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

Anubhav Bhatla, Hari Rohit Bhavsar, Sayandeep Saha, Biswabandan Panda
Indian Institute of Technology Bombay
bhatlaanubhav2001@gmail.com haribhavsar@cse.iitb.ac.in
sayandeepsaha@cse.iitb.ac.in biswa@cse.iitb.ac.in

CASPER SHArC Artifact IIT Bombay

Overview

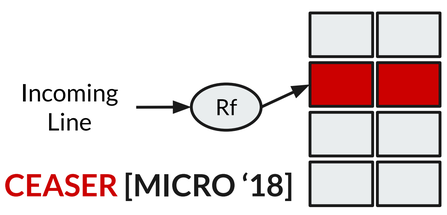
- We **systematize** the design space for secure randomized caches by identifying key **security knobs**
- We perform security analysis of each knob against **conflict-based attacks**. We also study which **combinations** of these knobs work
- We analyze these knobs against full- and low-**occupancy-based attacks** and compare them with **partitioning**-based designs

Background

Conflict-based attacks: (e.g., PRIME+PROBE [S&P '15]) Attacker *primes* LLC with **eviction set** to create conflict with the victim's data; ***probing latency*** reveals victim's cache accesses.
Occupancy-based attacks: (e.g., Website Fingerprinting [USEC '19]) Attacker observes victim's cache usage via changes in its LLC working set, leading to **coarse-grained leakage without an eviction set.**
Low-occupancy-based attacks: [USEC '25] Uses a much **smaller** (as low as 10% cache size) **buffer size** to observe the victim cache usage.

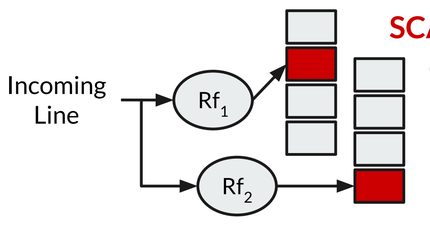
Popular Secure Randomized Designs

Randomization + Remapping

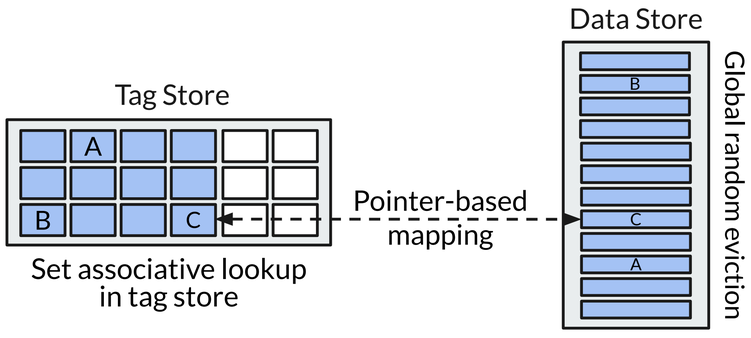


CEASER [MICRO '18]

Skews + Randomization + Remapping



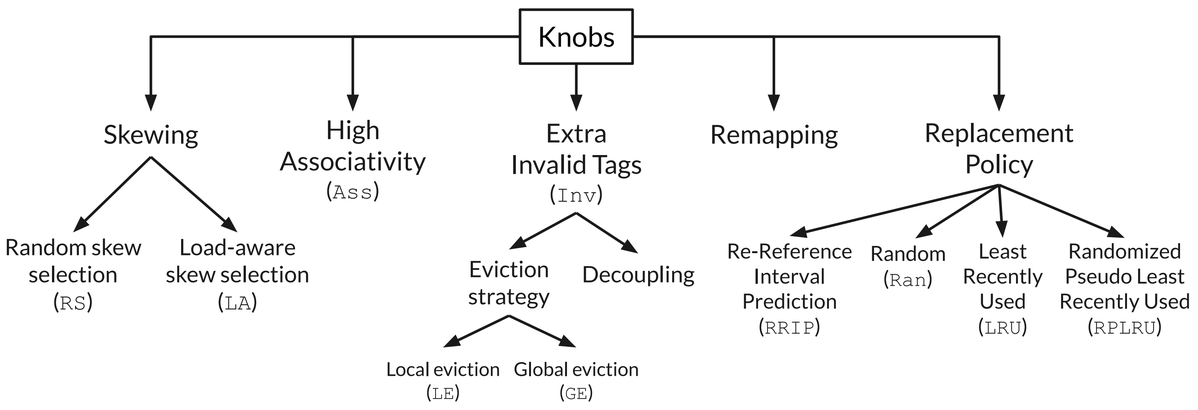
SCATTERCACHE [S&P '19]
CEASER-S [ISCA '19]



