Design, Automation and Test in Europe Conference
(DATE 2024)

Prime+Reset: Introducing A Novel Cross-World Covert-Channel Through Comprehensive Security Analysis on ARM TrustZone

Yun Chen*
National University of Singapore

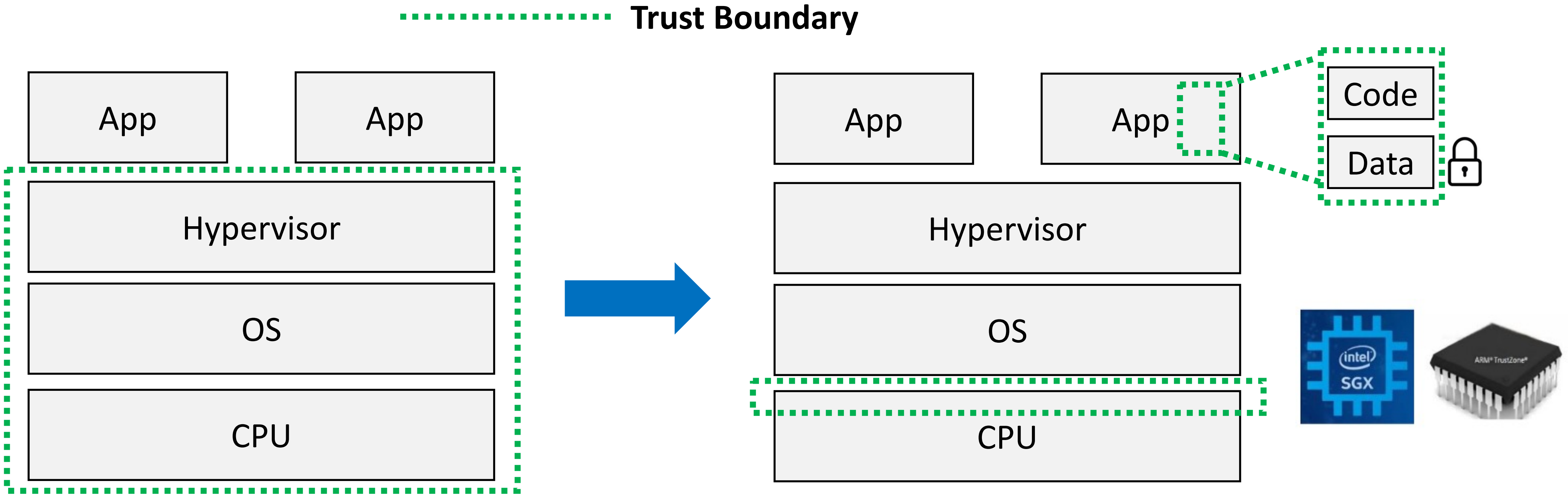
Arash Pashrashid*
National University of Singapore
Huawei Research Center

Yongzheng Wu
Huawei Research Center

Trevor E. Carlson
National University of Singapore

* Equal Contribution

Trusted Execution Environments



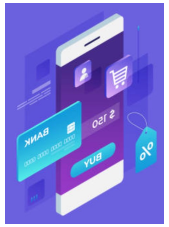
Classic Security Framework

TEE Framework

Applications:

