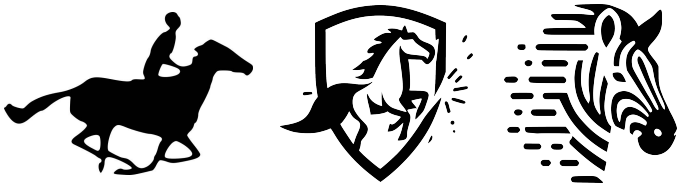


Controlling Controlled-Channel Attacks:

The Art of Blurring Enclave Page-Access Traces in Space and Time

Jo Van Bulck

joint work with collaborators from KUL, Intel, EPFl, Technion, Georgia Tech

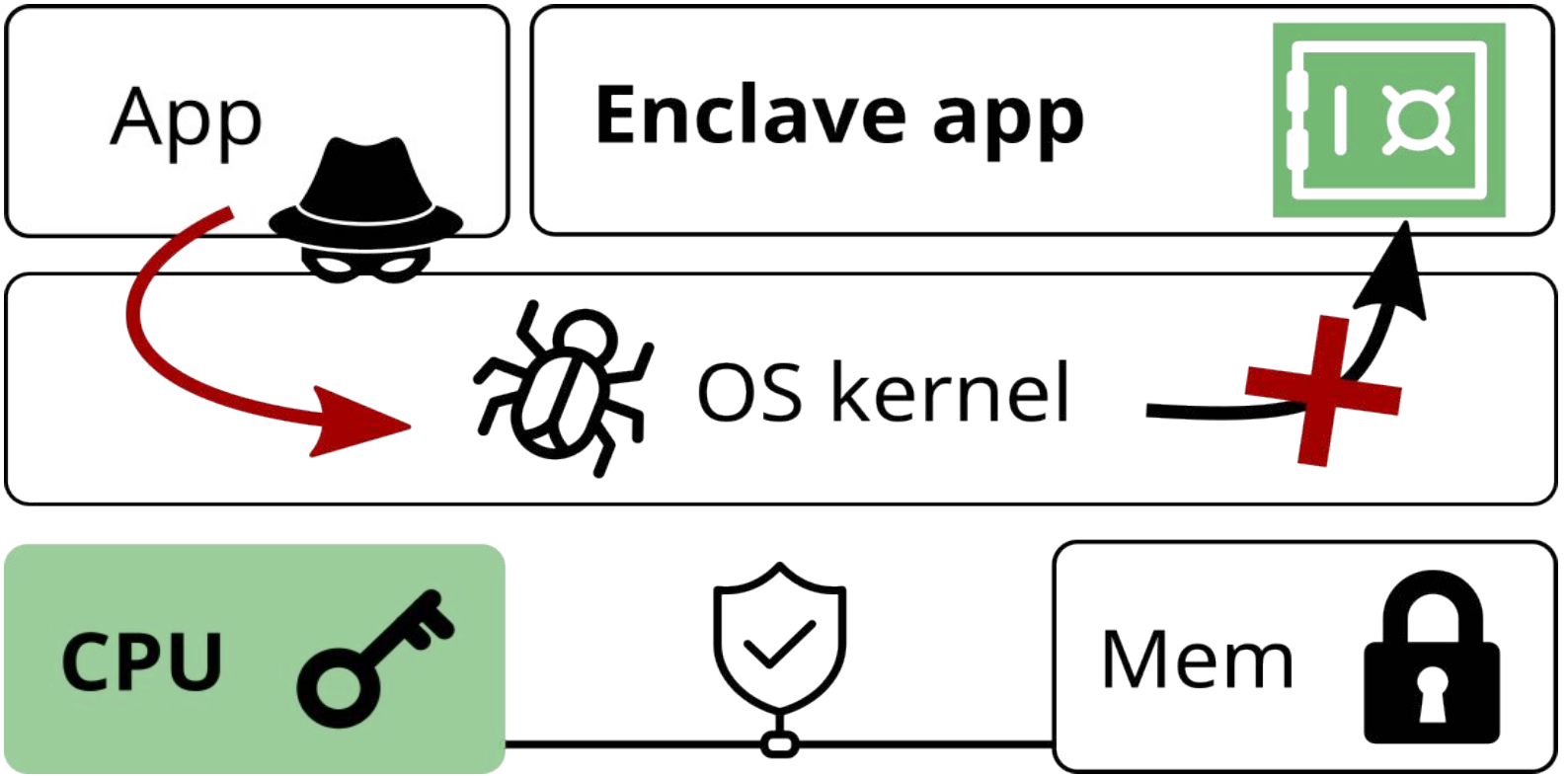
DistriNet, KU Leuven, Belgium jo.vanbulck@cs.kuleuven.be vanbulck.net

Intel Academic Security Conference, Portland – Sept 28, 2026

DistriNet

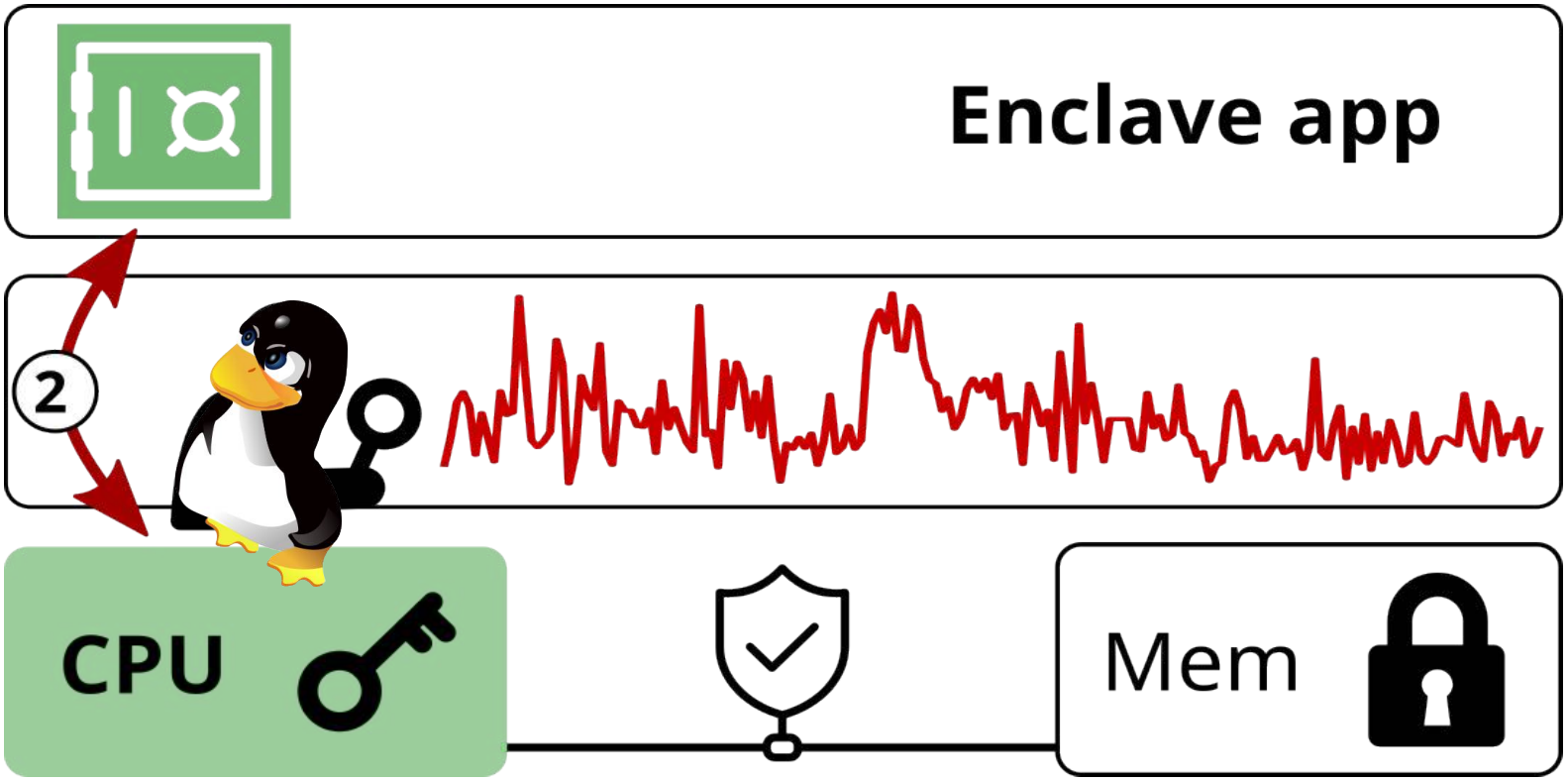
KU LEUVEN

Enclaved Execution: Reducing Attack Surface



Intel SGX: Hardware-level isolation and attestation

Enclaved Execution: Privileged Side Channels



Game changer: Untrusted OS → New class of powerful side channels!

