Last-Level Cache Side-Channel Attacks Are Feasible in the Modern Public Cloud

Zirui Neil Zhao, Adam Morrison, Christopher W. Fletcher, Josep Torrellas

University of Illinois Tel Aviv University

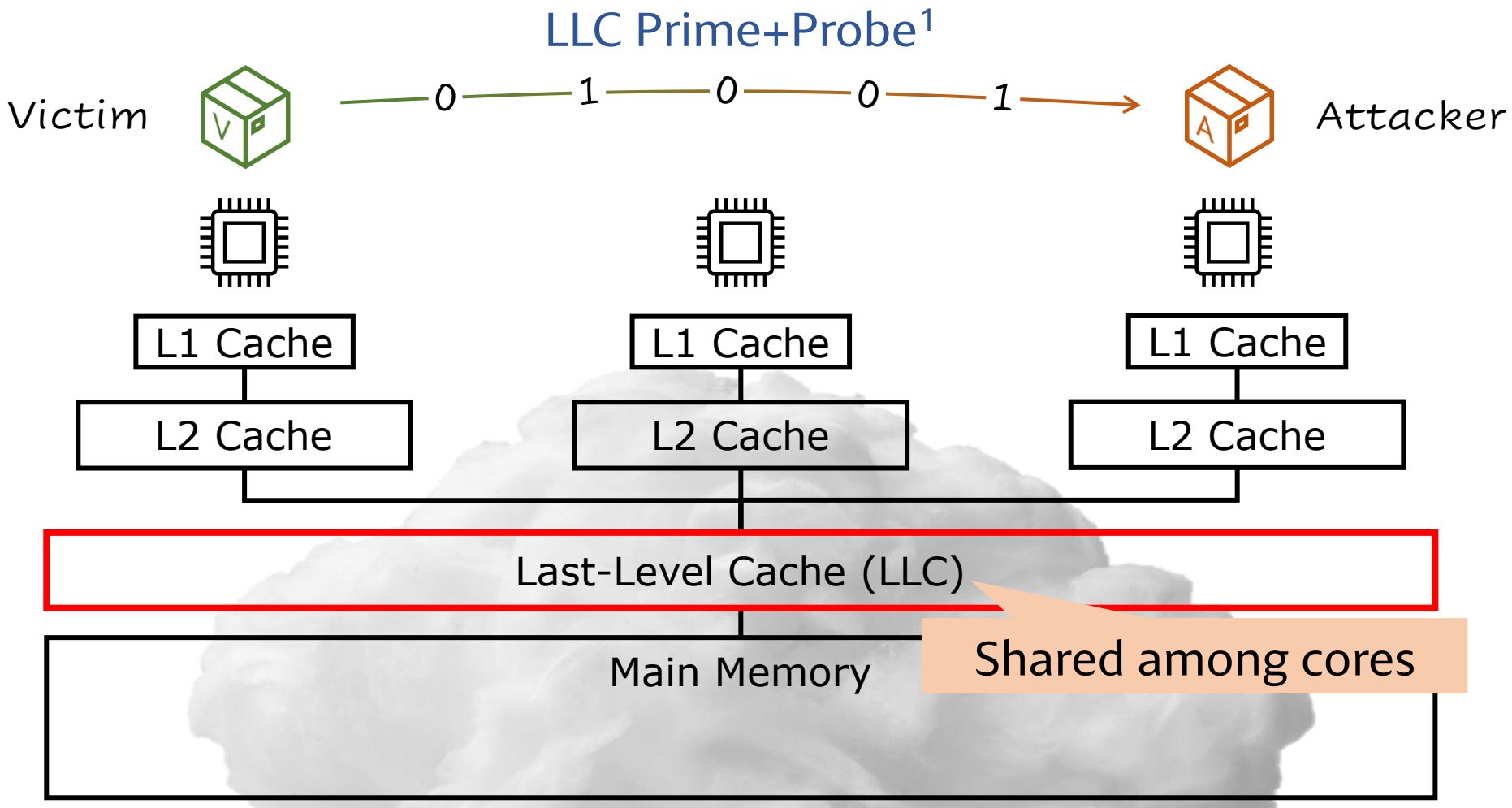
ASPLOS '24 – Session 2B: Side Channels

ILLINOIS

TEL AVIV UNIVERSITY

i-acoma group

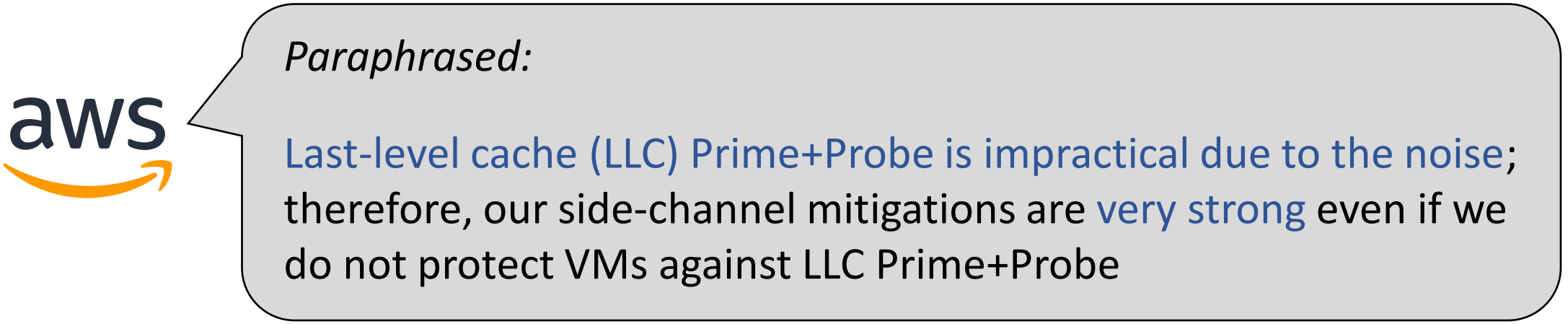
Shared Last-Level Cache (LLC) Enables Information Leakage



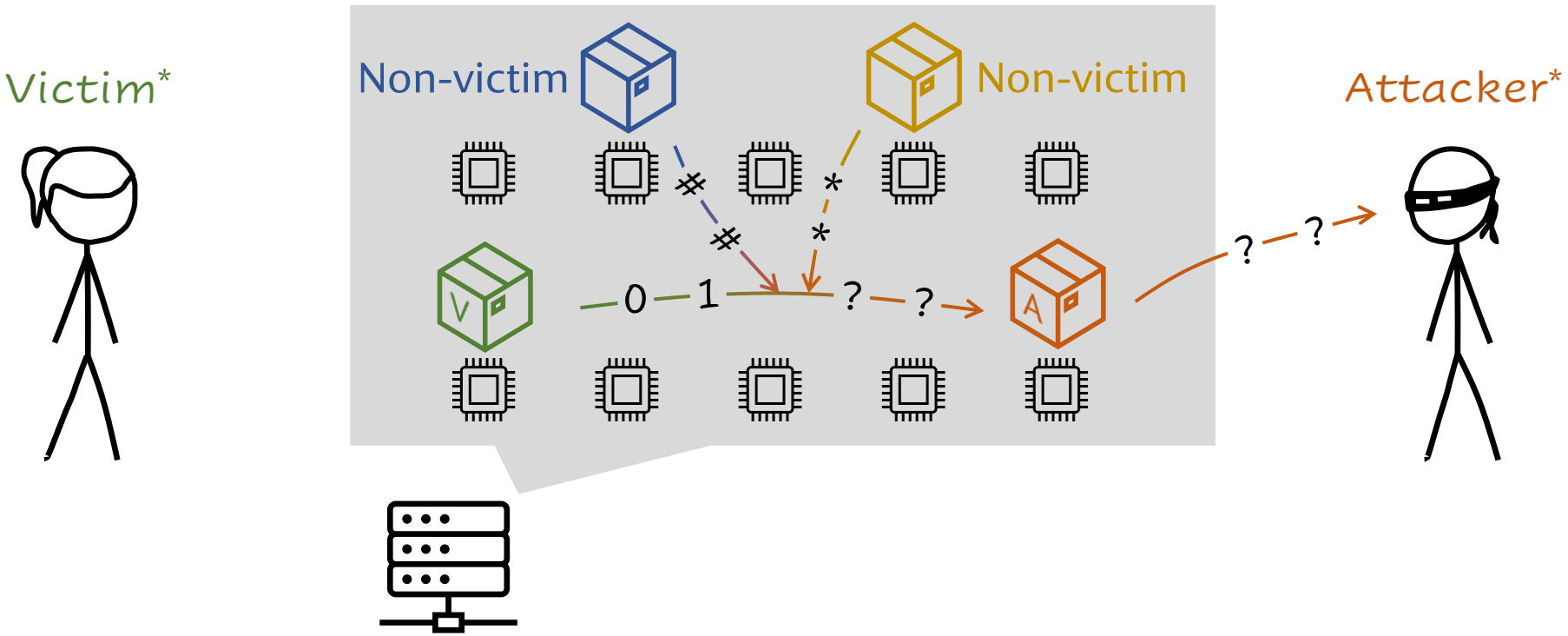
[1]Liu et al., Last-Level Cache Side-Channel Attacks are Practical (S&P 2015)

Cloud Vendors Claim LLC Prime+Probe is Impractical

Example: AWS Whitepaper – The Security Design of the AWS Nitro System (Version November 18, 2022)

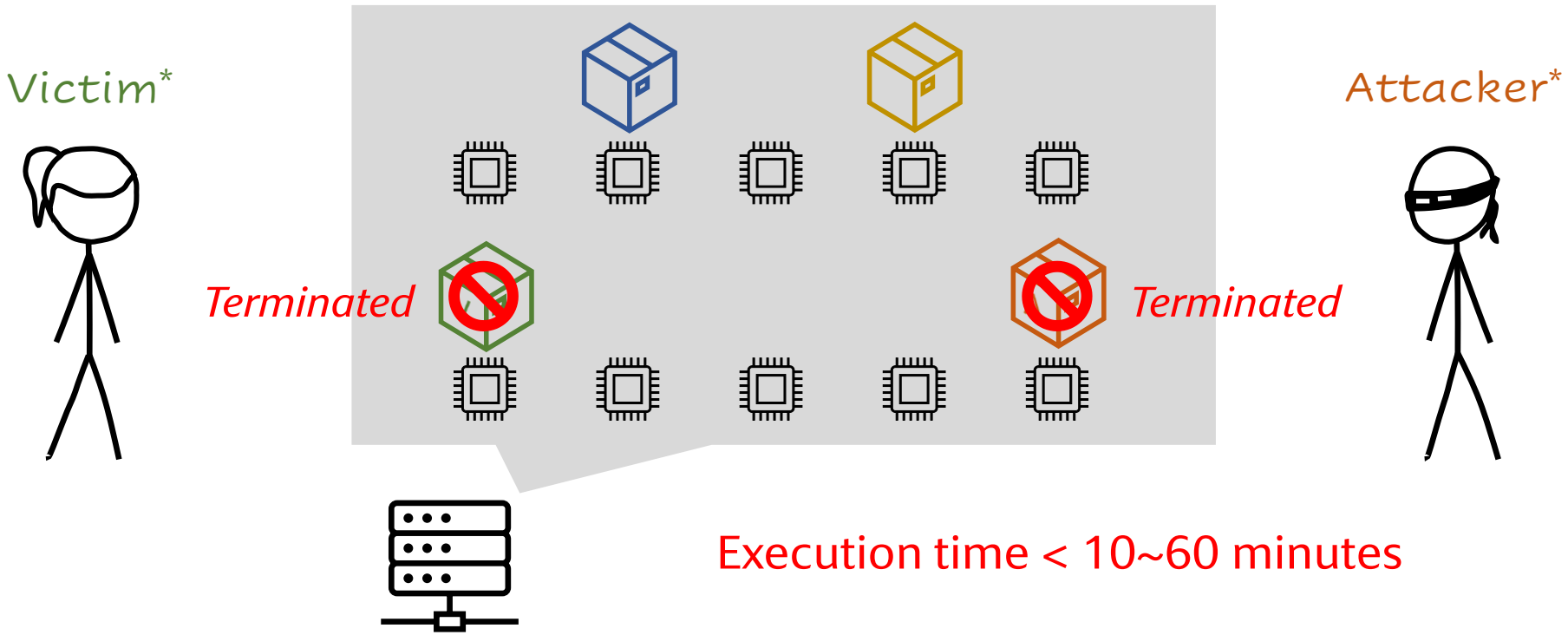


Attacker's Challenge 1: Production Cloud is Noisy



*Characters are based on https://xkcd.com/2176 and https://xkcd.com/1808 (under a CC Attribution-NonCommercial 2.5 License)

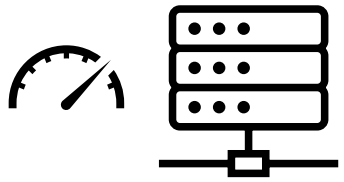
Attacker's Challenge 2: Modern Clouds (e.g., FaaS) are Dynamic



*Characters are based on https://xkcd.com/2176 and https://xkcd.com/1808 (under a CC Attribution-NonCommercial 2.5 License)

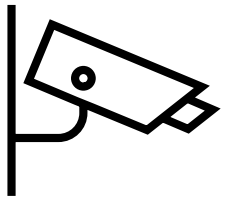
Contributions of This Work (at 31,000 Feet)

Cross-tenant information leakage with LLC Prime+Probe Google Cloud

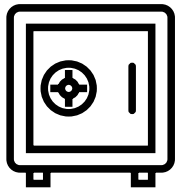


1. Fast LLC Channel Setup

> 10 hours → 2.4 minutes



2. Noise-Resilient Victim Monitoring



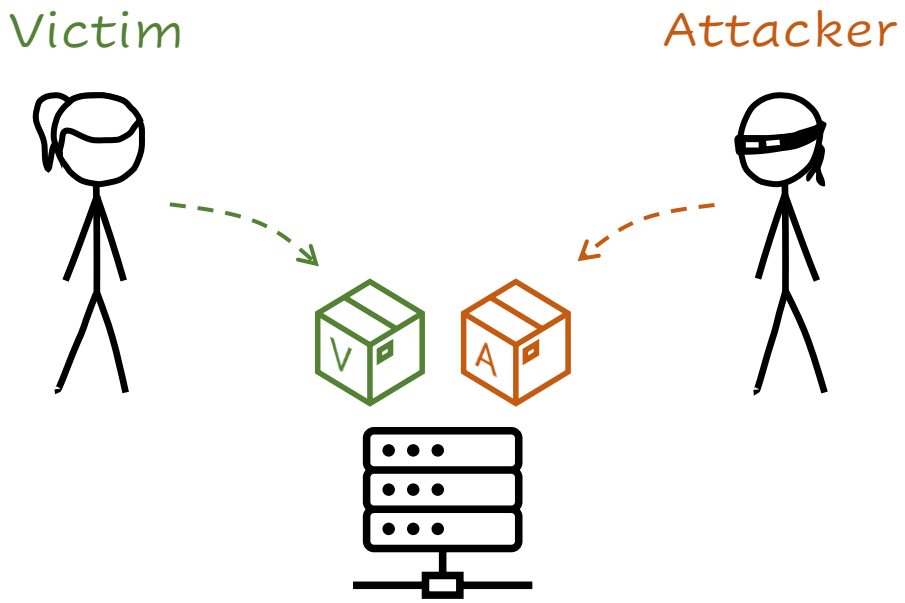
3. Information Extraction

+ Google filed a **critical-level bug** to their product team

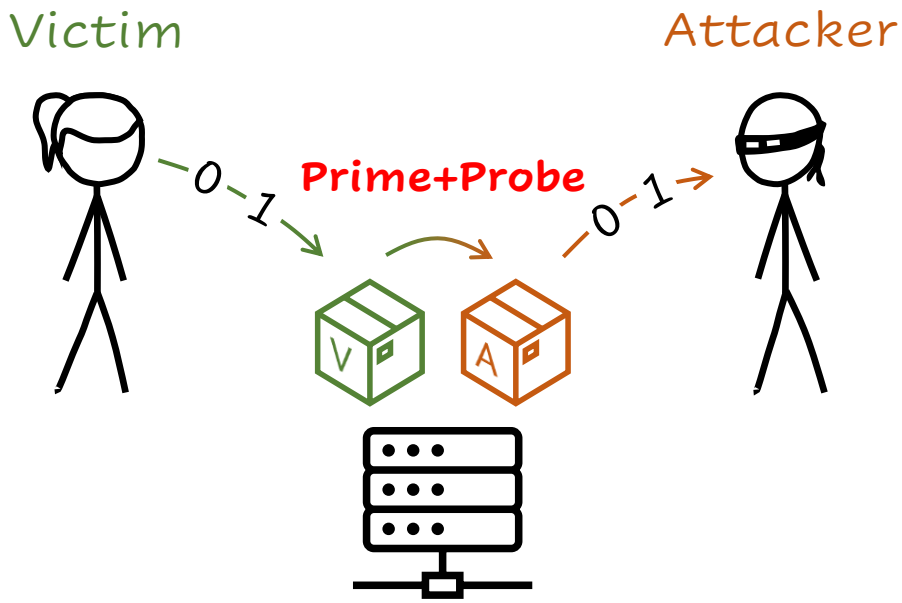
+ AWS revised their whitepaper on February 15, 2024