Biometric Authentication

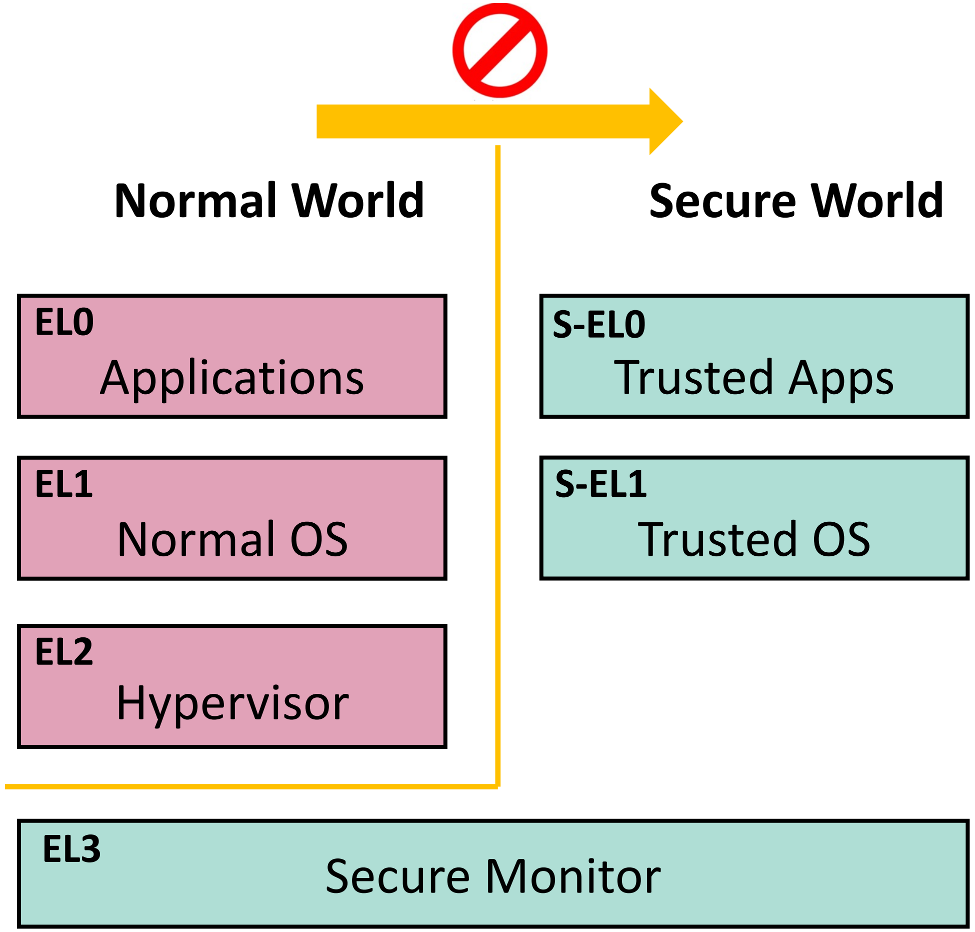
Electronic Payments



Digital Rights Management

NETFLIX

TrustZone TEE Architecture and Its Limitations



Limitations and Problems:

- TEEs are still vulnerable to various microarchitectural side channel attacks
- Lack of comprehensive microarchitectural security analysis of TrustZone

Prior Works:

- Limited scope: Only considering cache and PMUs as a source of leakage

Motivation:

- A comprehensive side/covert-channel vulnerability analysis on TrustZone
- Detecting a new leakage source

Leakage Analysis Criteria

Question: Are components of an OoO processor isolated between the Secure World and the Normal World?

Requirements for a successful side/covert channel:

RQ1

Shared resources among different execution domains that can create resource contention and execution footprints (e.g., cache)