SGX-Step: A Practical Attack Framework for Precise Enclave Execution Control

Jo Van Bulck
imec-DistriNet, KU Leuven
jo.vanbulck@cs.kuleuven.be

Frank Piessens
imec-DistriNet, KU Leuven
frank.piessens@cs.kuleuven.be

Raoul Strackx
imec-DistriNet, KU Leuven
raoul.strackx@cs.kuleuven.be

@SysTEX 2017

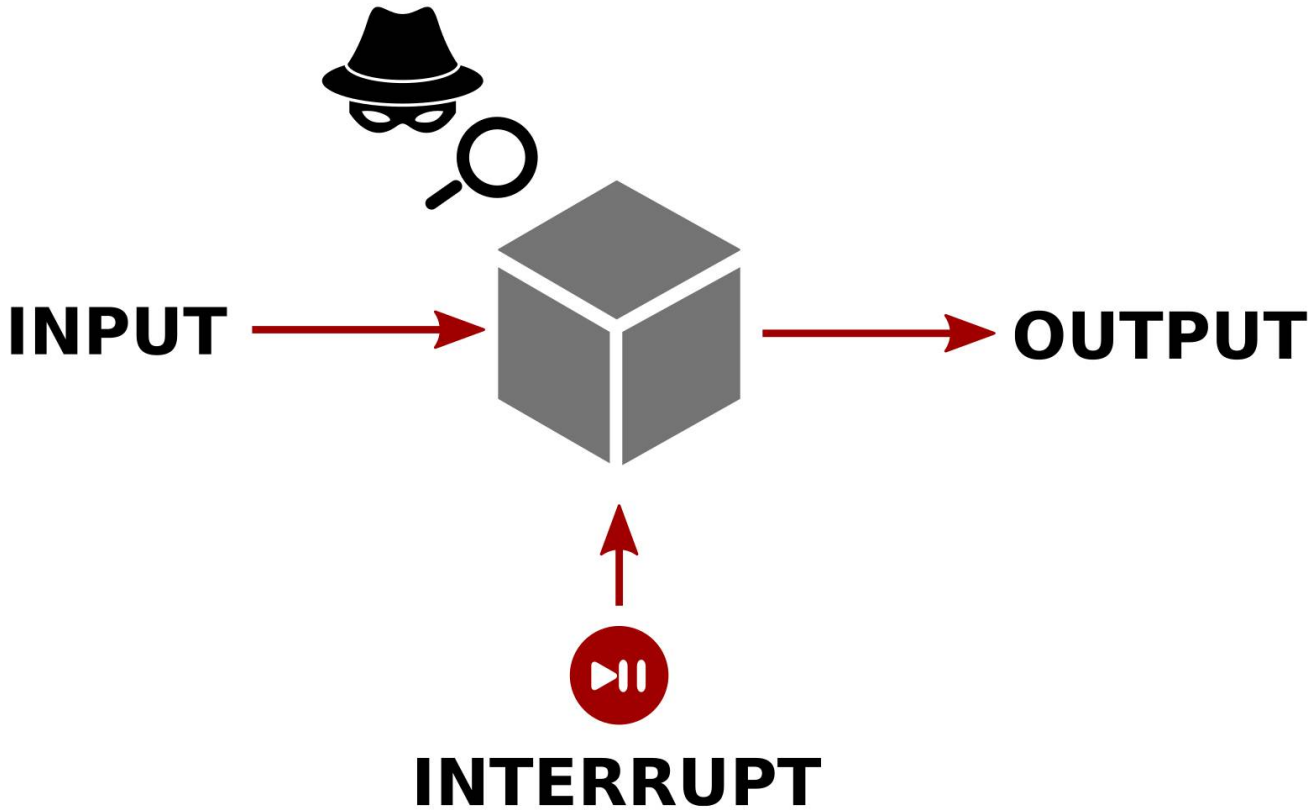
Challenge: Side-Channel Sampling Rate

Slow
shutter speed

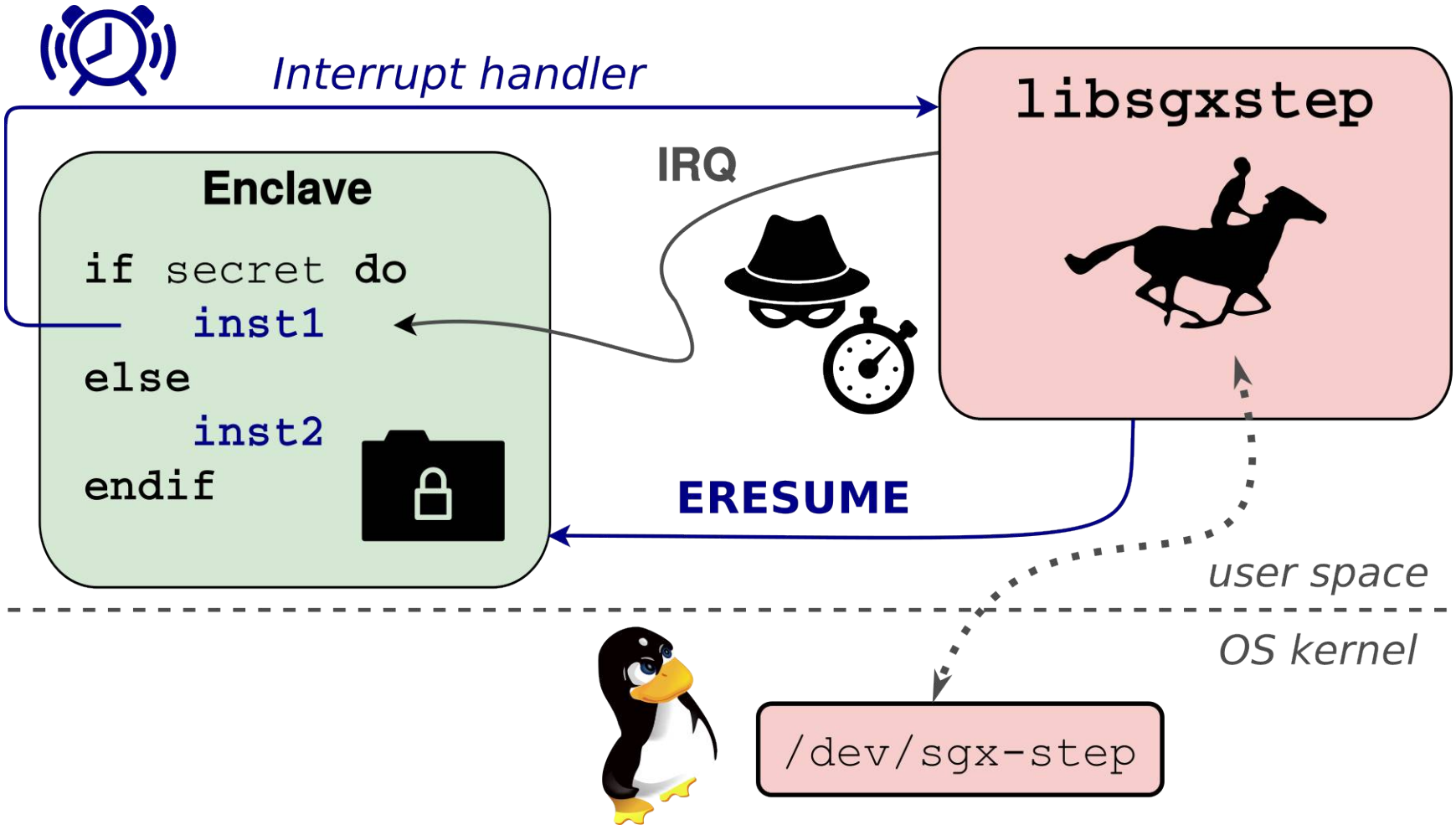
Medium
shutter speed

Fast
shutter speed

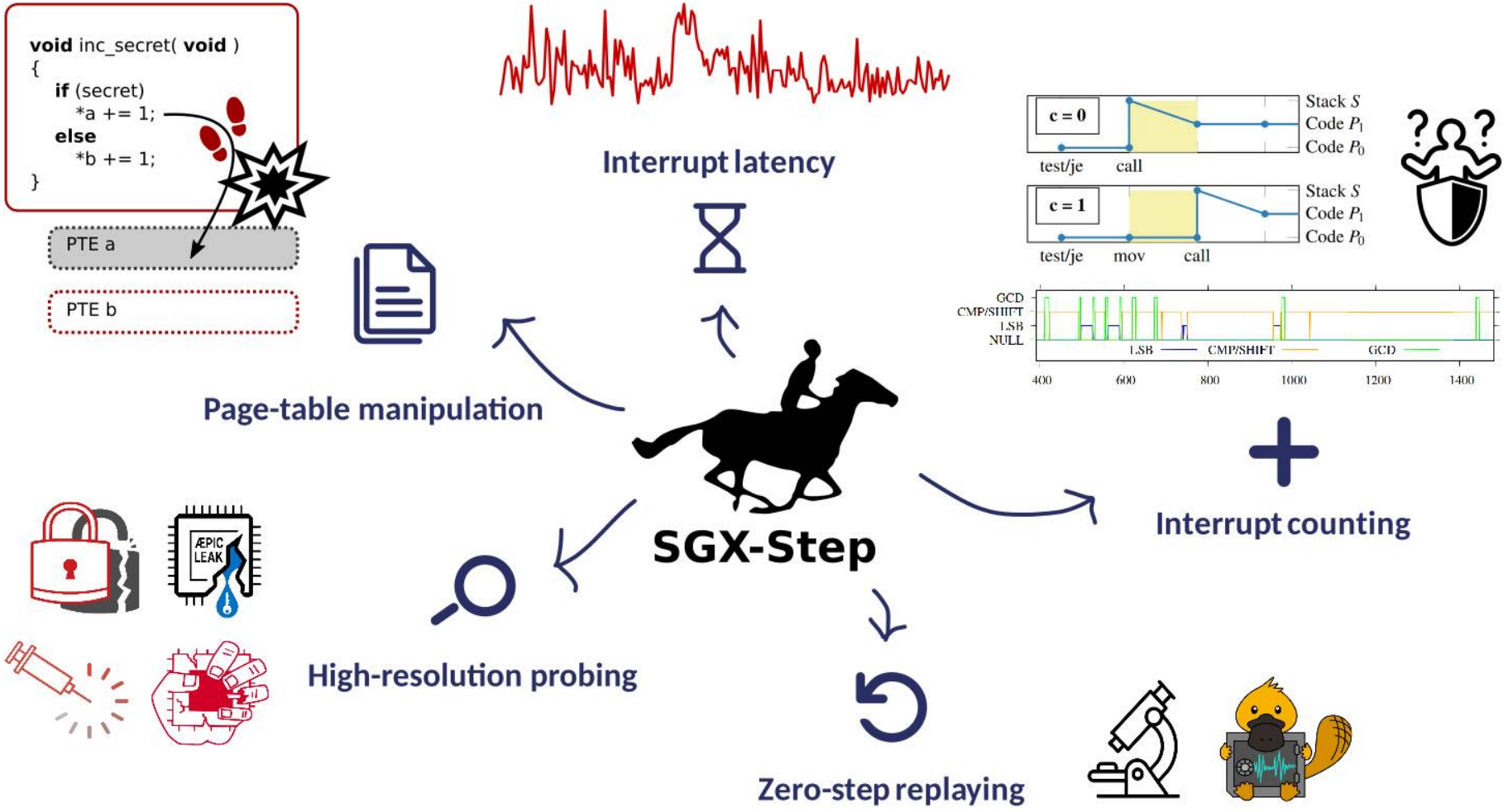