Threat Model & Assumptions

✓ **Step 1:** Co-Location



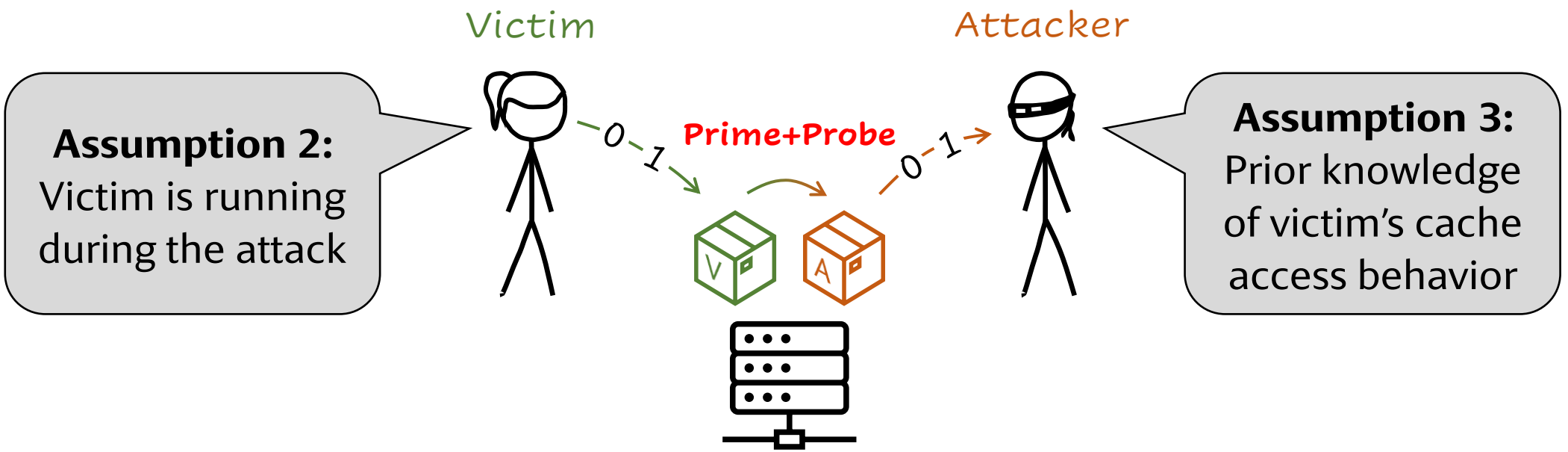
Step 2: LLC Prime+Probe



✓ Everywhere All at Once: Co-Location Attacks on Public Cloud FaaS
(ASPLOS '24 – Session 1D)

Threat Model & Assumptions

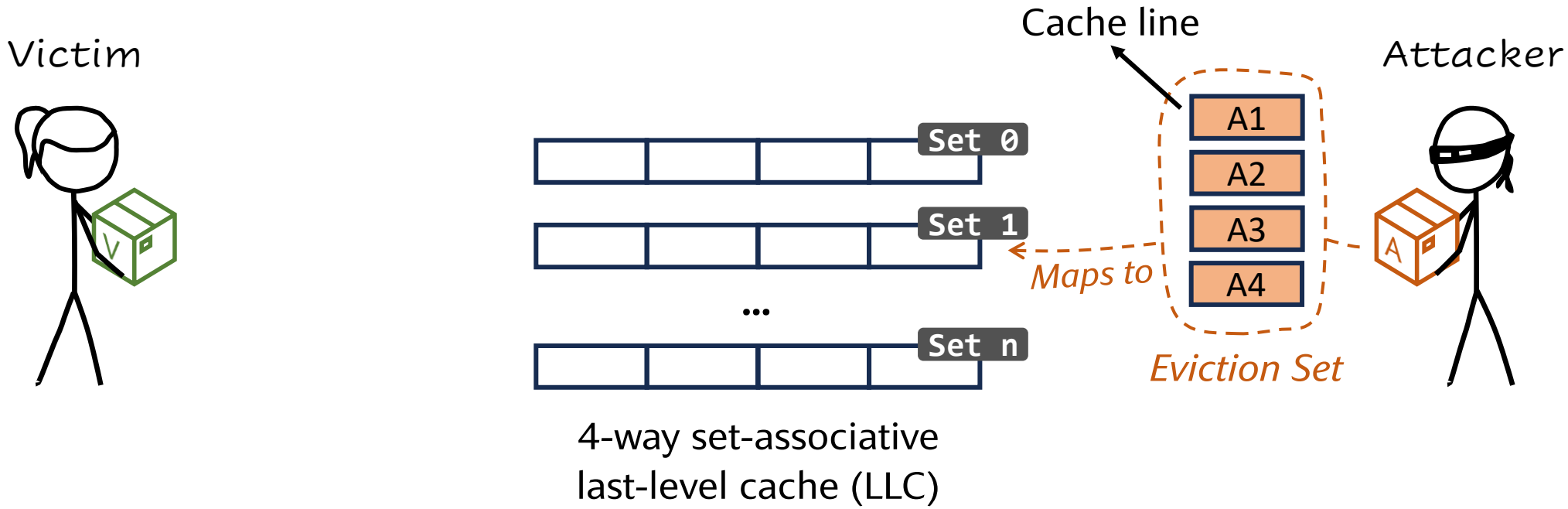
Step 2: LLC Prime+Probe



Assumption 1: Co-location

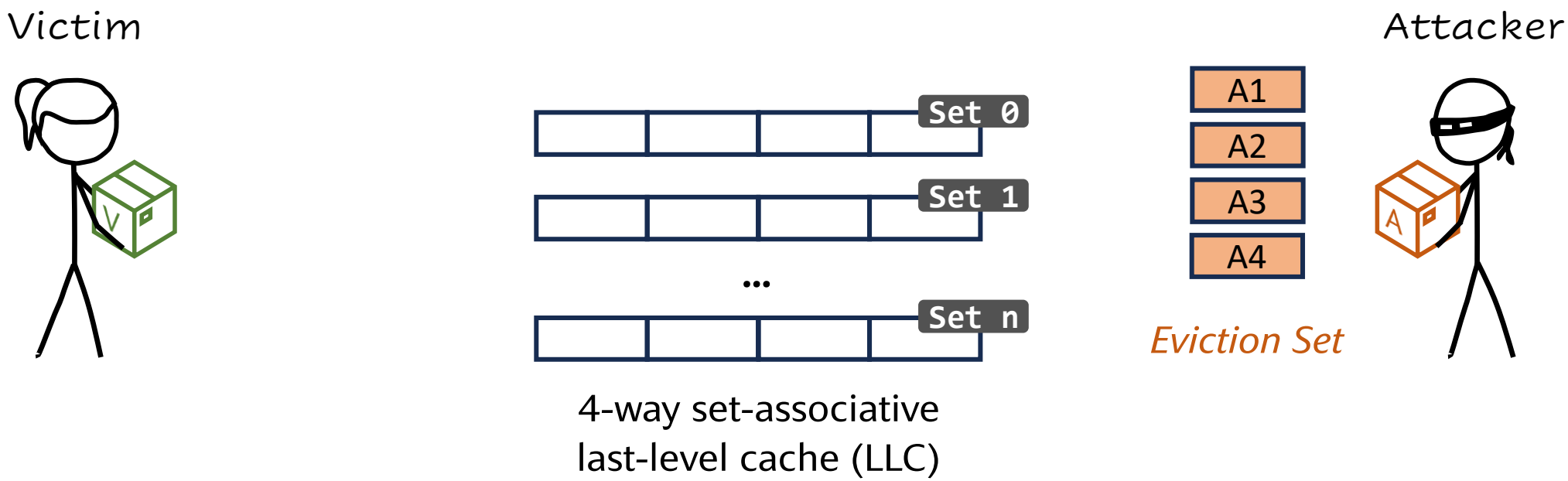
Background: LLC Prime+Probe Attack

Eviction set: A set of cache lines that fully occupy a cache set



Background: LLC Prime+Probe Attack

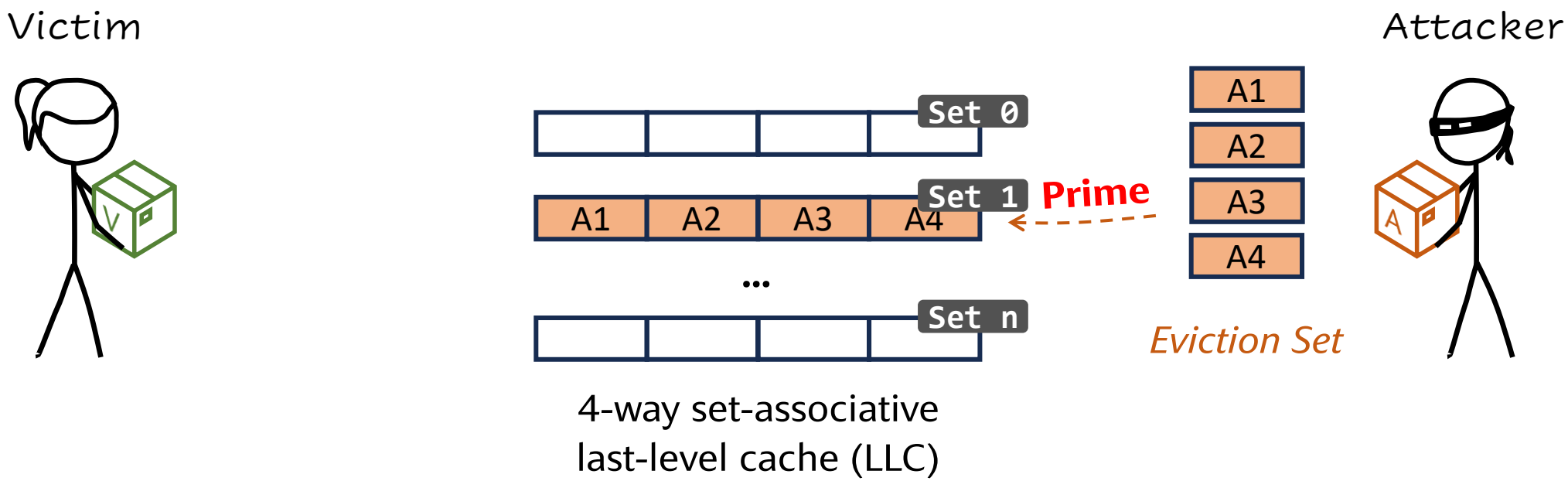
Eviction set ⇒ Monitor memory accesses to an LLC set with **Prime+Probe**



4-way set-associative
last-level cache (LLC)