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

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

Security Knobs and Metrics Used



METRIC I: Eviction Rate

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

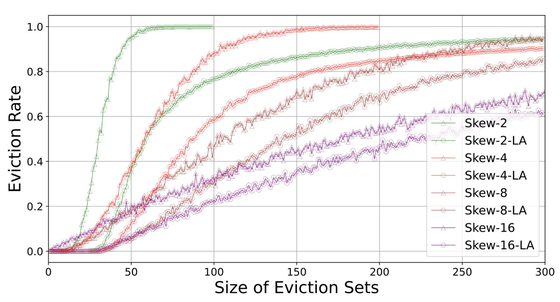
```
Loop start
1. Select random target x
2. Generate eviction set E for x
3. Access x
4. Access E
5. Check whether x is evicted
end
```

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)

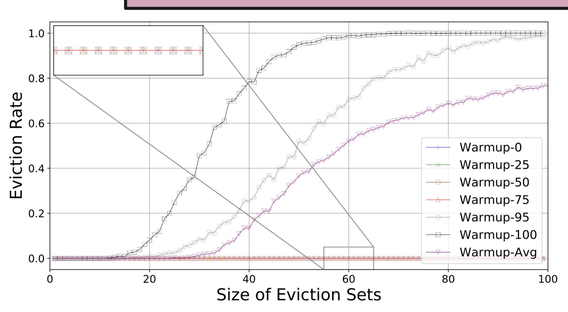
Knob 1: Skewing

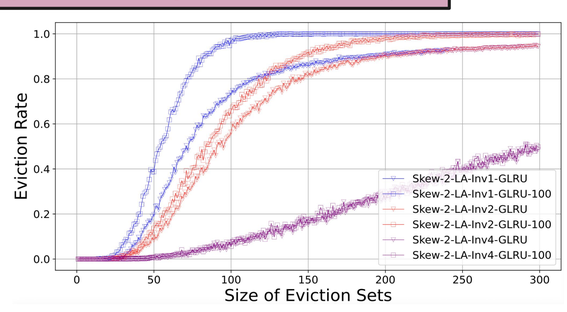


Skewing helps improve security

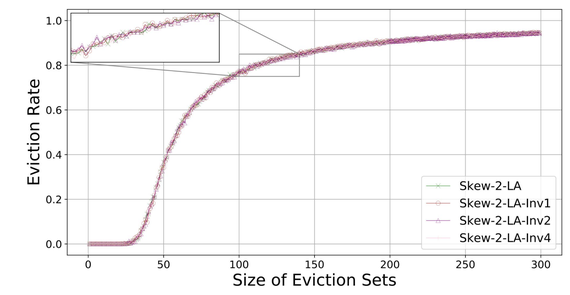
LA does better than Random (don't forget the warm-up state!)