SGX-Step: Executing Enclaves one Instruction at a Time



SGX-Step: Executing Enclaves one Instruction at a Time



SGX-Step: A Versatile Open-Source Attack Framework



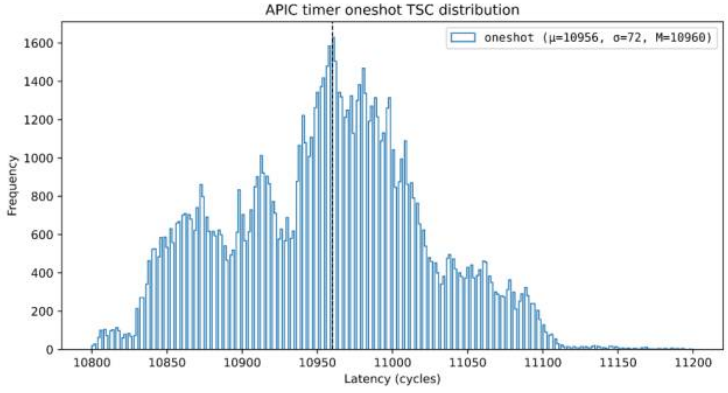
AEX-Notify: Thwarting Precise Single-Stepping Attacks through Interrupt Awareness for Intel SGX Enclaves

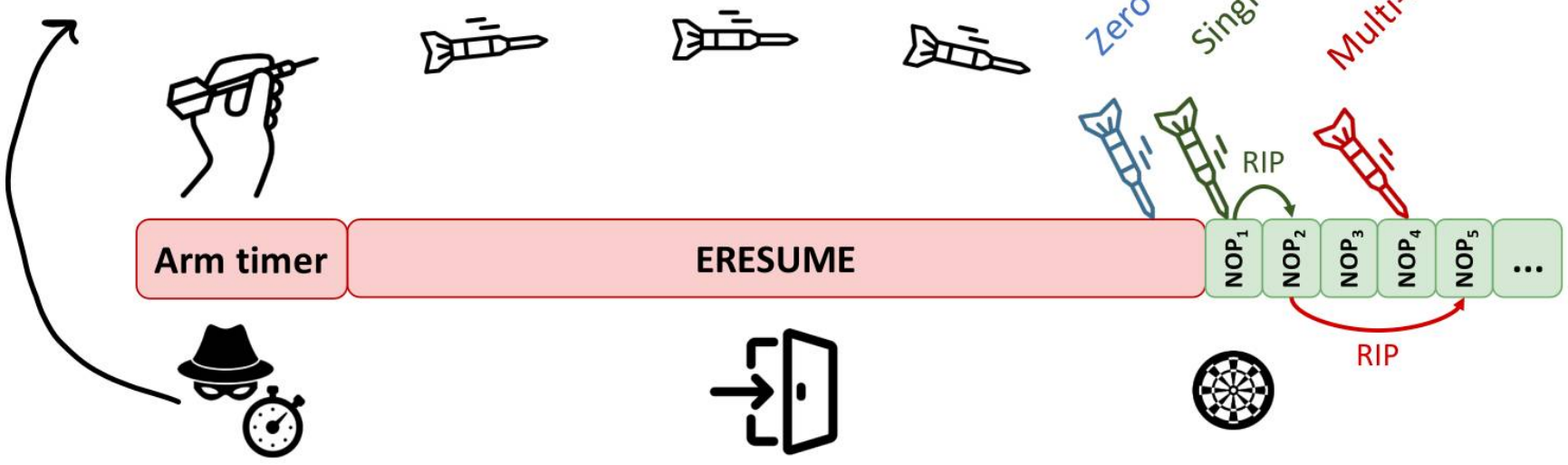
Scott Constable[1], Jo Van Bulck[2], Xiang Cheng[3], Yuan Xiao[1], Cedric Xing[1], Ilya Alexandrovich[1], Taesoo Kim[3], Frank Piessens[2], Mona Vij[1], and Mark Silberstein[4]