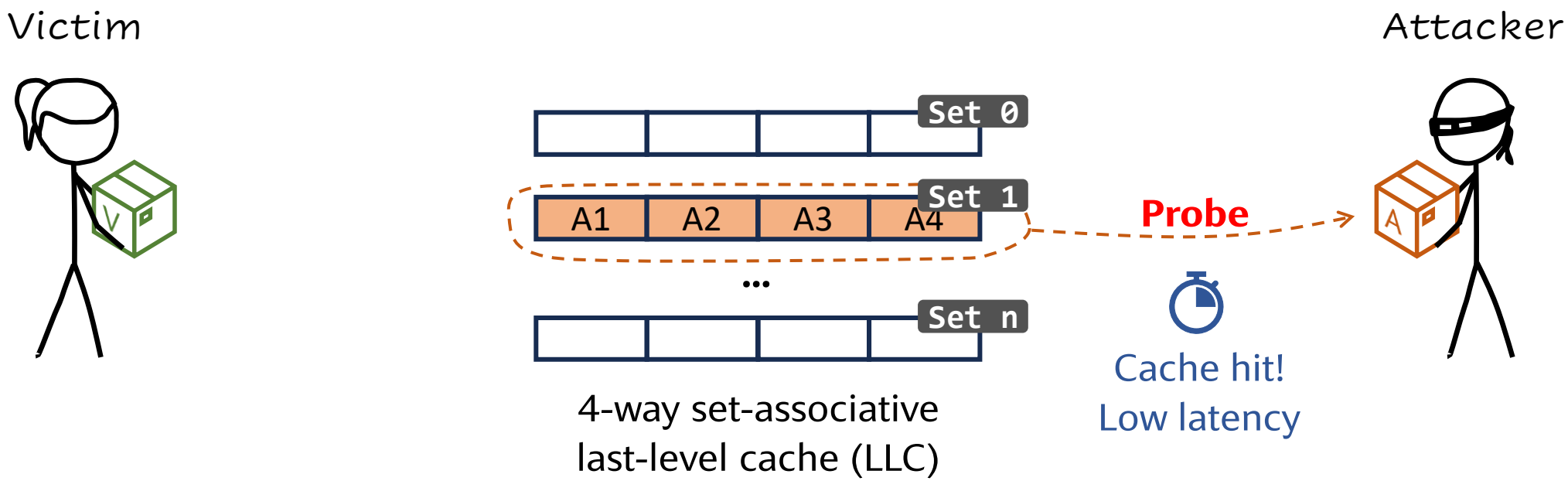
Background: LLC Prime+Probe Attack

Eviction set ⇒ Monitor memory accesses to an LLC set with **Prime+Probe**



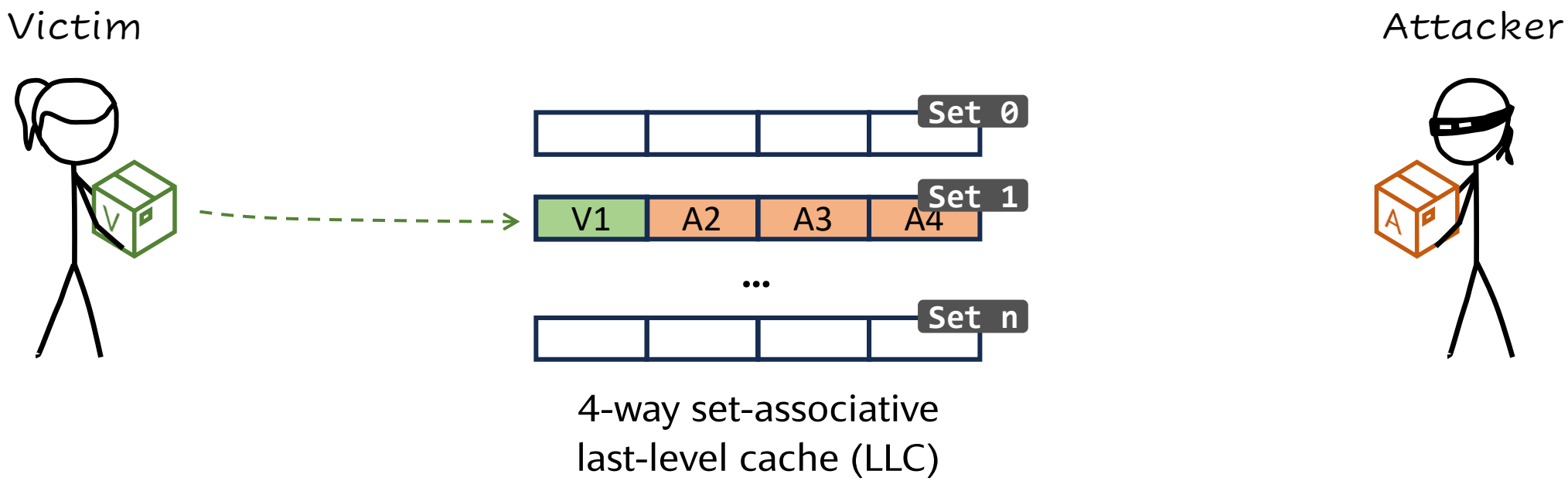
Background: LLC Prime+Probe Attack

Eviction set ⇒ Monitor memory accesses to an LLC set with **Prime+Probe**



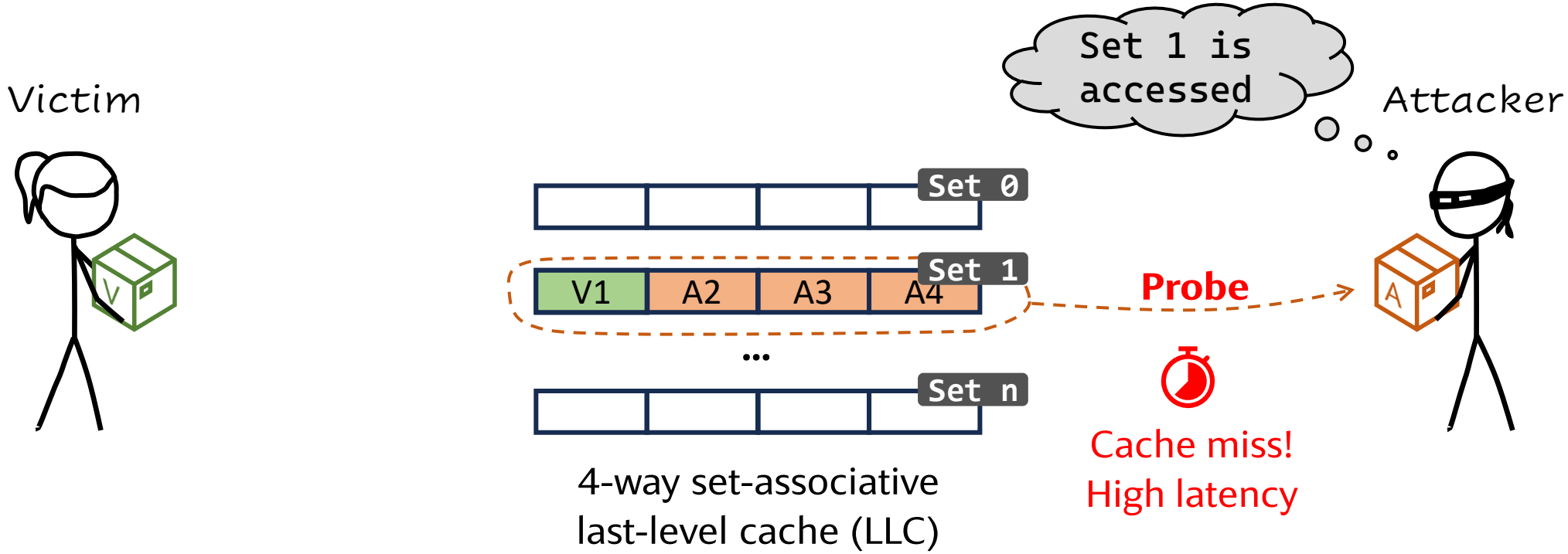
Background: LLC Prime+Probe Attack

Eviction set ⇒ Monitor memory accesses to an LLC set with **Prime+Probe**



Background: LLC Prime+Probe Attack

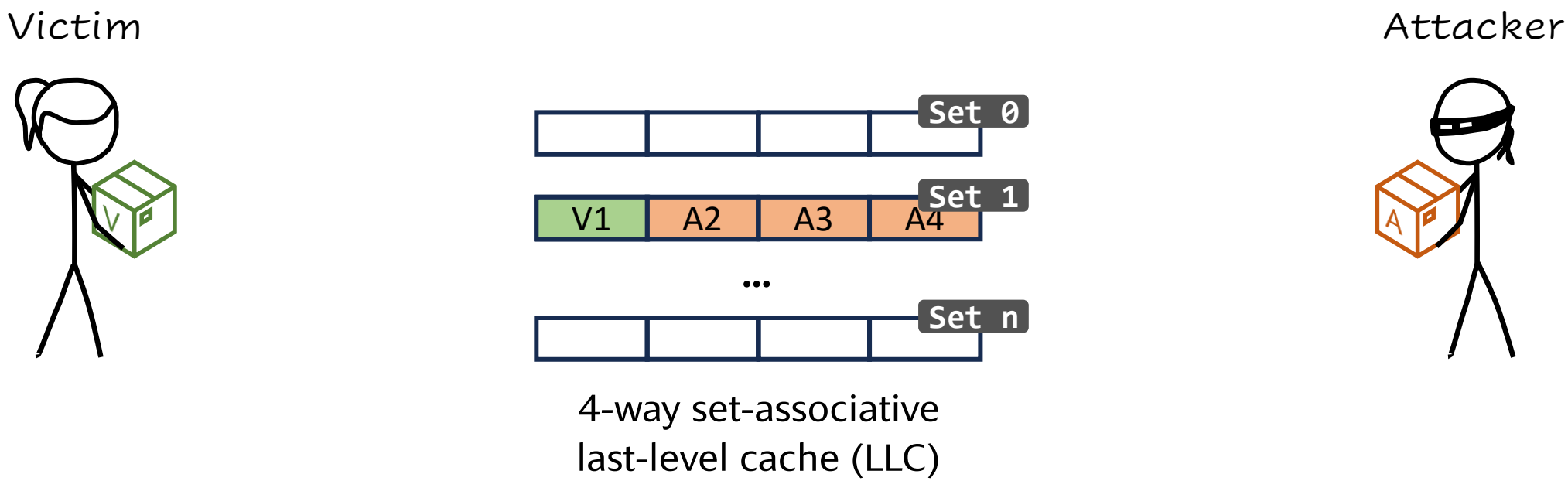
Eviction set ⇒ Monitor memory accesses to an LLC set with **Prime+Probe**



Background: LLC Prime+Probe Attack

Eviction set ⇒ Monitor memory accesses to an LLC set with **Prime+Probe**

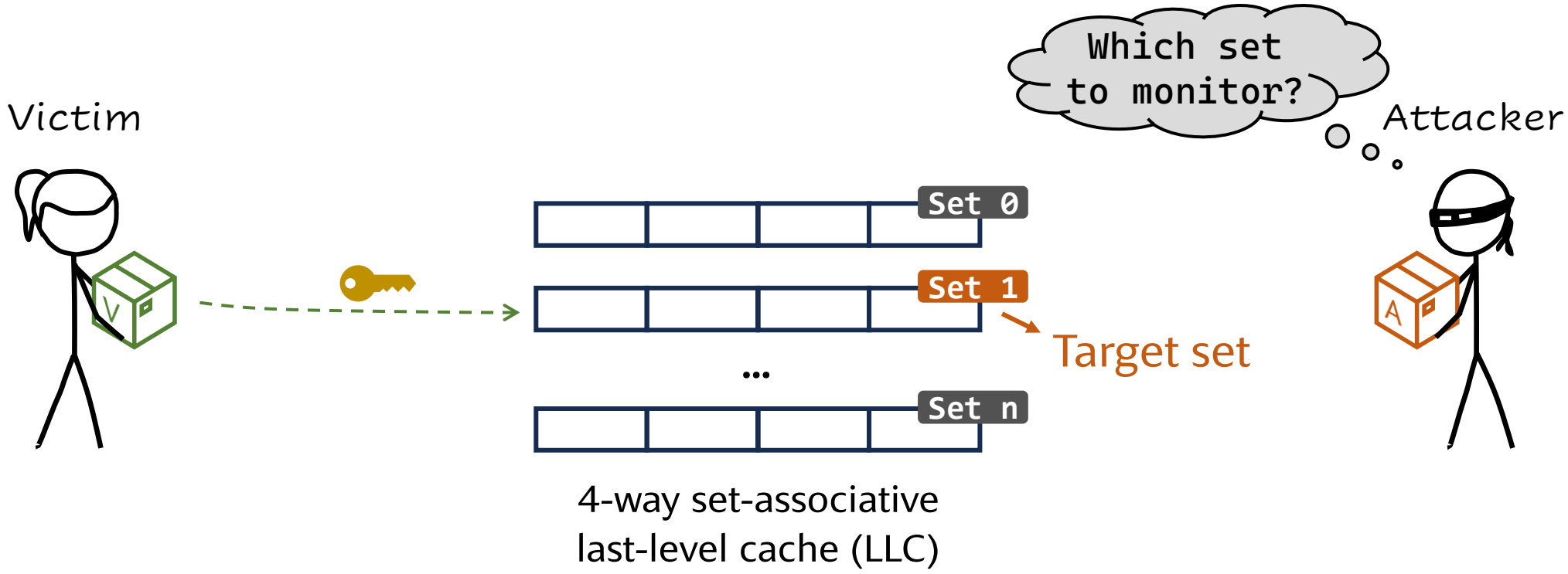
Takeaway



4-way set-associative
last-level cache (LLC)

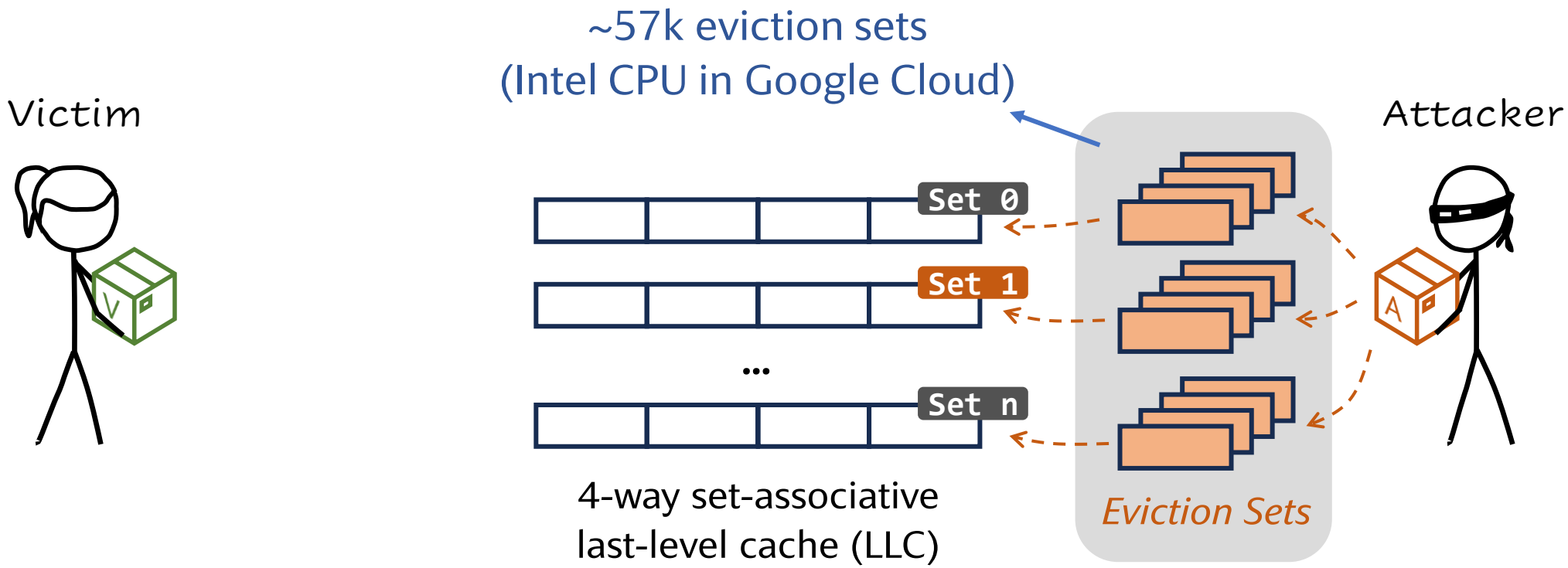
An Unprivileged Attacker Does Not Know the Target Set

Target set: An LLC set accessed by the victim in a secret-dependent manner



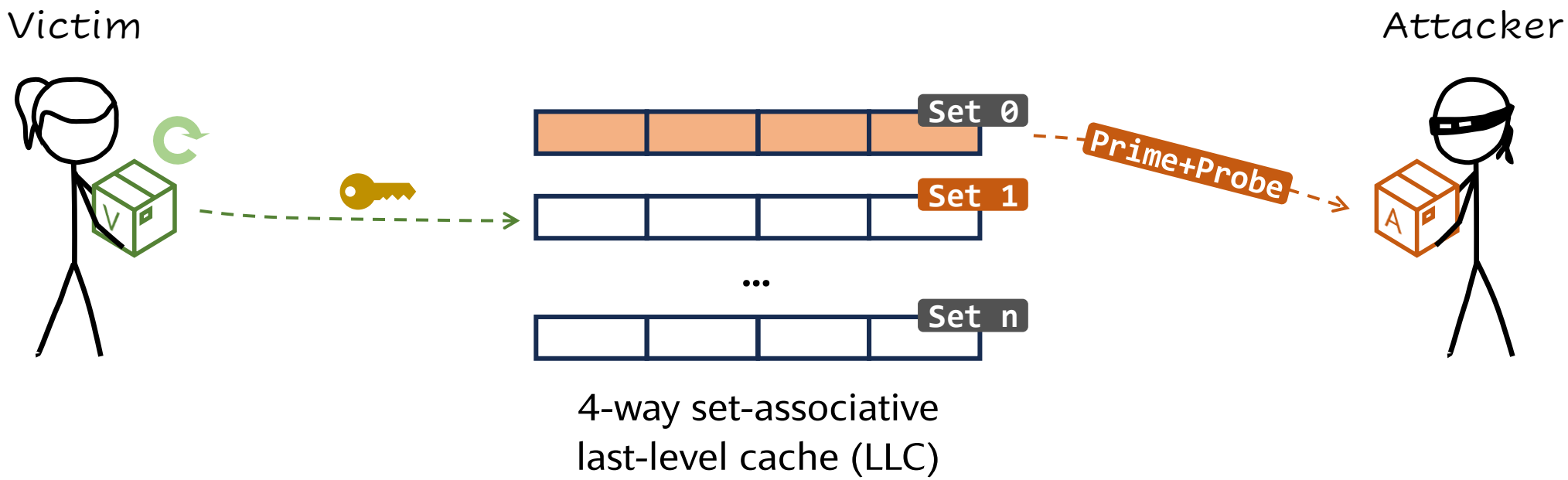
Step 2.1: Build Many Eviction Sets

Attacker needs an eviction set for every LLC set in the system



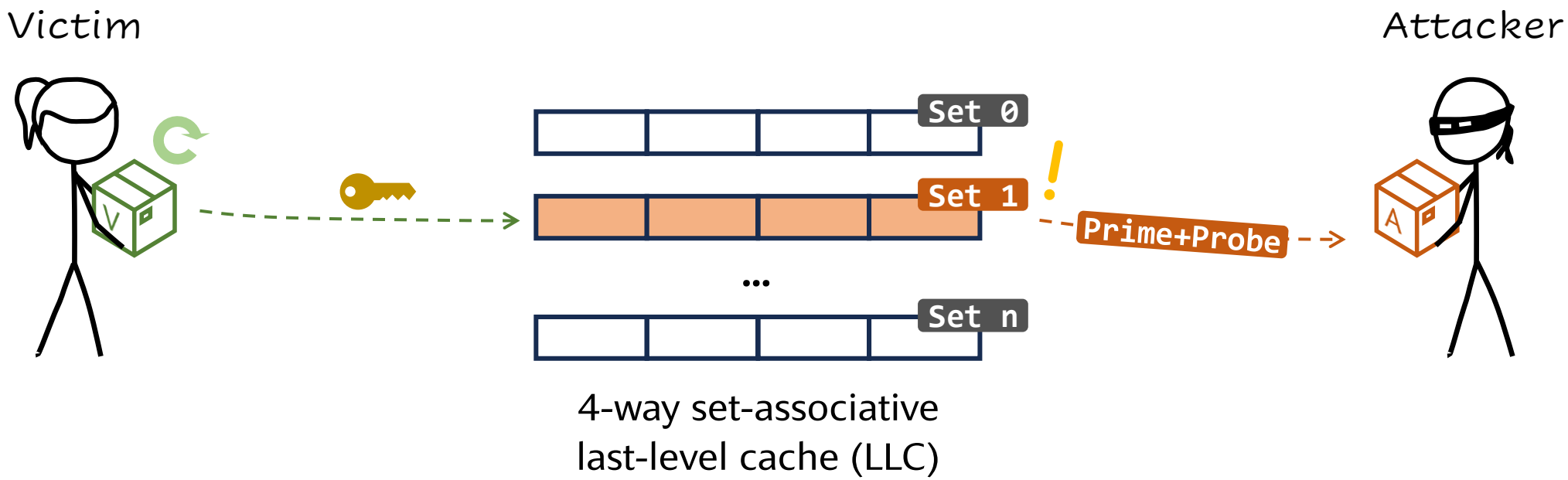
Step 2.2: Identify Target LLC Set to Monitor