Cache **warm-up state** has a **significant impact** on security

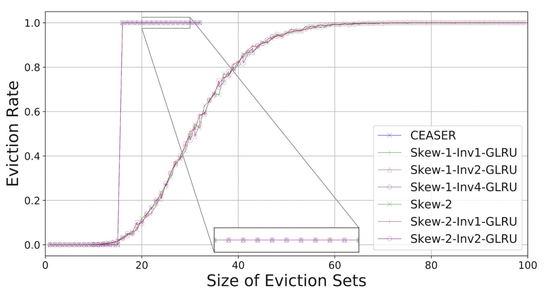




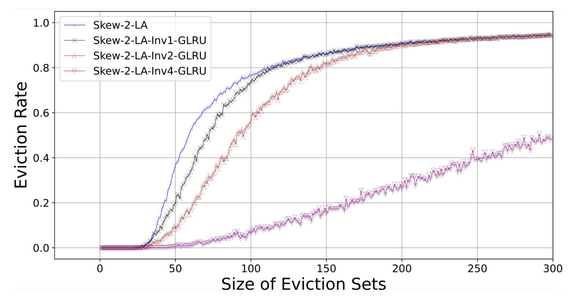
Knob 2: Extra Invalid Tags



Skew + LA + Inv ❌

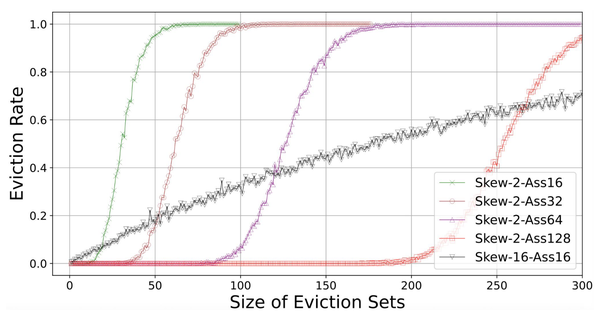


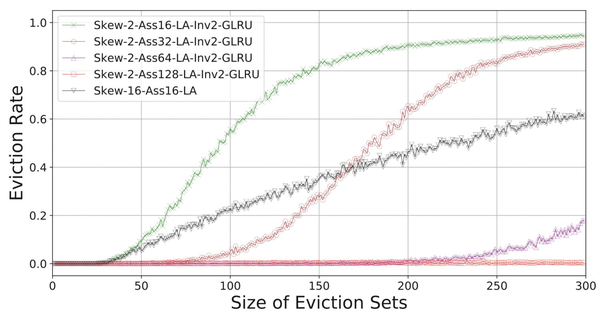
Skew + Inv + GE ❌



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

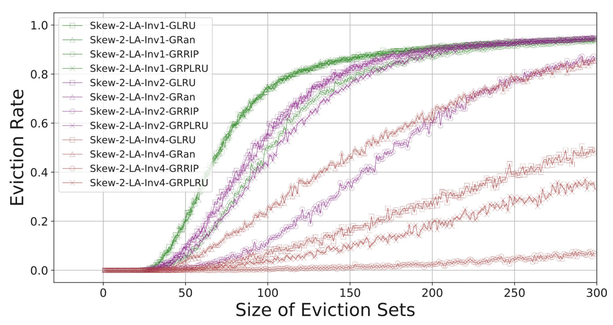
Knob 3: High Associativity

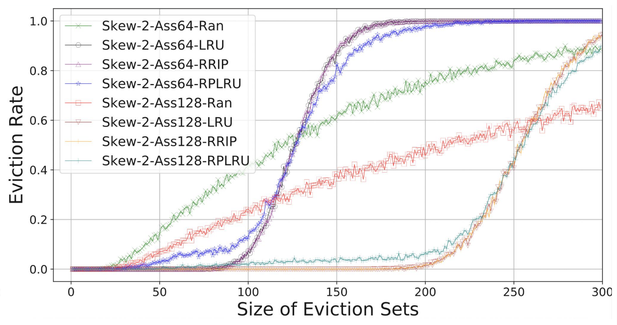




High associativity provides **significant security gains**, even with just two skews.

Knob 4: Replacement Policy

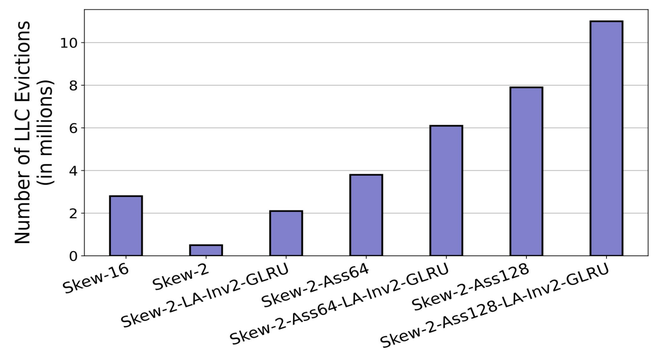




Random performs worse than LRU and RRIP

GRPLRU performs similarly to GLRU, while being **more practical**

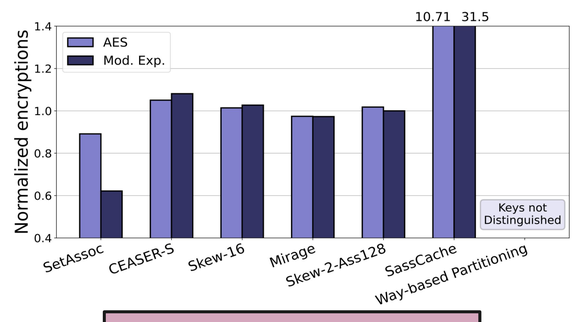
Knob 5: Remapping

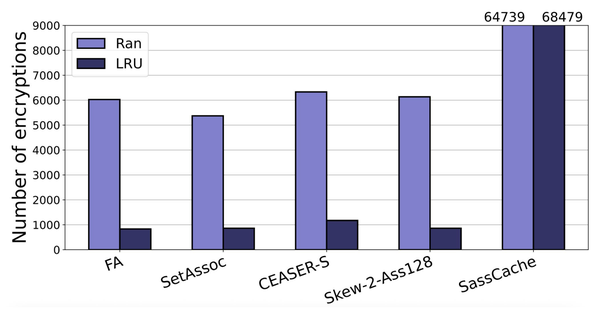


Conflict Testing is **>10x faster** than PPP and requires **less number of LLC evictions**

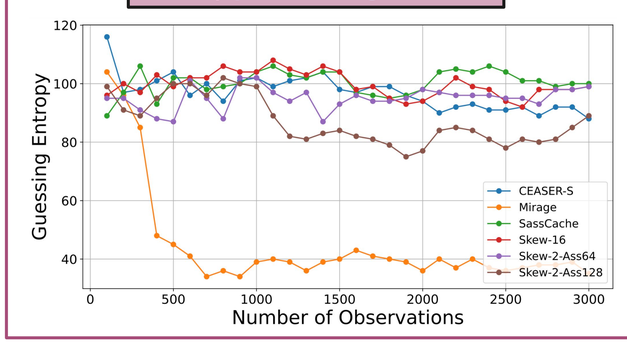
High associativity provides **high remapping periods**. **Inv+LA+GE** improves it further.

Evaluation against Occupancy Attacks





Only **partitioning** works



Full-occupancy: random > deterministic
Full-occupancy: local ~ global eviction
Low-occupancy: local > global eviction

Open Problems:

- Unified security metric for occupancy-based attacks
- SassCache's security against multiple adversary processes

Supported by TrustLab, IIT Bombay