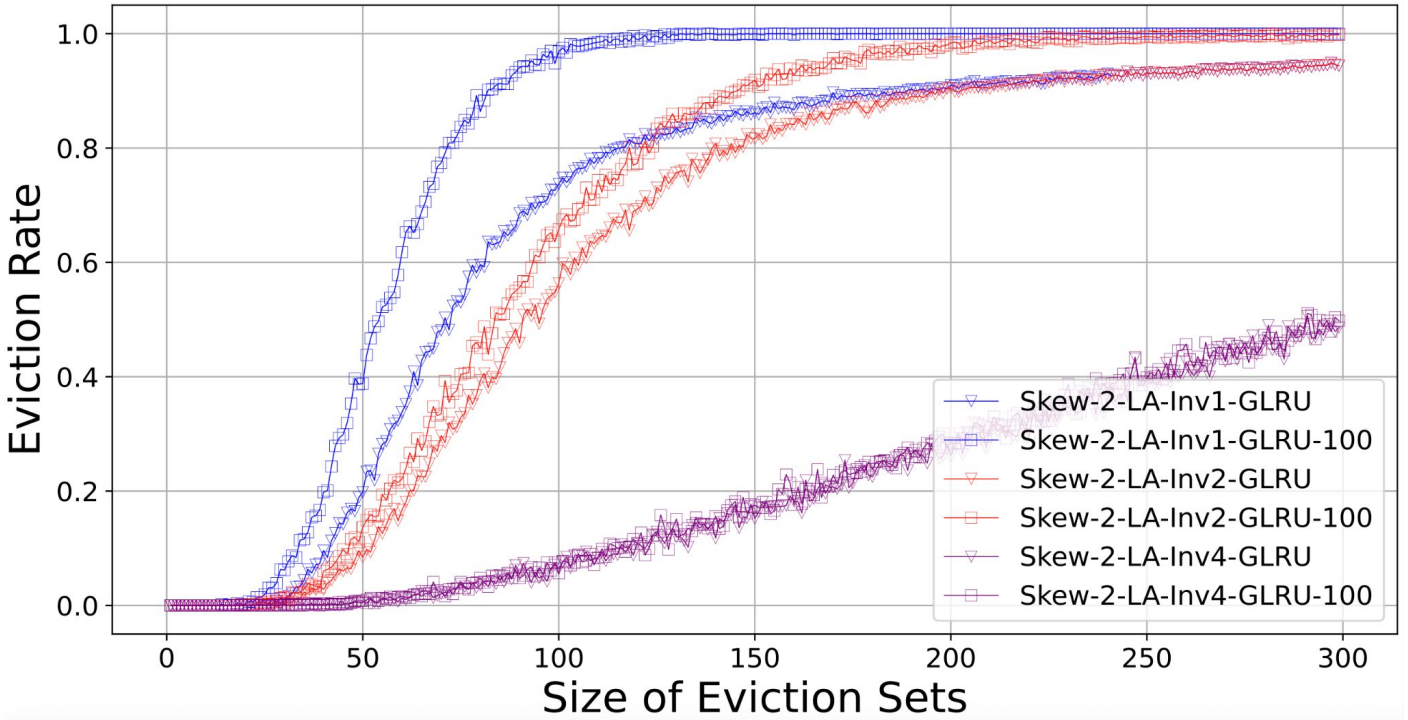
Why not use this technique by default?
>10x slower than the original

Impact of Warm-up State



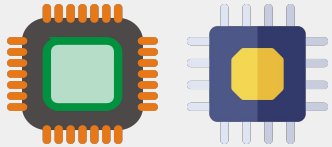
Cache warm-up state has a significant impact on security

Impact of Warm-up State



Cache warm-up state has a significant impact on security

Evaluating Design Trade-offs



Security

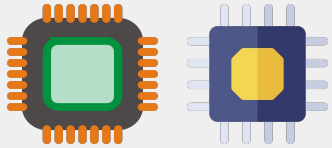
Knobs Used

Performance

Logic

Power

Evaluating Design Trade-offs



Security Knobs Used Performance Logic Power

Design	LLC Evictions Needed to Create Eviction Set	Knobs Used	Performance Overhead	Logic Overhead	Dynamic Power Overhead	Static Power Overhead
Skew-2 (CEASER-S)	0.5 million	Skews, Remapping	-1.3%	1.9%	2.5%	2%
Skew-16 (ScatterCache)	2.8 million	Large number of skews, Remapping	0.1%	1.7%	0.5%	1.4%
Skew-2-LA-Inv2-GLRU	2.3 million	Skews, Load-aware, Invalid Tags, Global Eviction, Remapping	1.3%	1.9%	5.2%	2%
Mirage	Not Possible	Skews, Load-aware, Invalid Tags, Global Eviction	0.2%	18.6%	-0.2%	19.6%
Skew-2-Ass64	3.8 million	Skews, High Associativity, Remapping	-2.1%	2.3%	1.8%	3.3%
Skew-2-Ass128	7.9 million	Skews, High Associativity, Remapping	-2.2%	2.4%	1.7%	6.4%
Skew-2-Ass64-LA-Inv2-GLRU	6.1 million	Skews, High Associativity, Load-aware, Invalid Tags, Global Eviction, Remapping	-2.2%	2.3%	1.8%	3.3%
Skew-2-Ass128-LA-Inv2-GLRU	11.0 million	Skews, High Associativity, Load-aware, Invalid Tags, Global Eviction, Remapping	-2.5%	2.5%	1.7%	6.4%
SassCache (coverage = 39%)	Not Possible	Skews, Soft Partitioning	2.3%	2.4%	7.4%	1.2%

Thank You!

Supported by TrustLab, IIT Bombay