Attacker collects an access trace from *each* LLC set
⇒ Checks whether the access trace matches victim's access behavior



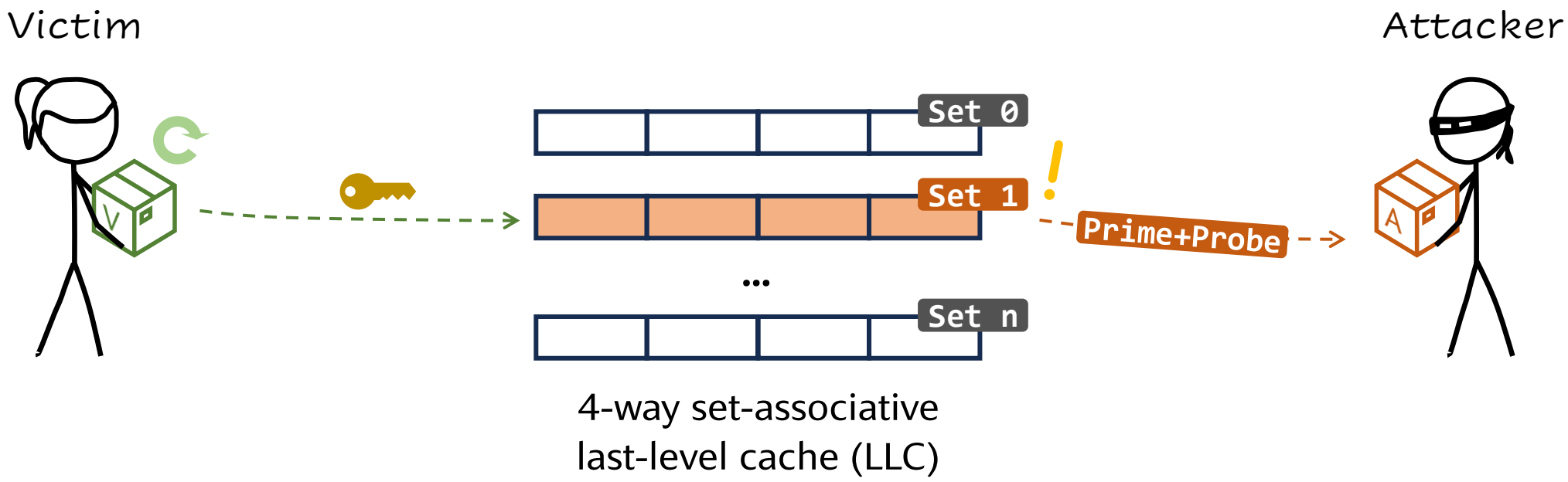
Step 2.2: Identify Target LLC Set to Monitor

Attacker collects an access trace from *each* LLC set
⇒ Checks whether the access trace matches victim's access behavior

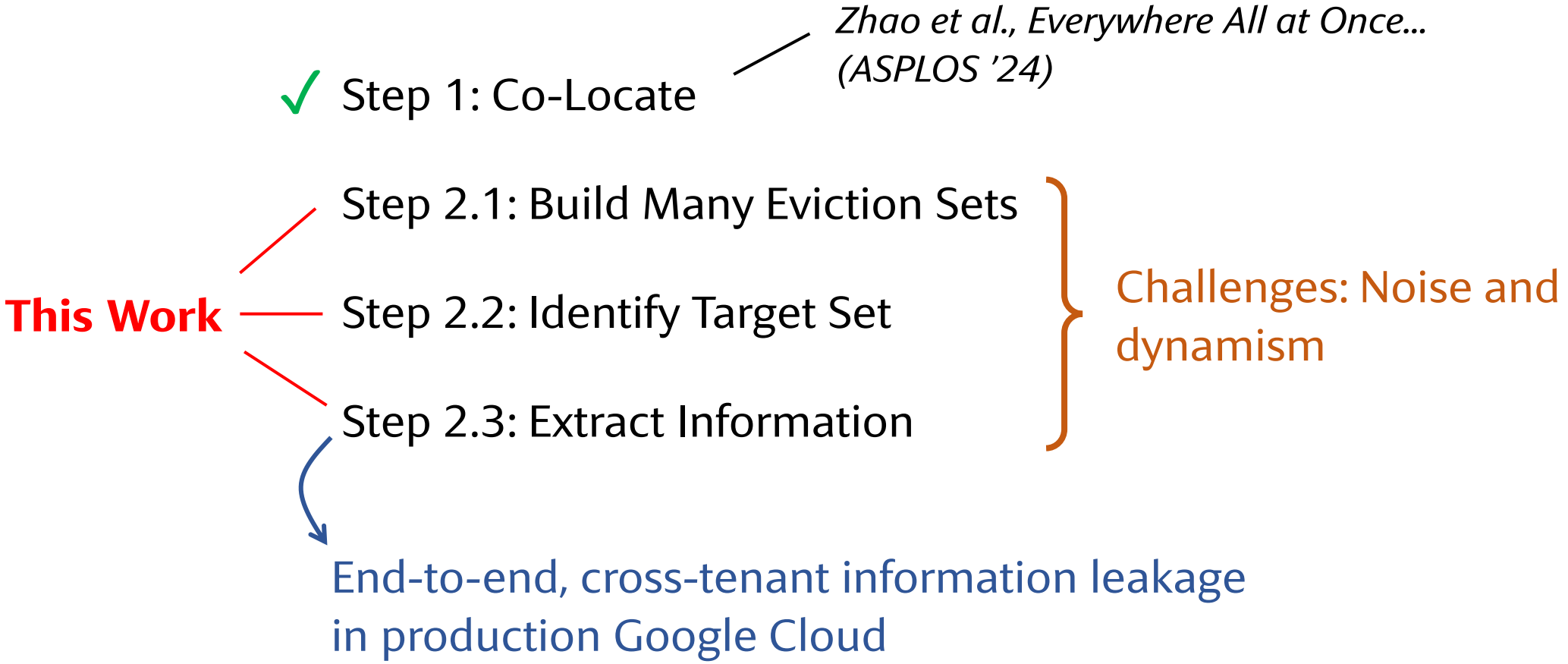


Step 2.3: Extract Information from the Victim

Attacker monitors the target set and extracts the sensitive information



Attack Roadmap



Attack Roadmap

✓ Step 1: Co-Locate

Zhao et al., Everywhere All at Once... (ASPLOS '24)

Step 2.1: Build Many Eviction Sets

Step 2.2: Identify Target Set

Step 2.3: Extract Information

Challenges: Noise and dynamism

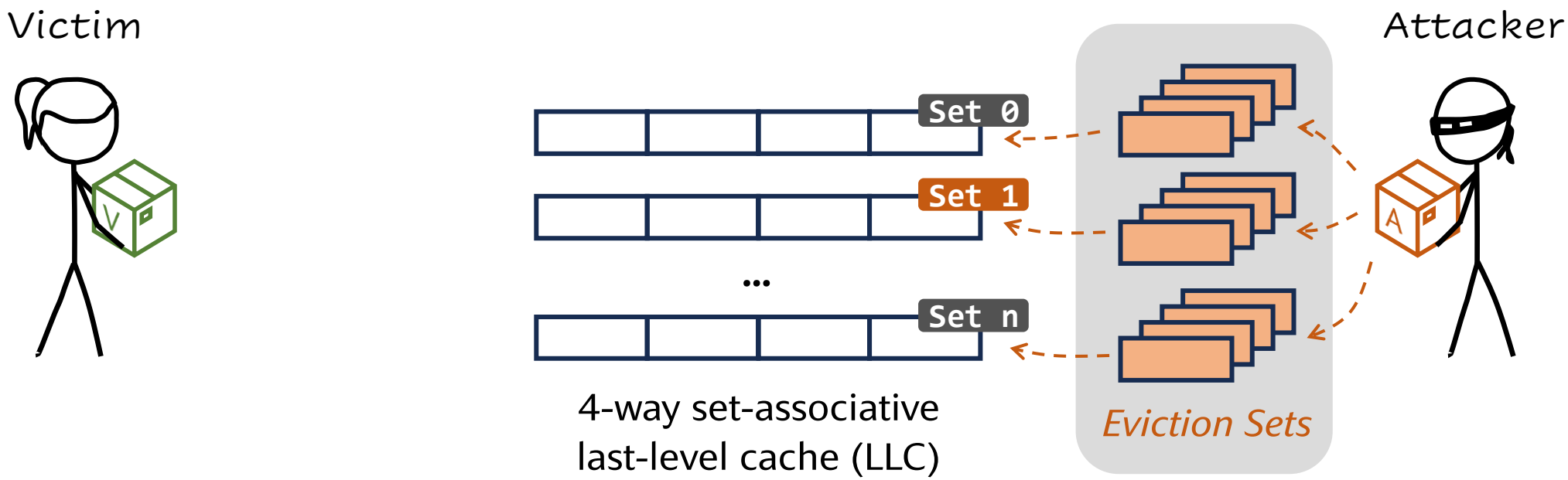
End-to-end, cross-tenant information leakage in production Google Cloud

Our Contribution: Fast Eviction Set Construction

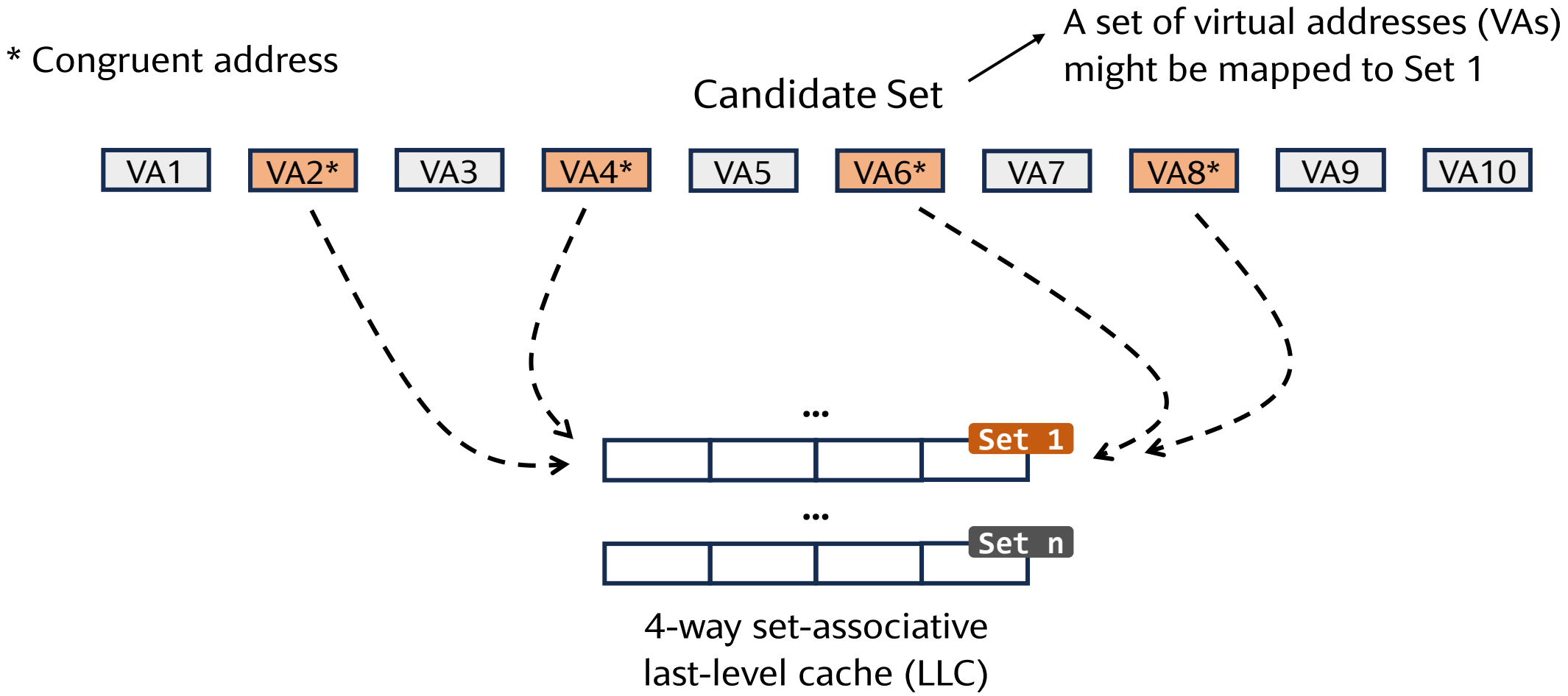
Contribution: Fast eviction set construction in ***~2.4 minutes*** Google Cloud

Existing approaches: would take >10 hours to complete due to noise

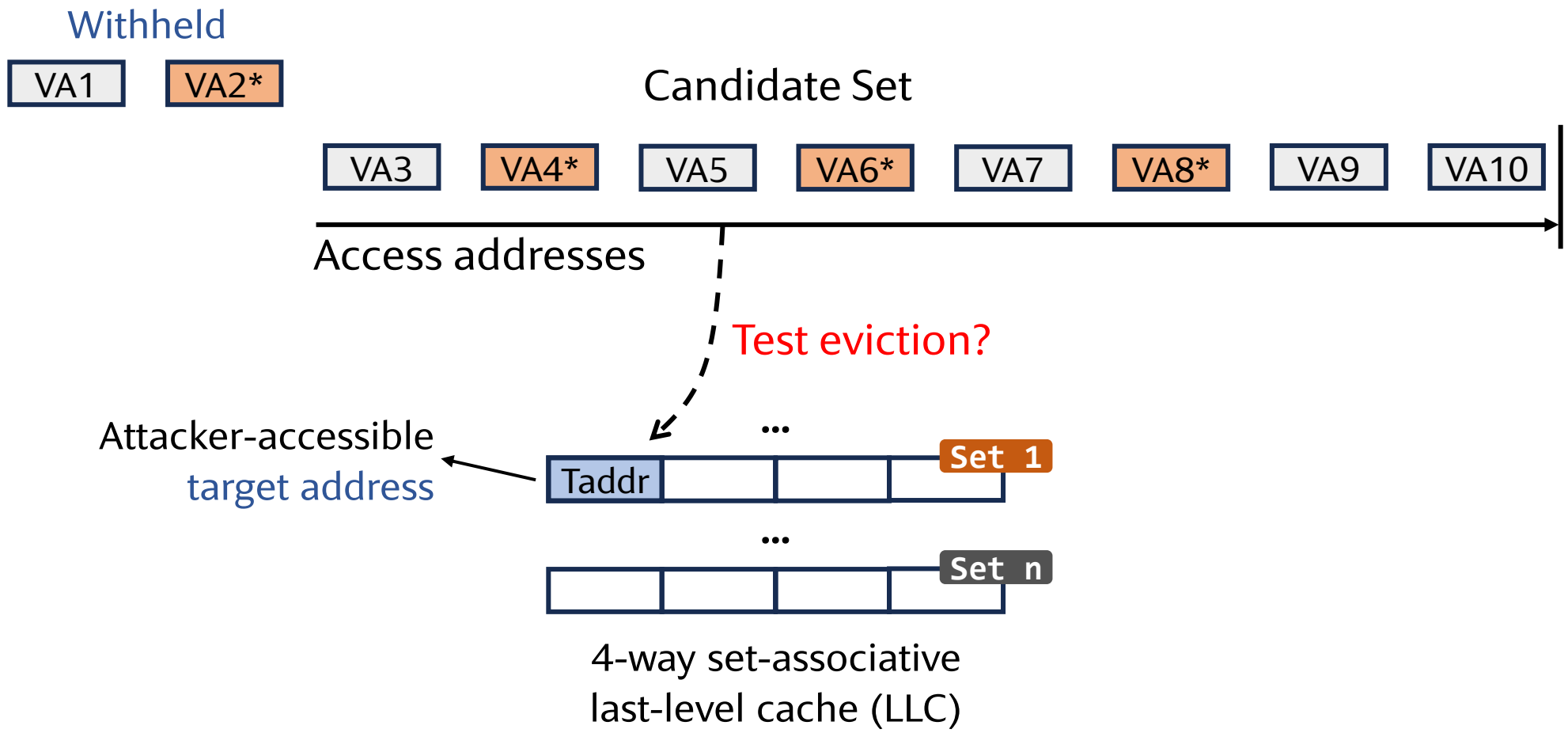
↳ Exceeds execution time limits!