[1]*Intel Corporation,* [2]*imec-DistriNet, KU Leuven,* [3]*Georgia Institute of Technology,* [4]*Technion*

@USENIX Security 2023

Root-Causing SGX-Step: Aiming the Timer Interrupt

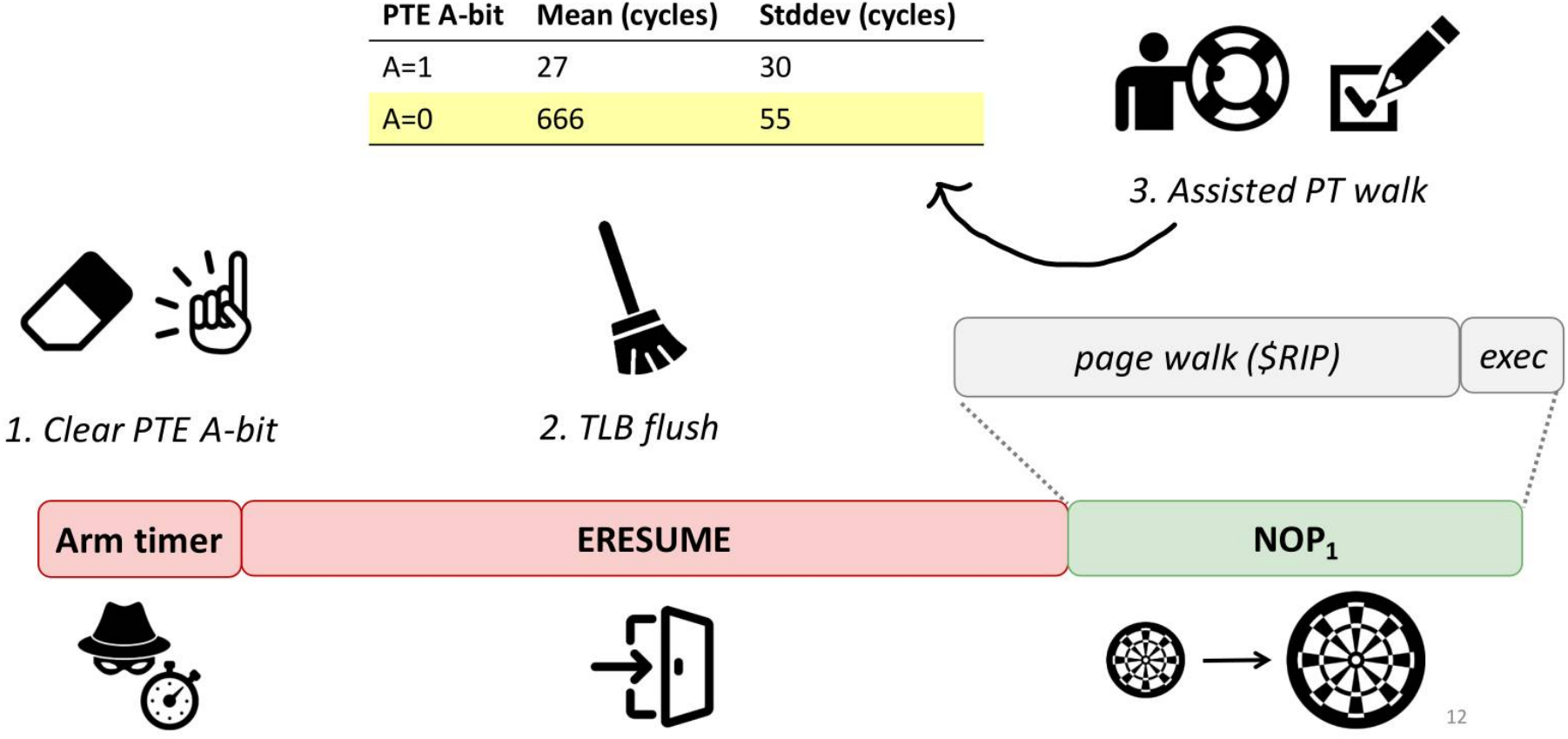




Constable et al. “AEX-Notify: Thwarting Precise Single-Stepping Attacks through Interrupt Awareness for Intel SGX Enclaves”, USENIX Security 2023.

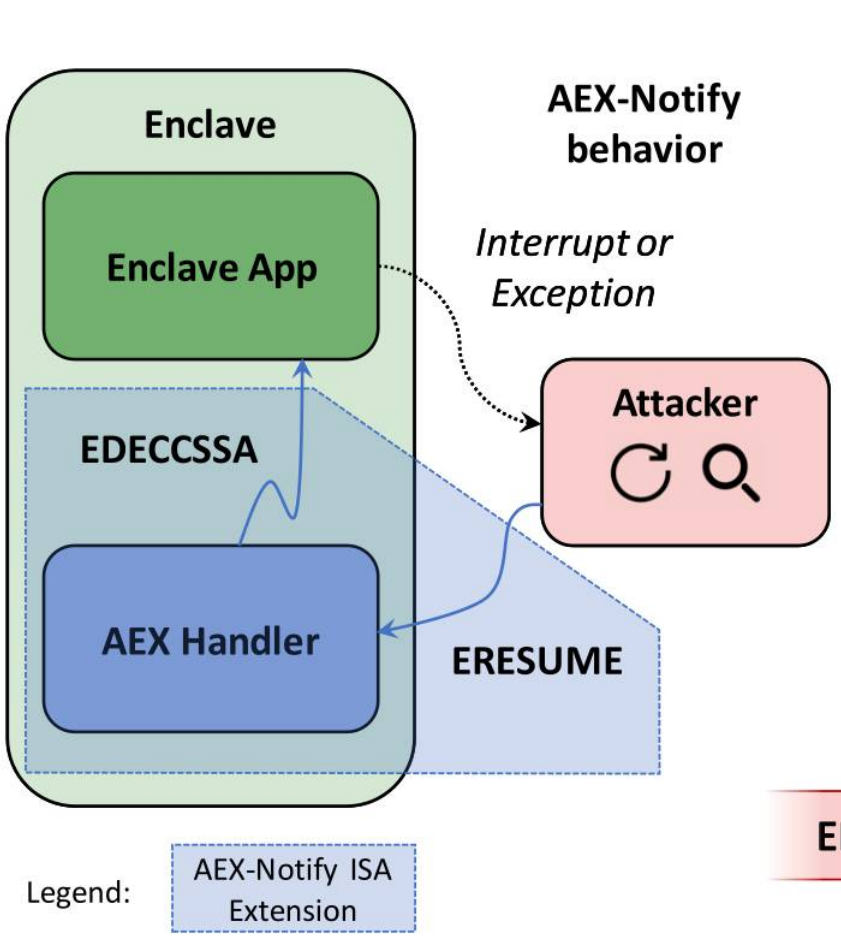
Root-Causing SGX-Step: Microcode Assists to the Rescue!