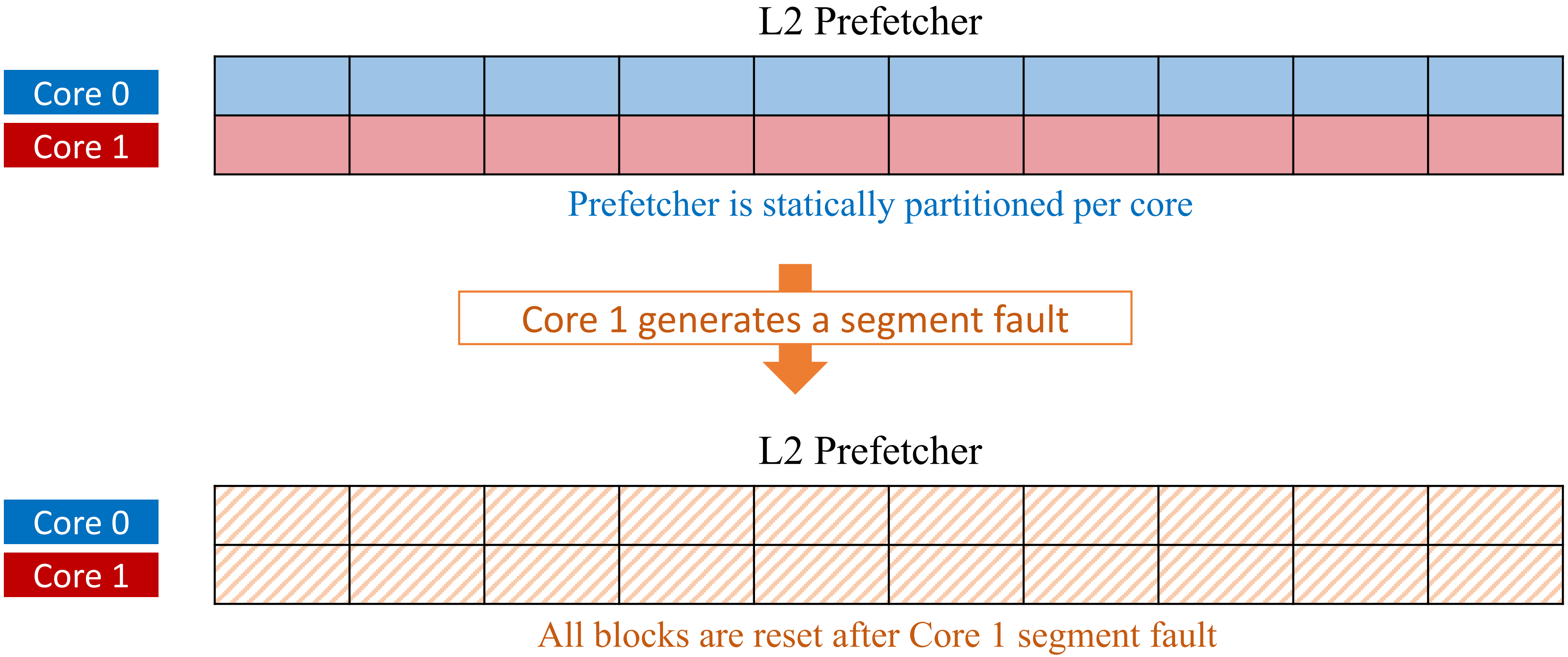
RQ2

Microarchitectural events that can create distinguishable and leaking events during execution (e.g., out-of-order execution)

Analysis result:

Component	Issue Ports	TLB	L1 Prefetcher	L1 Cache	L2 Prefetcher
Leakage?	No	No	No	No	Yes
Reason	Pipeline flushing	Pipeline flushing	Entry invalidating	Entry invalidating	Resetting between worlds

Our Observation from L2 Prefetcher



L2 Prefetcher

Core 0

Core 1

Prefetcher is statically partitioned per core

Core 1 generates a segment fault

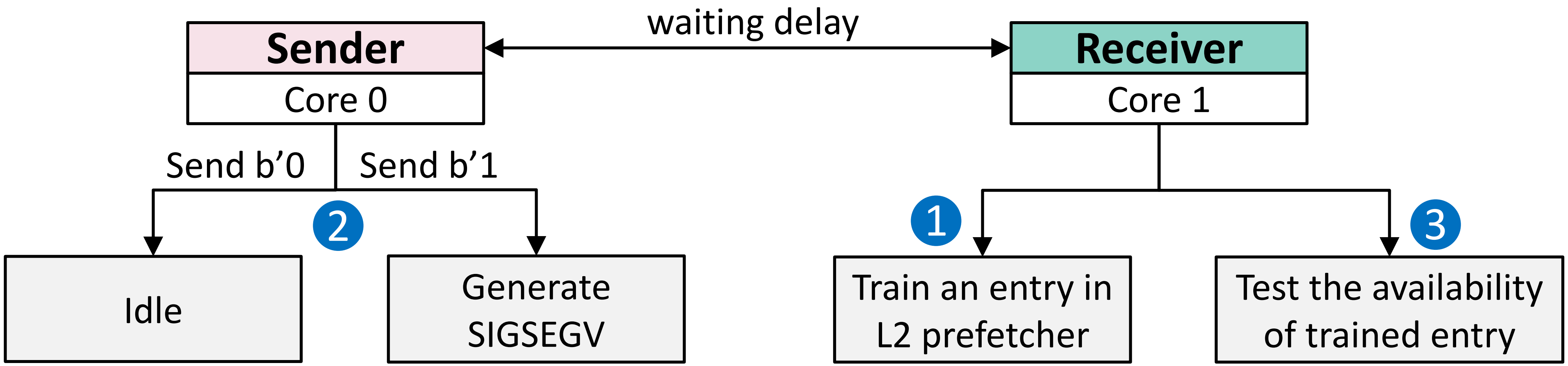
L2 Prefetcher

Core 0

Core 1

All blocks are reset after Core 1 segment fault

Prime+Reset: Covert-Channel using the L2 Prefetcher



waiting delay

Sender

Core 0

Send b’0

Send b’1

2

Idle

Generate SIGSEGV

Receiver

Core 1

1

Train an entry in L2 prefetcher

3

Test the availability of trained entry

Attack	Error Rate	Max Throughput
Prime+Count[1]	<2%	~ 1 Kib/s
*μ*arch-Count[2]	<2%	12 Kib/s
Prime+Reset (*ours*)	<2%	776 Kib/s