Background: Constructing an Eviction Set

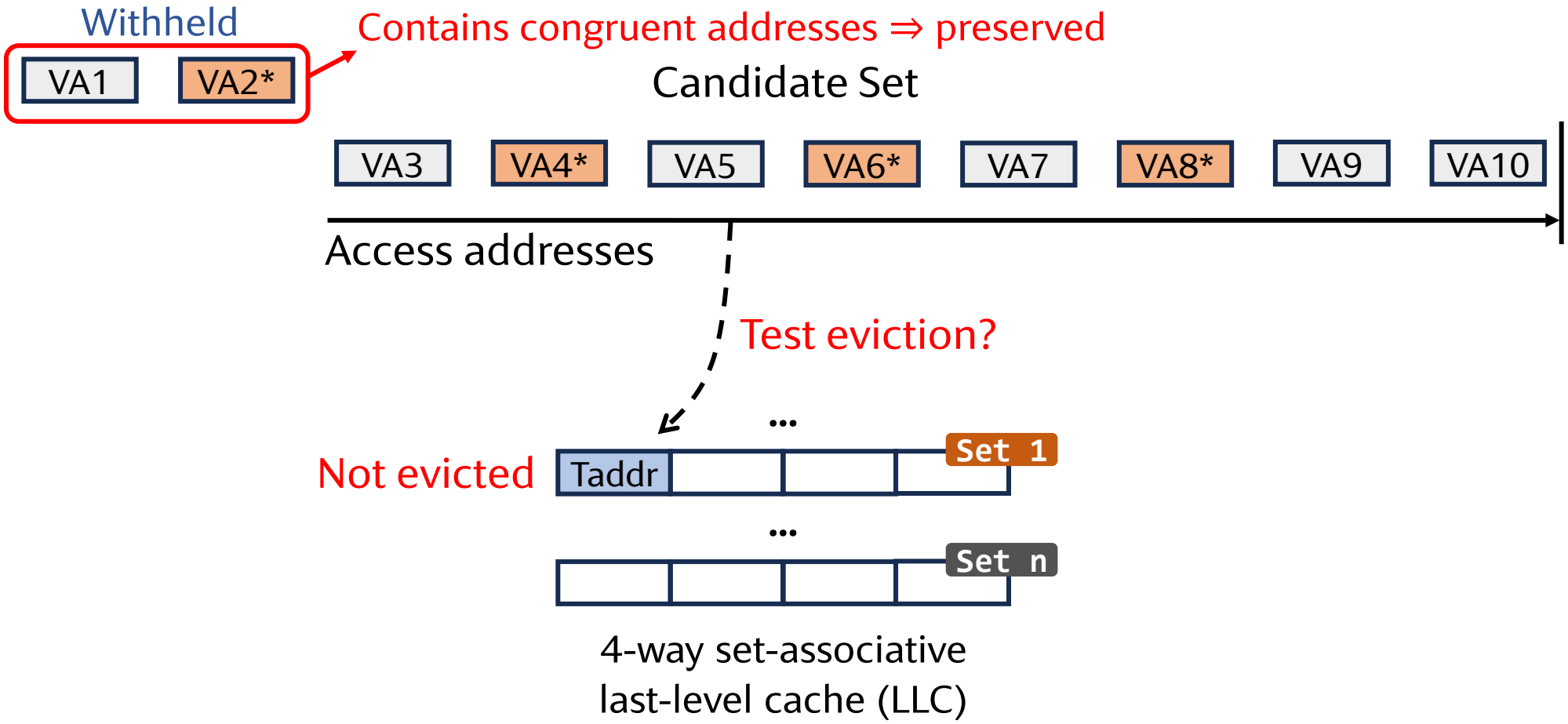


Background: Pruning the Candidate Set (Group Testing[1,2])



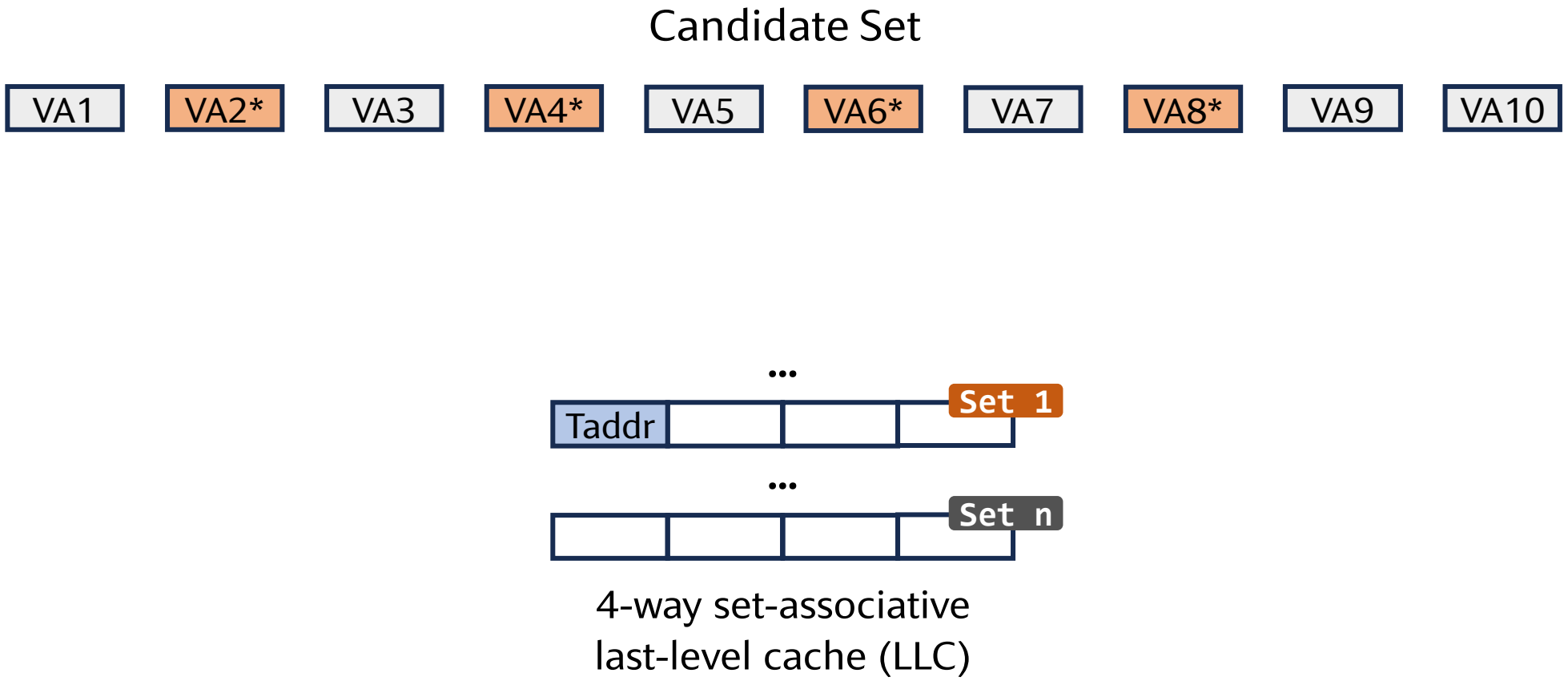
[1]Vila et al., Theory and Practice of Finding Eviction Sets (S&P 2019)
[2]Qureshi et al., New Attacks and Defense for Encrypted-Address Cache (ISCA 2019)

Background: Pruning the Candidate Set (Group Testing[1,2])



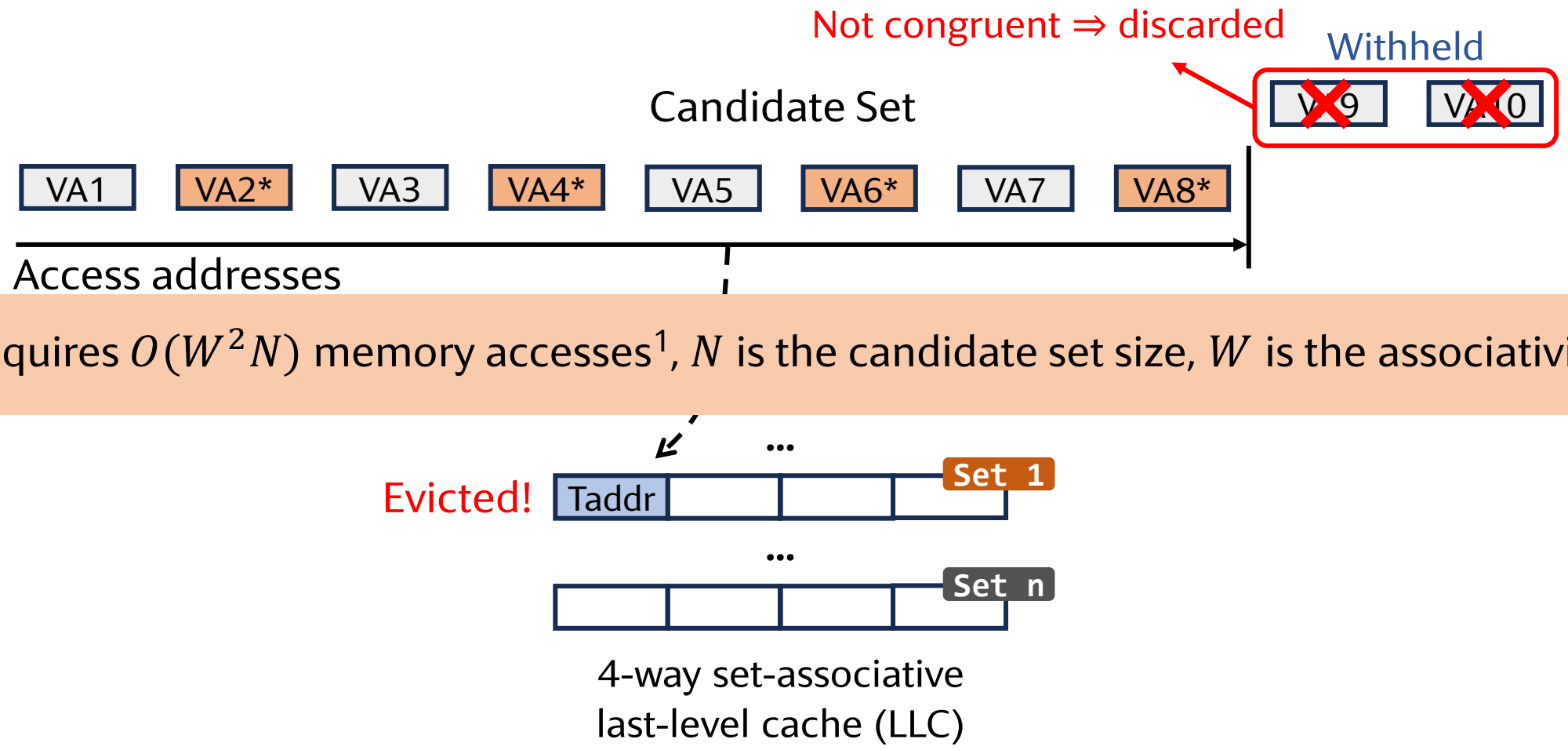
[1]Vila et al., Theory and Practice of Finding Eviction Sets (S&P 2019)
[2]Qureshi et al., New Attacks and Defense for Encrypted-Address Cache (ISCA 2019)

Background: Pruning the Candidate Set (Group Testing[1,2])



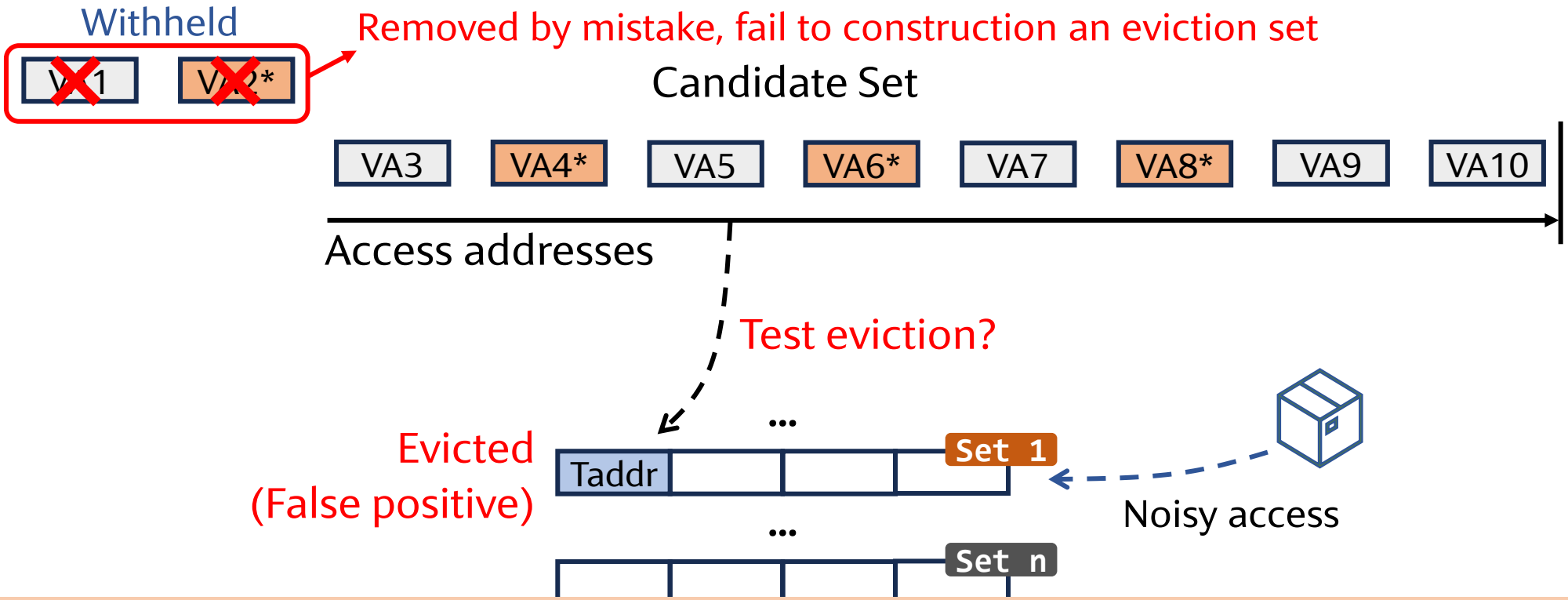
[1]Vila et al., Theory and Practice of Finding Eviction Sets (S&P 2019)
[2]Qureshi et al., New Attacks and Defense for Encrypted-Address Cache (ISCA 2019)