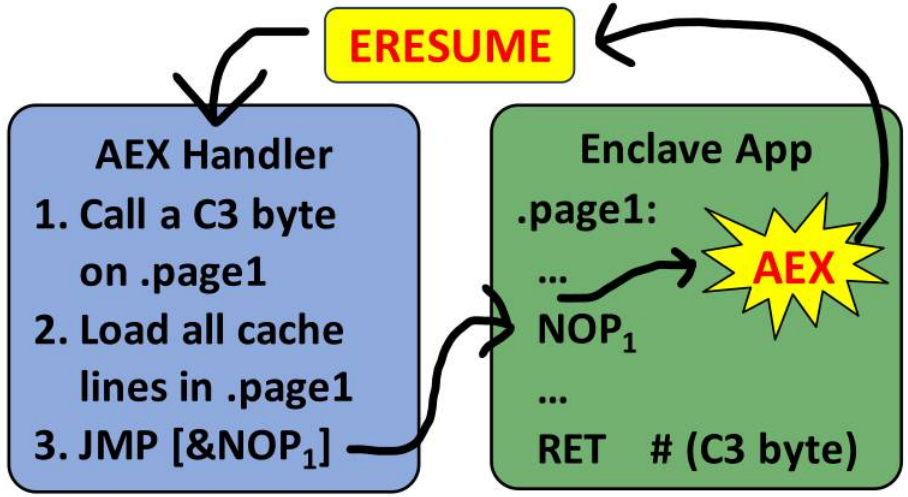
PTE A-bit	Mean (cycles)	Stddev (cycles)
A=1	27	30
A=0	666	55

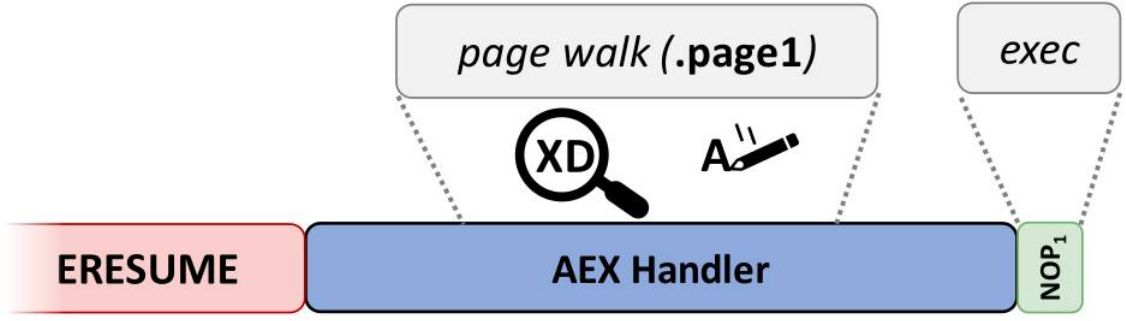


Constable et al. “AEX-Notify: Thwarting Precise Single-Stepping Attacks through Interrupt Awareness for Intel SGX Enclaves”, USENIX Security 2023.

AEX-Notify: Hardware-Software Co-Design Solution







Constable et al. “AEX-Notify: Thwarting Precise Single-Stepping Attacks through Interrupt Awareness for Intel SGX Enclaves”, USENIX Security 2023.

CHAPTER 8
ASYNCHRONOUS ENCLAVE EXIT NOTIFY AND THE EDECCSSA USER LEAF FUNCTION

8.1 INTRODUCTION

Asynchronous Enclave Exit Notify (AEX-Notify) is an extension to Intel® SGX that allows Intel SGX enclaves to be notified after an asynchronous enclave exit (AEX) has occurred. EDECCSSA is a new Intel SGX user leaf function (ENCLU[EDECCSSA]) that can facilitate AEX notification handling, as well as software exception handling. This chapter provides information about changes to the Intel SGX architecture that support AEX-Notify and ENCLU[EDECCSSA].

The following list summarizes the a
details are provided in Section 8.3)

- SECS.ATTRIBUTES.AEXNOTIFY
- TCS.FLAGS.AEXNOTIFY: This e
- SSA.GPRSGX.AEXNOTIFY: Enclave-writable byte that allows enclave software to dynamically enable/disable AEX notifications.

An AEX notification is delivered by ENCLU[ERESUME] when the following conditions are met:

SGX-Step led to **new x86 processor instructions!**

→ shipped in millions of devices ≥ 4th Gen Xeon CPU

There's a Catch: Page-Granular Spatial Resolution

Finally note that our proposed mitigation does not protect against interrupting enclaves and observing application code and data page accesses at a coarse-grained 4 KiB spatial resolution. In contrast to the fine-grained, instruction-granular interrupt-driven attacks we consider in this work, such controlled-channel attacks have received ample attention [18, 47, 56, 59] from the research community.

Constable et al. "AEX-Notify: Thwarting Precise Single-Stepping Attacks through Interrupt Awareness for Intel SGX Enclaves", USENIX Security 2023.

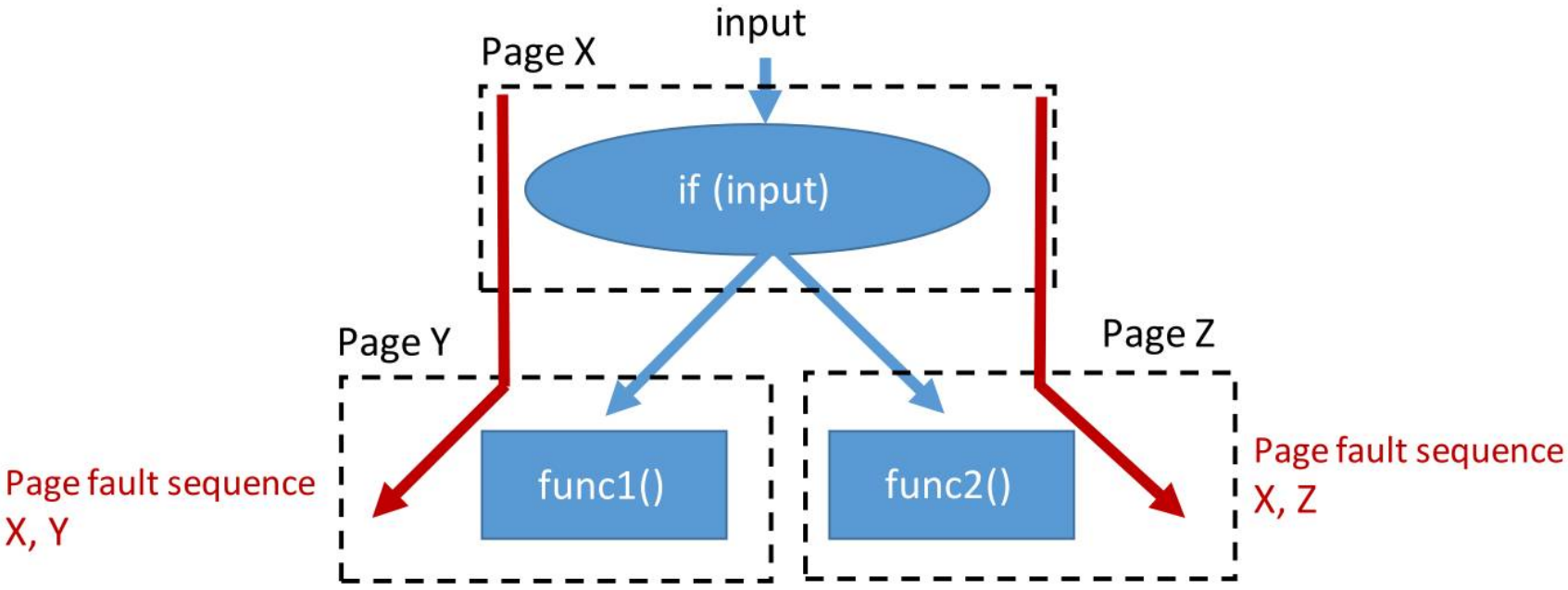
TLBlur: Compiler-Assisted Automated Hardening against Controlled Channels on Off-the-Shelf Intel SGX Platforms

Daan Vanoverloop[1], Andrés Sánchez*[2,4], Flavio Toffalini[2,3], Frank Piessens[1], Mathias Payer[2], and Jo Van Bulck[1]

[1]*DistriNet, KU Leuven,* [2]*EPFL,* [3]*RUB,* [4]*Amazon*

@USENIX Security 2025

Idea: Page Faults as a Side Channel

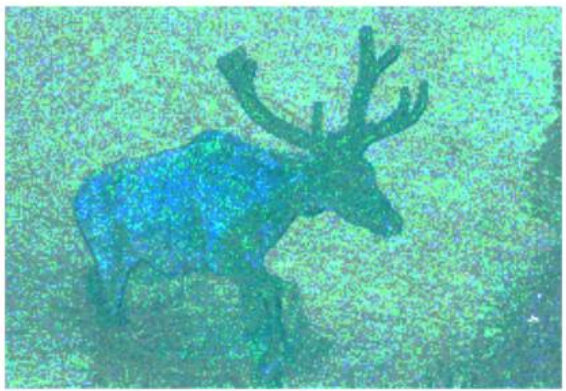
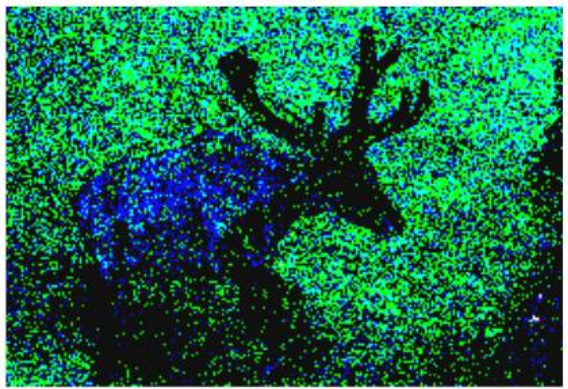


Xu et al.: "Controlled-channel attacks: Deterministic side channels for untrusted operating systems", Oakland 2015.