Background: Pruning the Candidate Set (Group Testing[1,2])



[1]Vila et al., Theory and Practice of Finding Eviction Sets (S&P 2019)
[2]Qureshi et al., New Attacks and Defense for Encrypted-Address Cache (ISCA 2019)

This Work: Test Eviction is Susceptible to Noise



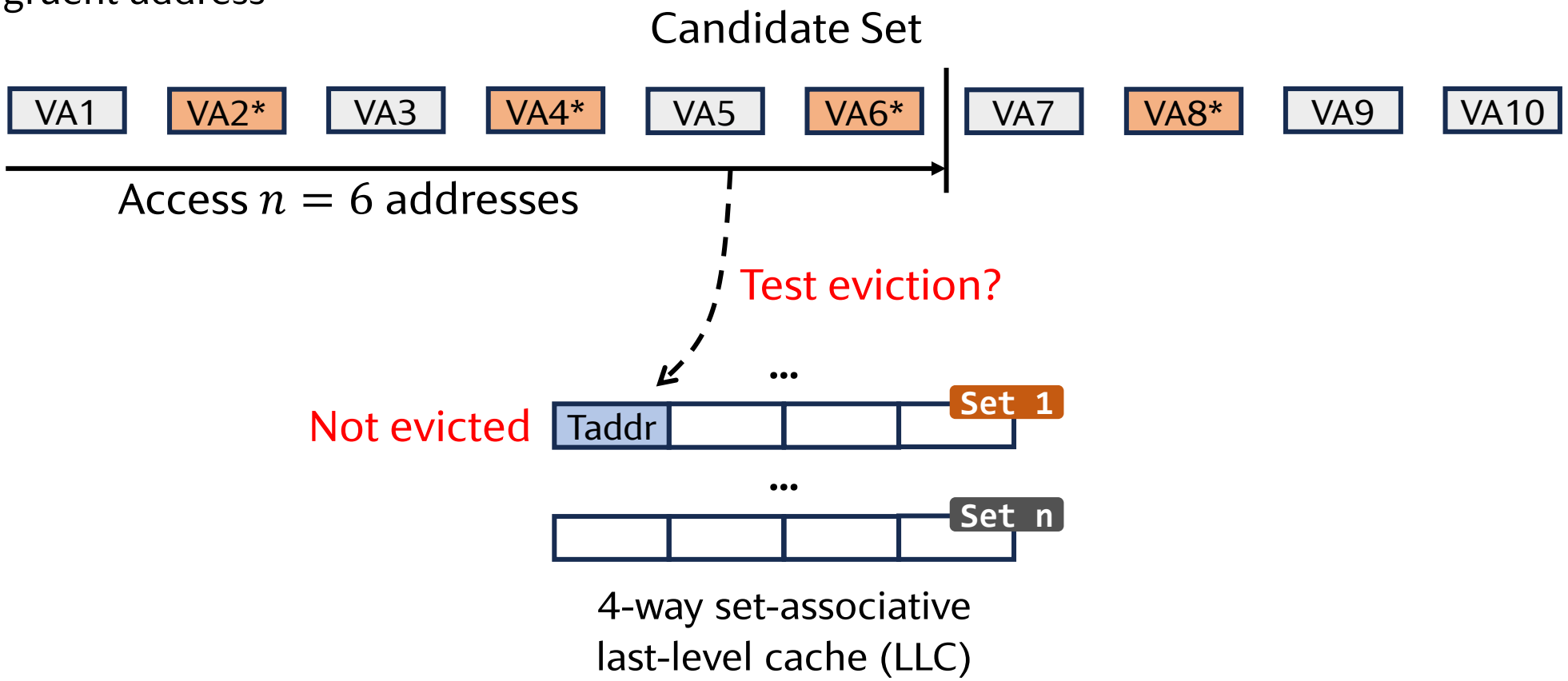
Prime+Scope[1] is similarly susceptible to noise

Our paper provides more detailed quantitative analyses of both algorithms

[1]Purnal et al., Prime+Scope: Overcoming the Observer Effect for High-Precision Cache Contention Attacks (CCS 2021)

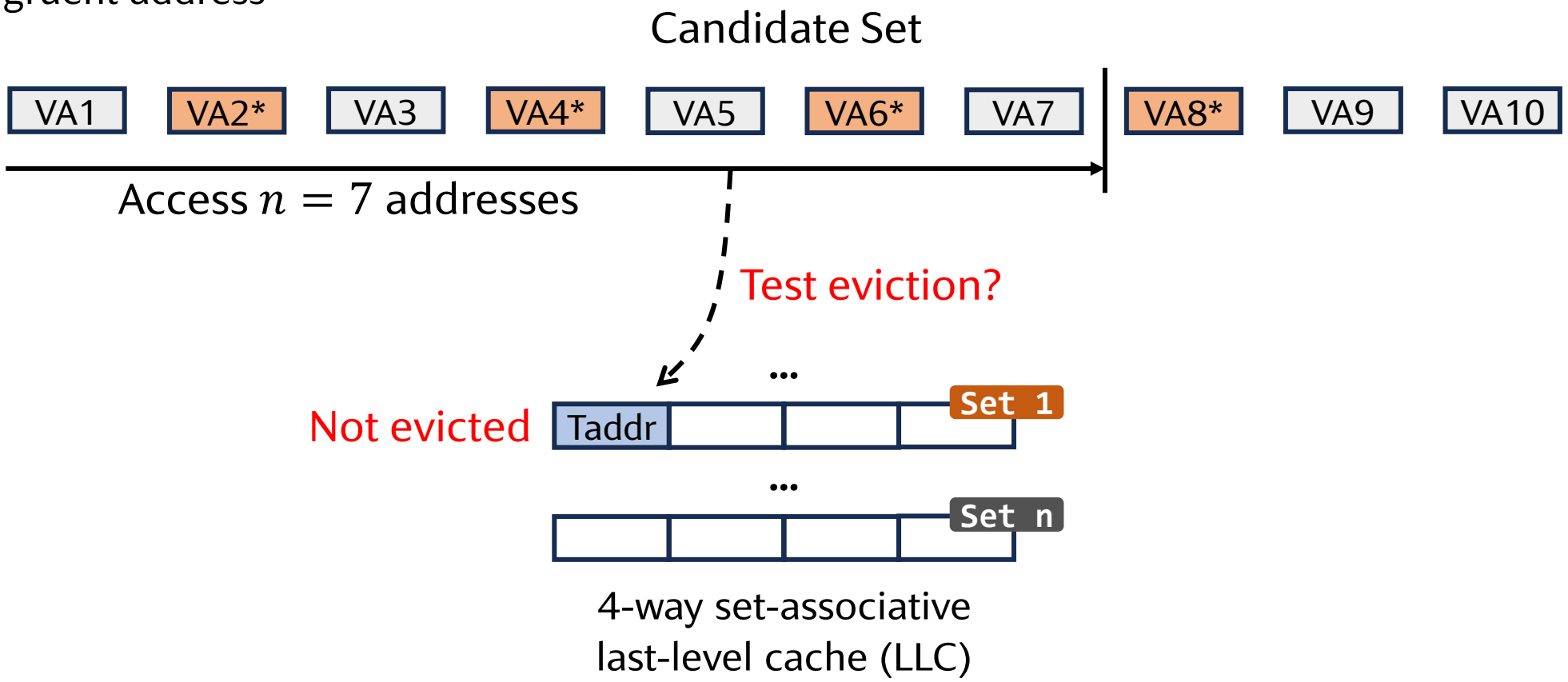
Intuition of the New Algorithm

* Congruent address

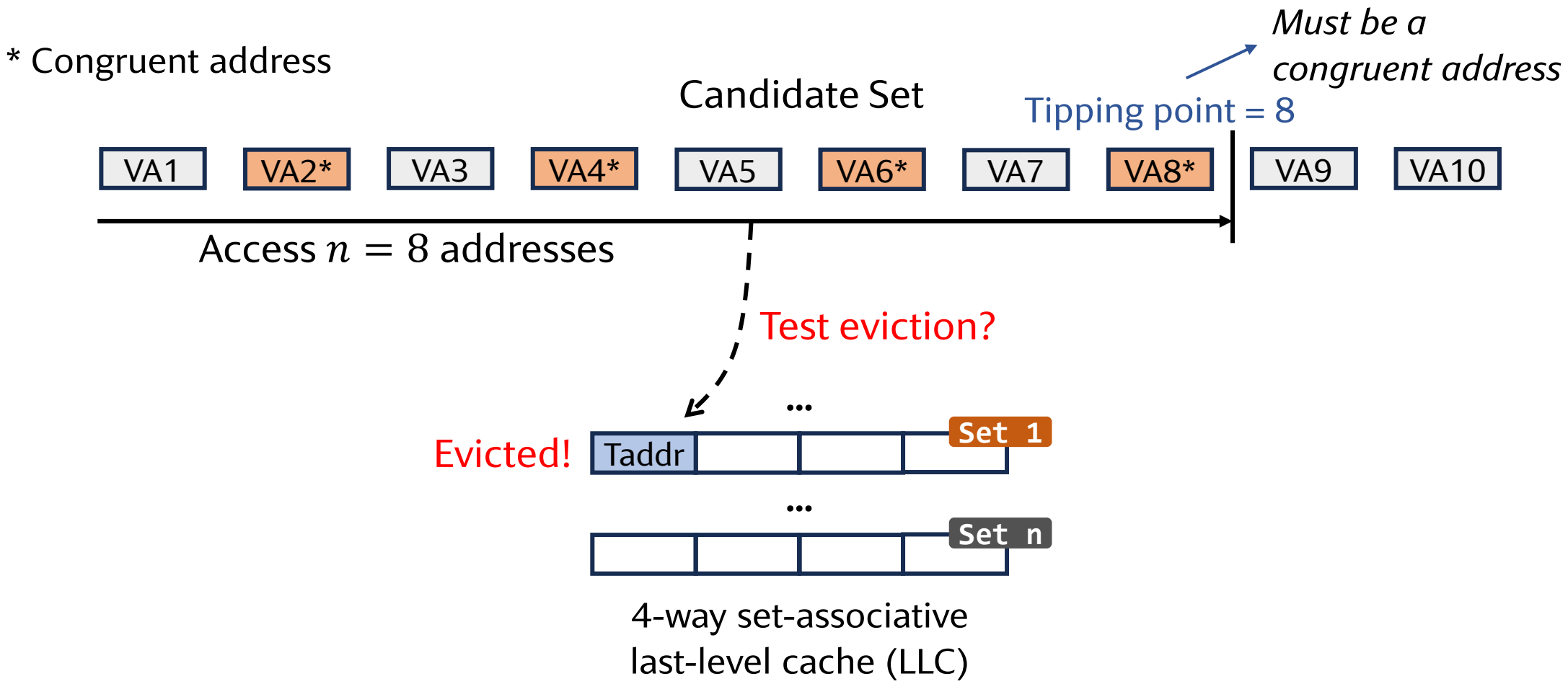


Intuition of the New Algorithm

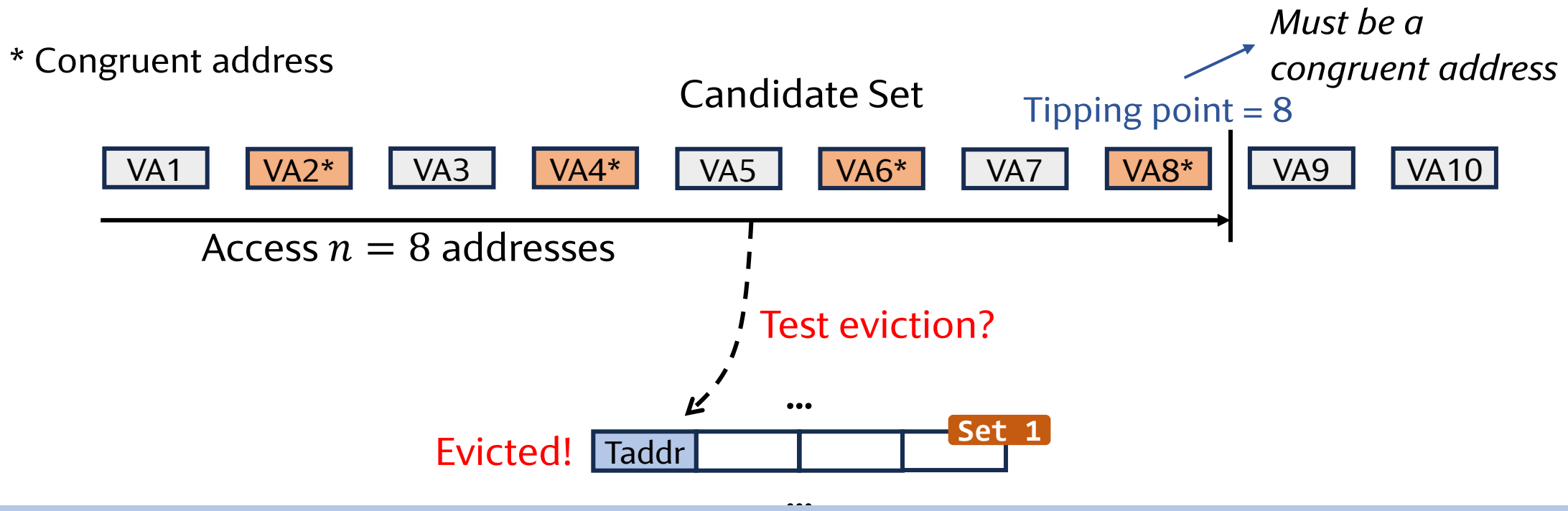
* Congruent address



Intuition of the New Algorithm



Intuition of the New Algorithm

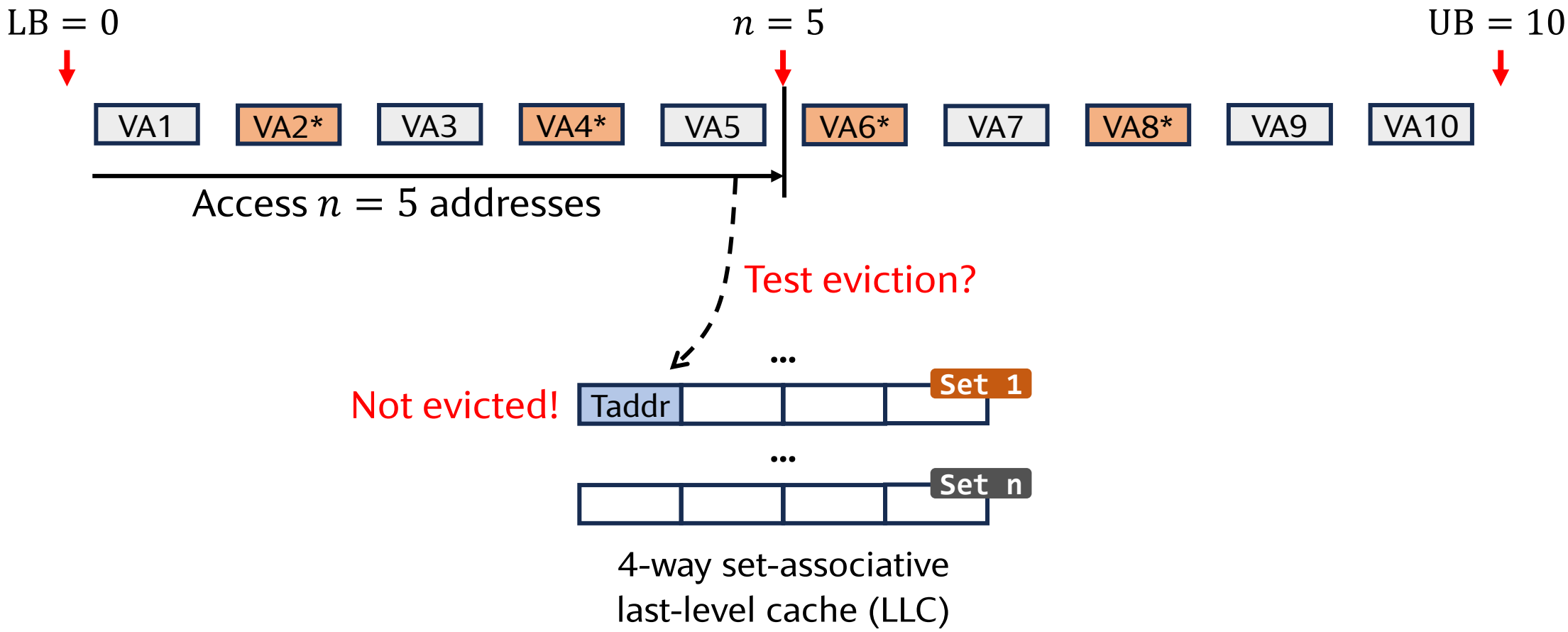


Idea: Identifying congruent addresses by finding tipping points

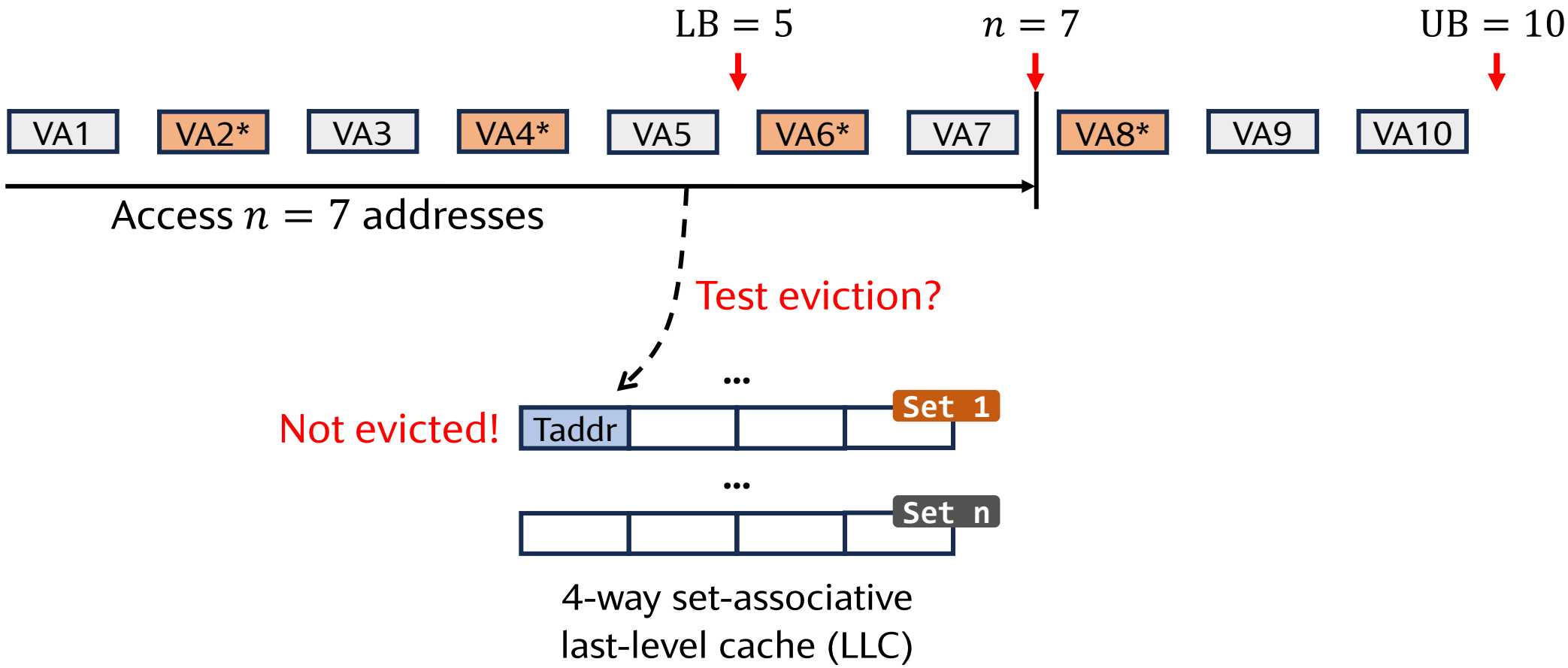
Insight: Can evict ⇒ $n \geq$ tipping point; Cannot evict ⇒ $n <$ tipping point

⇒ Speed up the process with **binary search**

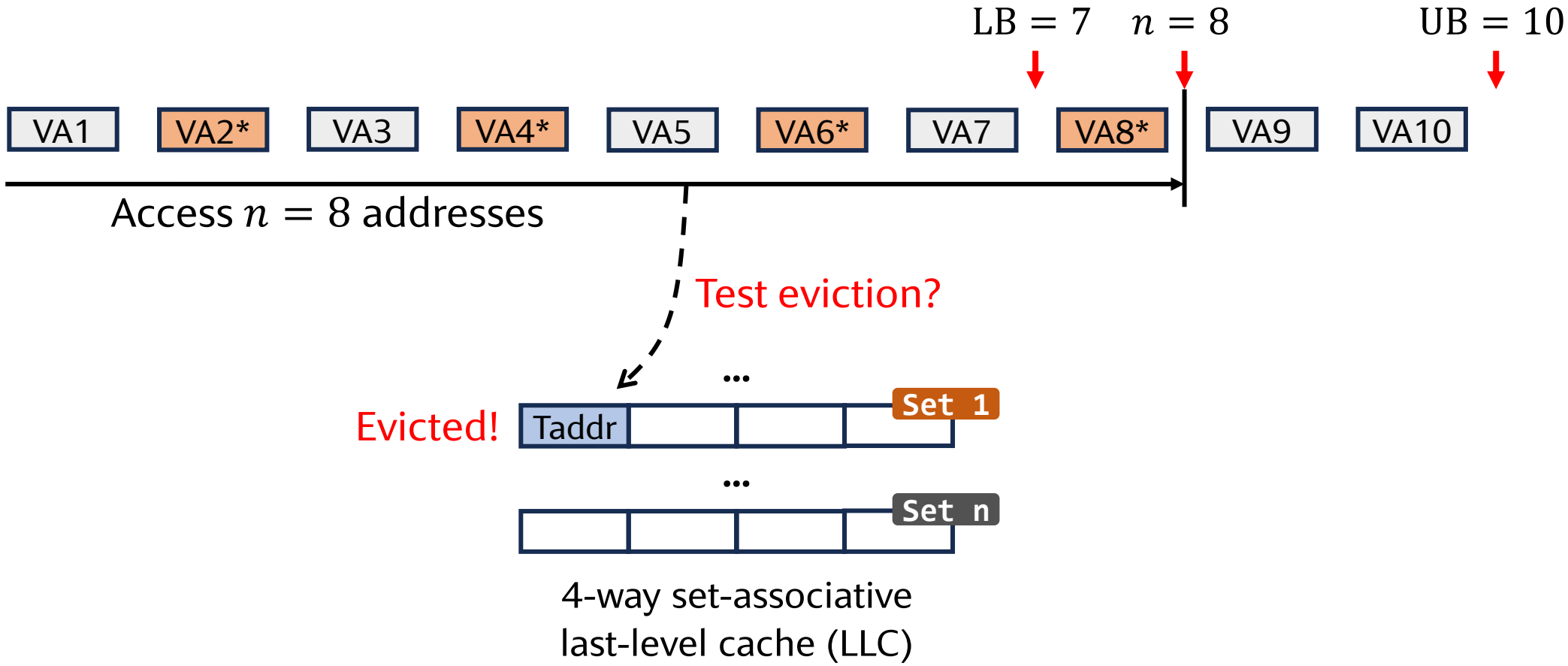
New Technique 1: A Binary Search-Based Algorithm



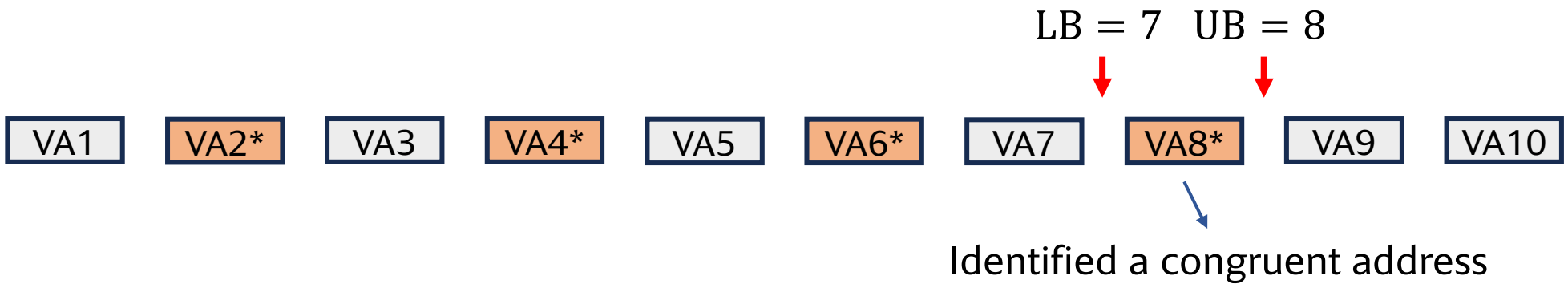
New Technique 1: A Binary Search-Based Algorithm

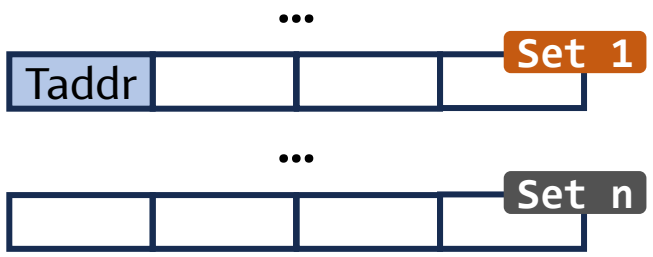


New Technique 1: A Binary Search-Based Algorithm



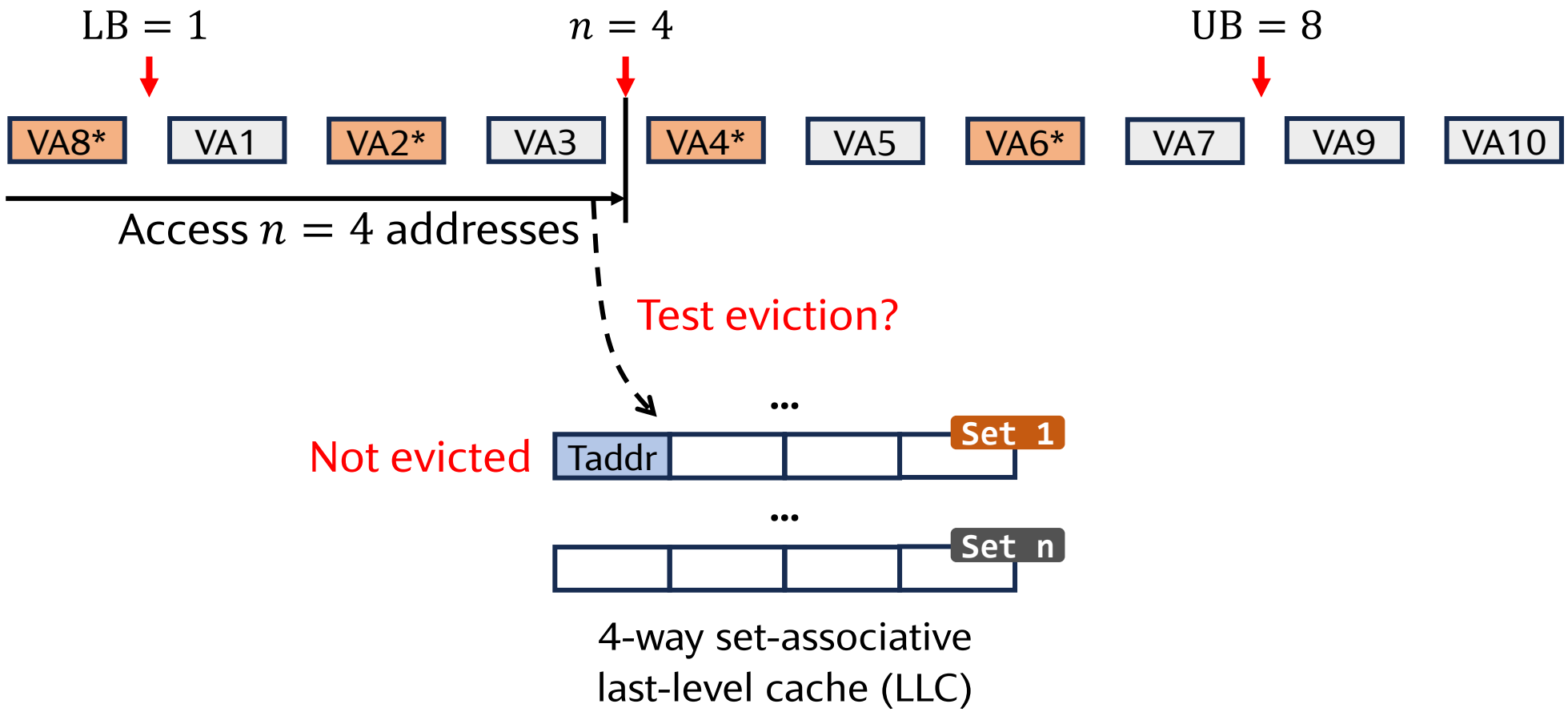
New Technique 1: A Binary Search-Based Algorithm