⇒ Page fault traces leak **private control data/flow**

Xu et al. "Controlled-Channel Attacks: Deterministic Side Channels for Untrusted Operating Systems", IEEE S&P 2015.

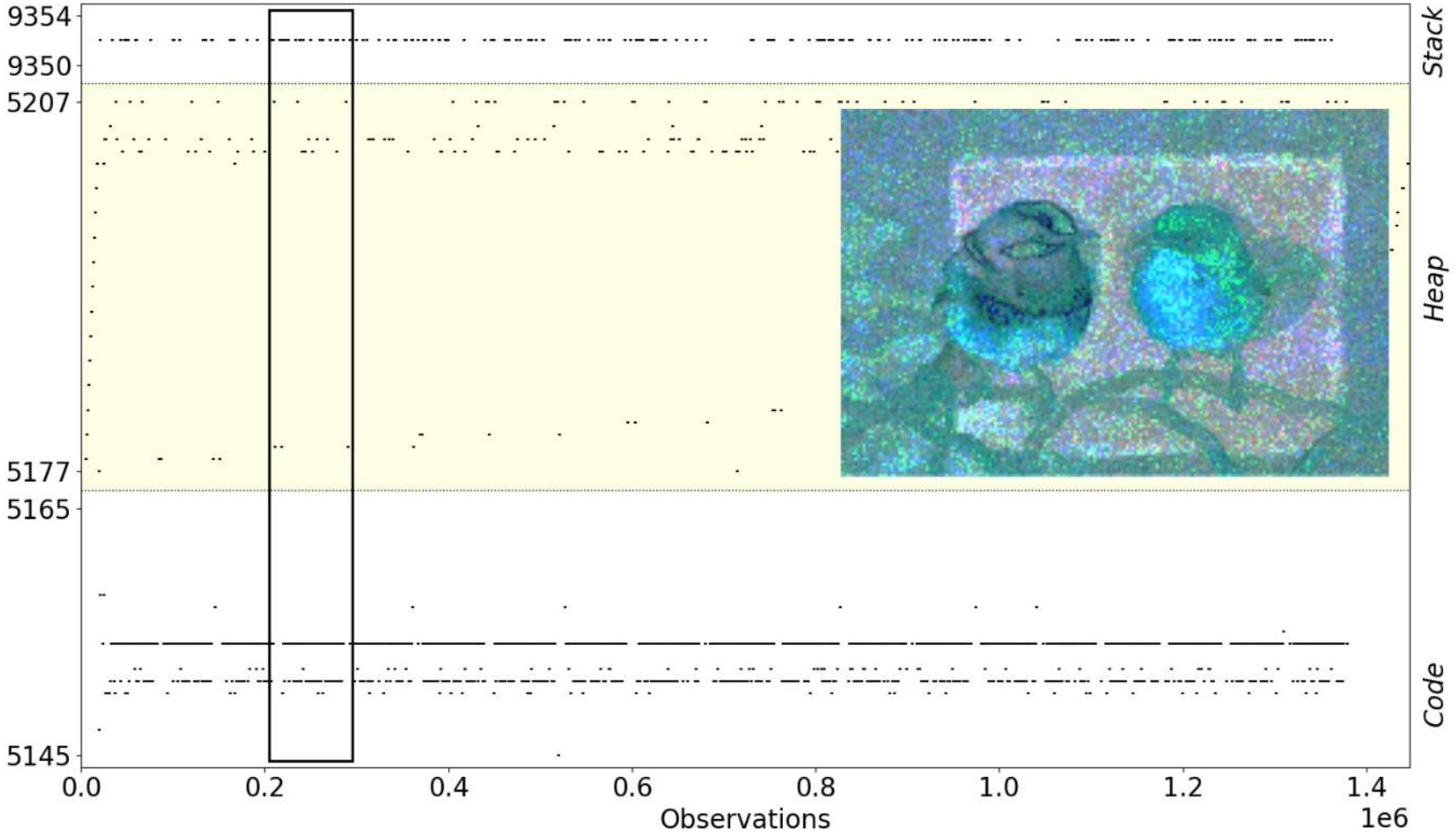
Why Mitigating Single-Stepping is Not Enough

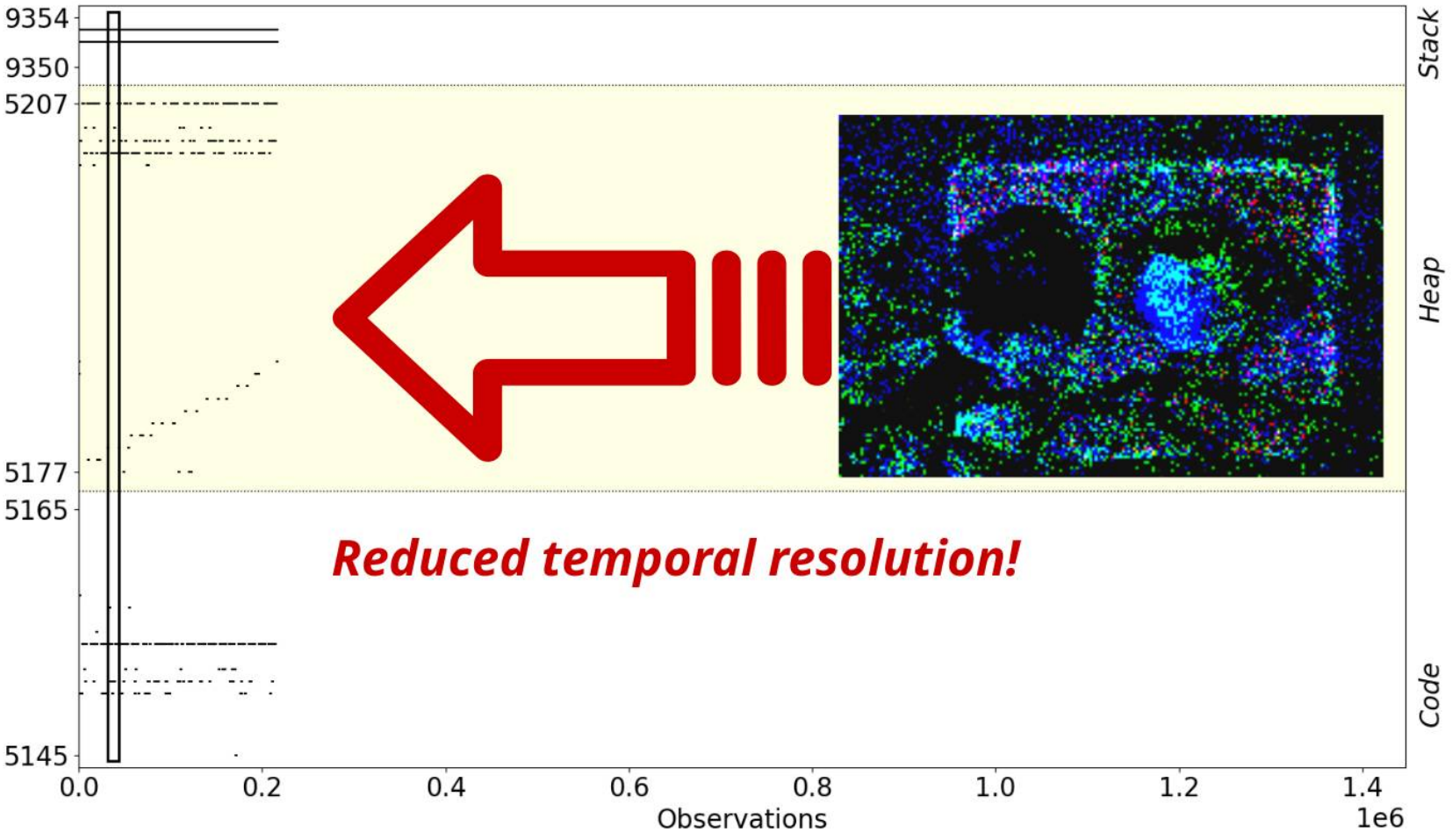
Original (left), Xu et al. (middle), our attack with AEX-Notify single-stepping defense (right)