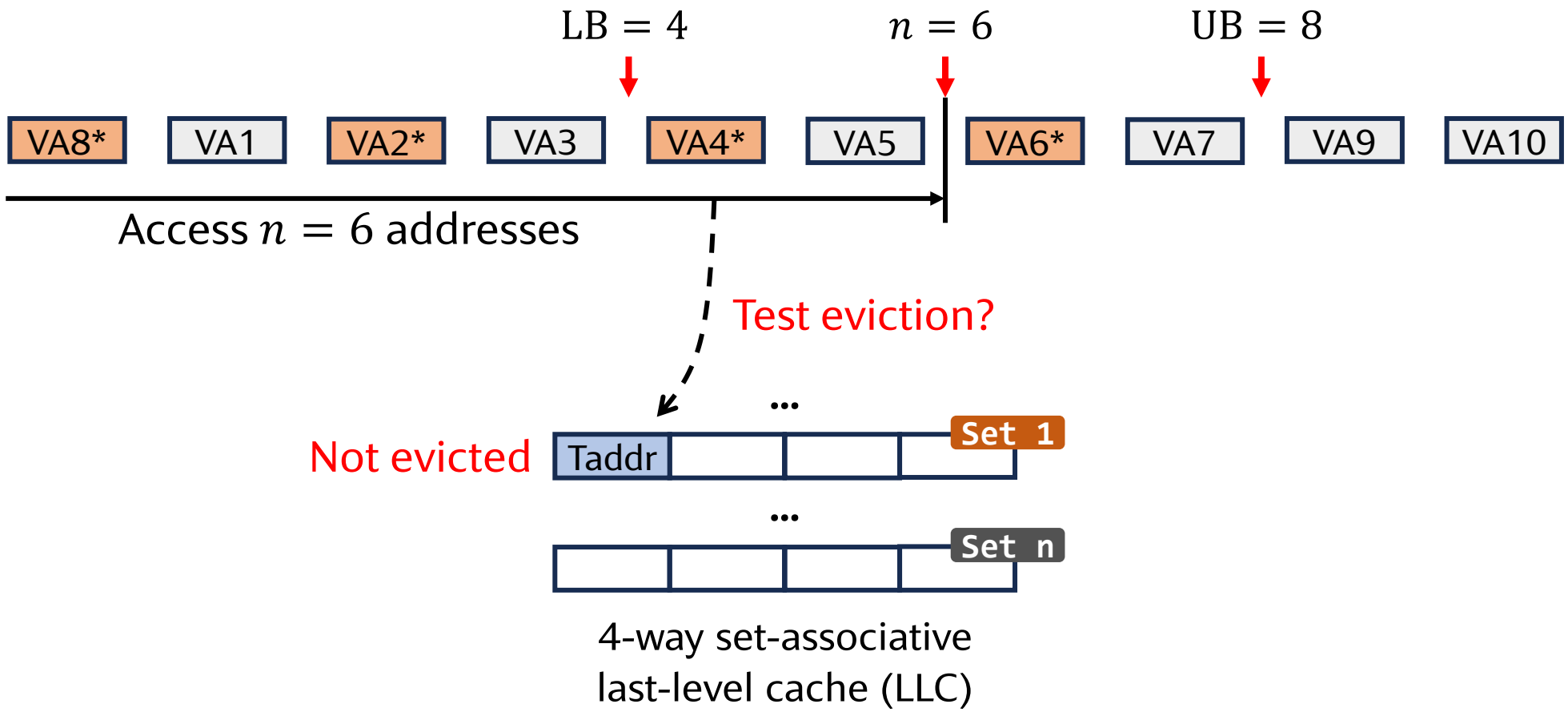


4-way set-associative
last-level cache (LLC)

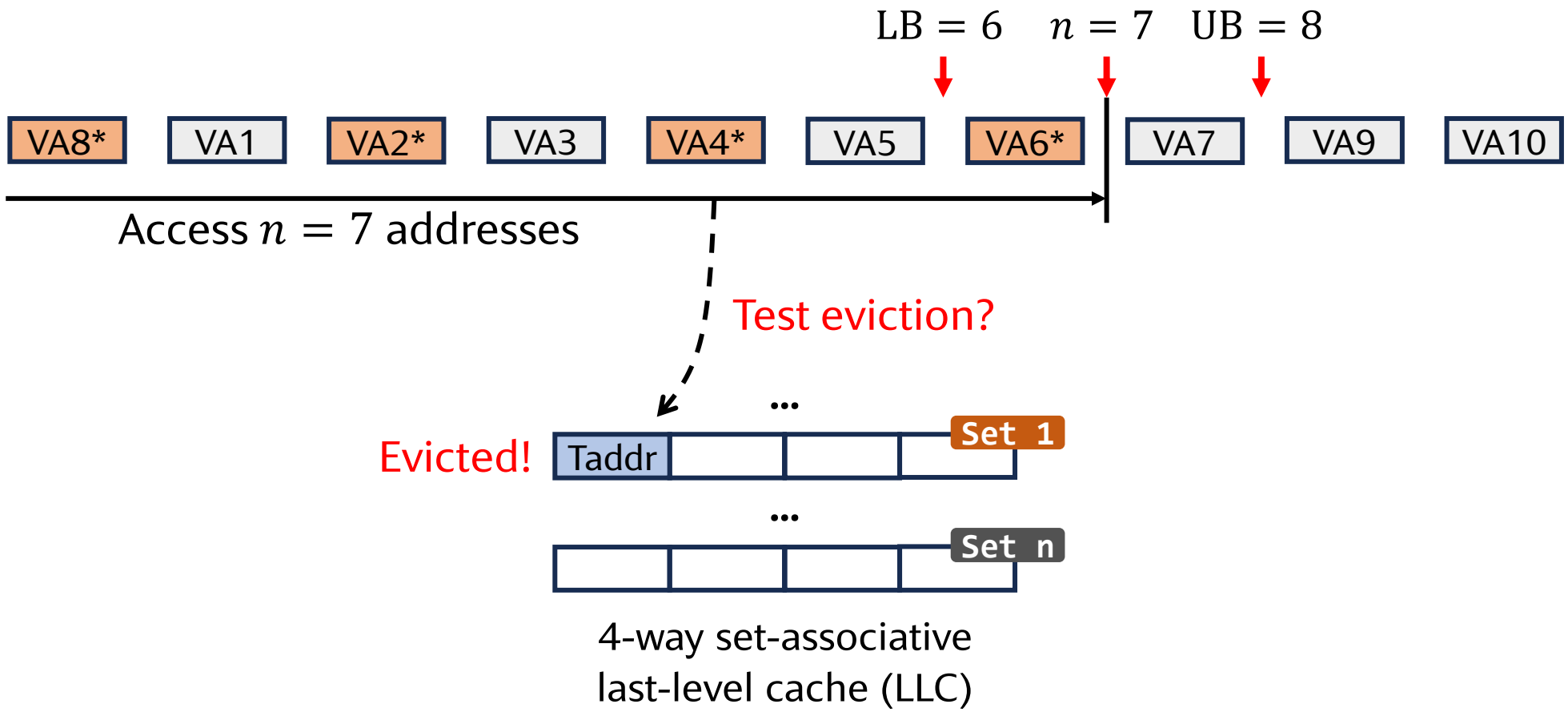
New Technique 1: A Binary Search-Based Algorithm



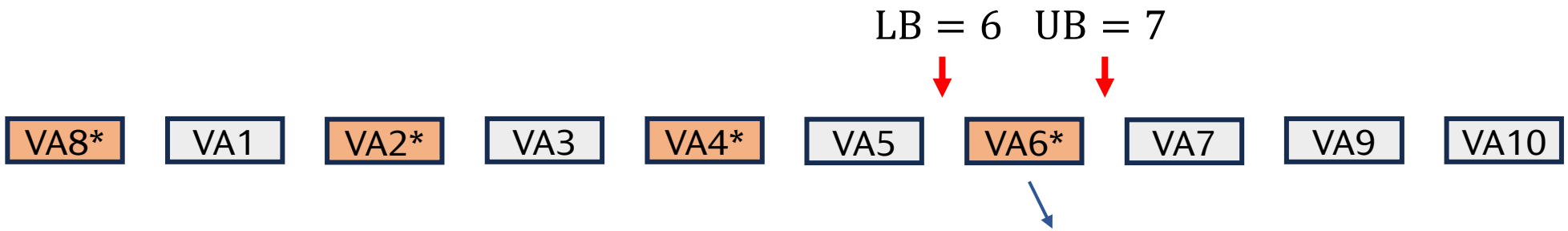
New Technique 1: A Binary Search-Based Algorithm

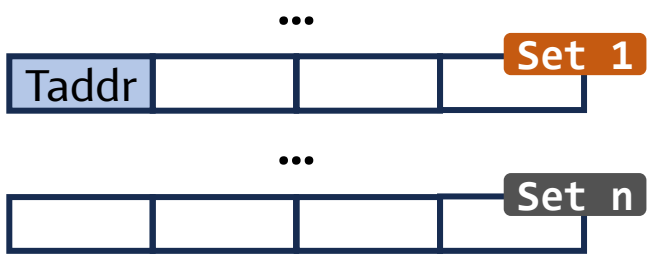


New Technique 1: A Binary Search-Based Algorithm



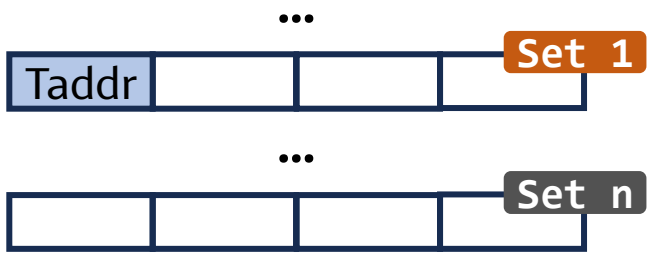
New Technique 1: A Binary Search-Based Algorithm





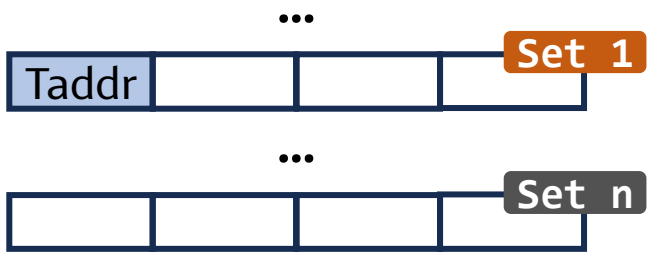
4-way set-associative
last-level cache (LLC)

New Technique 1: A Binary Search-Based Algorithm



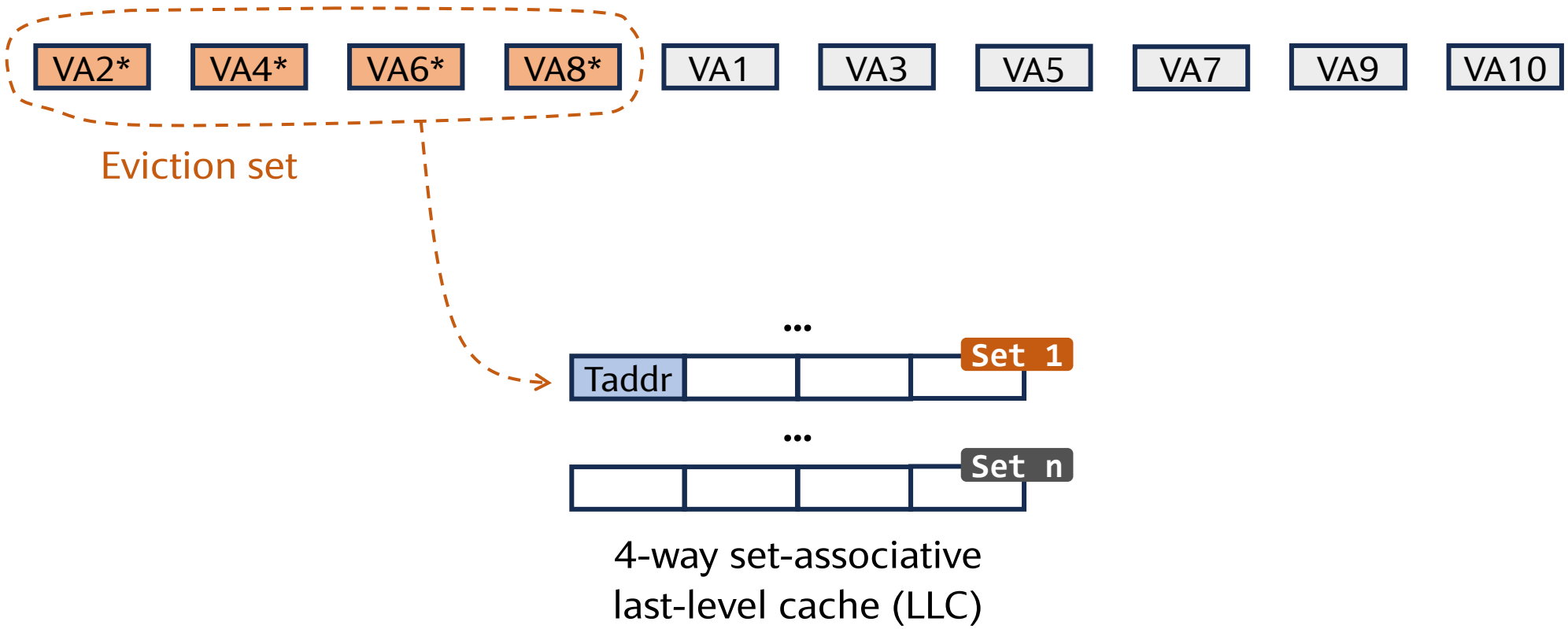
4-way set-associative
last-level cache (LLC)

New Technique 1: A Binary Search-Based Algorithm

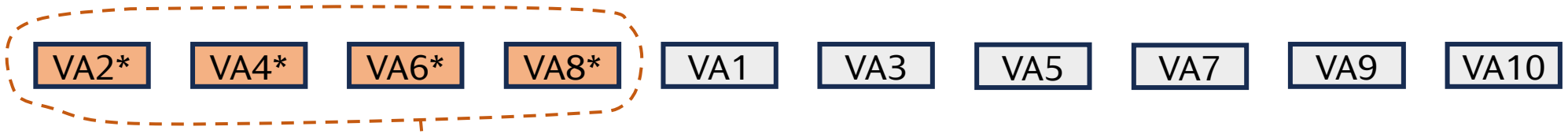


4-way set-associative
last-level cache (LLC)

New Technique 1: A Binary Search-Based Algorithm



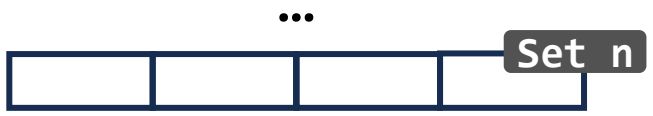
New Technique 1: A Binary Search-Based Algorithm



Intuition: Fewer accesses on caches with higher associativity

Our algorithm requires $\boldsymbol{O(WN \log N)}$ memory accesses

Group Testing requires $\boldsymbol{O(W^2 N)}$ memory accesses



4-way set-associative
last-level cache (LLC)

Intuition: Check out our paper for another optimization that reduces N

Attack Roadmap

Zhao et al., Everywhere All at Once... (ASPLOS '24)

✓ Step 1: Co-Locate

✓ Step 2.1: Build Many Eviction Sets

➡ **Step 2.2: Identify Target Set**

Challenges: Noise and dynamism

Insight: Victim accesses are periodic ⇒ Spectral analysis

End-to-end, cross-tenant information leakage in production Google Cloud

Attack Roadmap

Zhao et al., Everywhere All at Once... (ASPLOS '24)

✓ Step 1: Co-Locate

✓ Step 2.1: Build Many Eviction Sets

✓ Step 2.2: Identify Target Set

➡ **Step 2.3: Extract Information**

Challenges: Noise and dynamism

Insight: Overlap memory accesses of Prime/Probe to exploit memory-level parallelism ⇒ Low Prime/Probe latency ⇒ Good time resolution and noise resilience

Attack Roadmap

✓ Step 1: Co-Locate

Zhao et al., Everywhere All at Once... (ASPLOS '24)

✓ Step 2.1: Build Many Eviction Sets

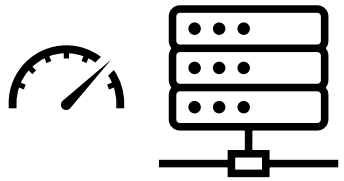
Target victim: A vulnerable ECDSA program from OpenSSL 1.0.1e

Result: Can extract an average of 68% of secret nonce bits (or a median value of 81%)

End-to-end, cross-tenant information leakage in production Google Cloud

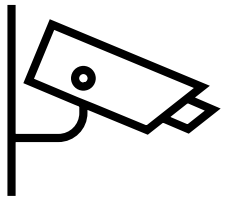
Conclusions

Cross-tenant information leakage with LLC Prime+Probe Google Cloud

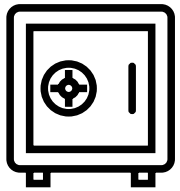


1. Fast LLC Channel Setup

> 10 hours → 2.4 minutes



2. Noise-Resilient Victim Monitoring



3. Information Extraction

+ Google filed a **critical-level bug** to their product team

+ AWS revised their whitepaper on February 15, 2024

GitHub Repo: https://github.com/zzrcxb/LLCFeasible