Vanoverloop et. al “TLBlur: Compiler-Assisted Automated Hardening against Controlled Channels on Off-the-Shelf Intel SGX Platforms”, USENIX’25.

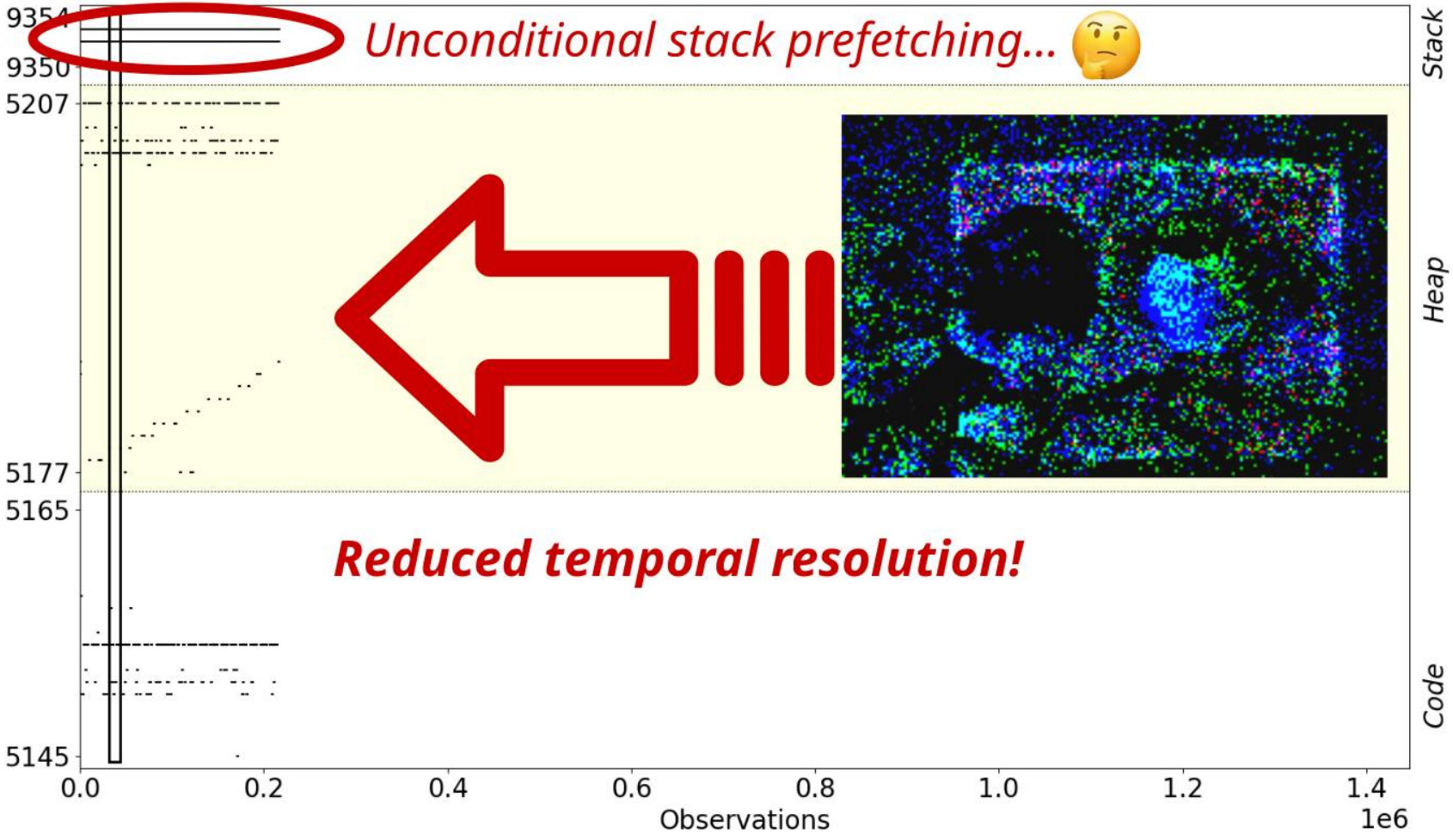
Libjpeg: AEX-Notify's Temporal Reduction in Practice



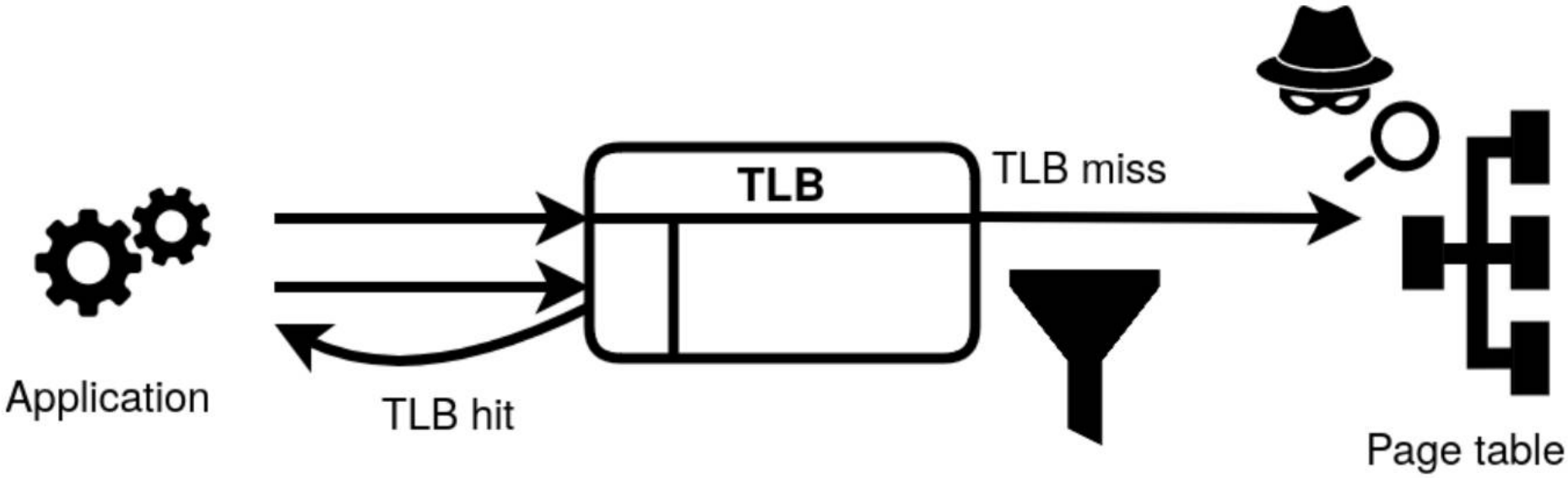
Libjpeg: AEX-Notify's Temporal Reduction in Practice



Libjpeg: AEX-Notify's Temporal Reduction in Practice

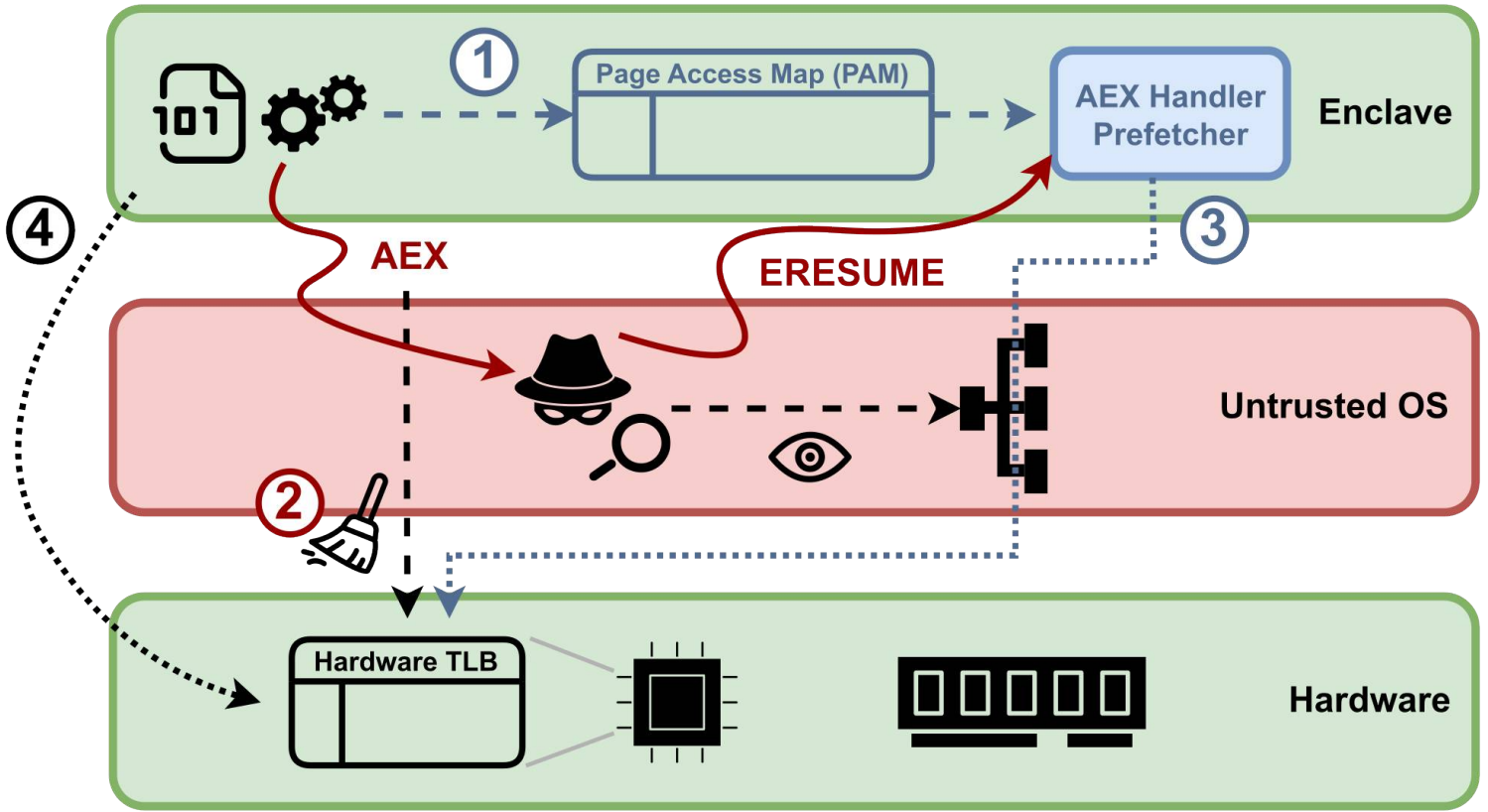


Idea: TLB as a “Filter” to Hide Page Accesses



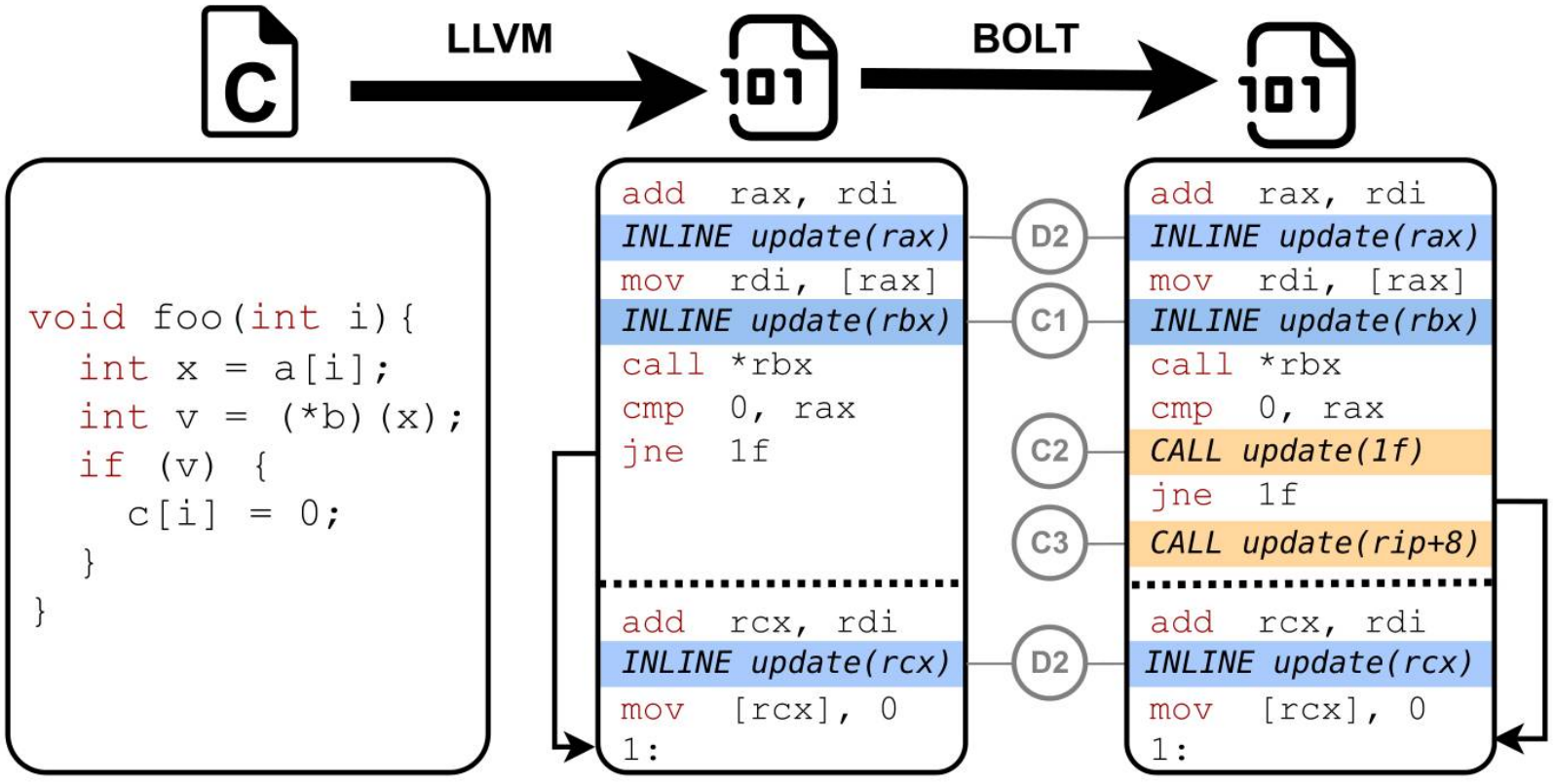
Vanoverloop et. al “TLBlur: Compiler-Assisted Automated Hardening against Controlled Channels on Off-the-Shelf Intel SGX Platforms”, USENIX’25.

TLBlur: Self-Monitoring and Restoring Enclave Page Accesses



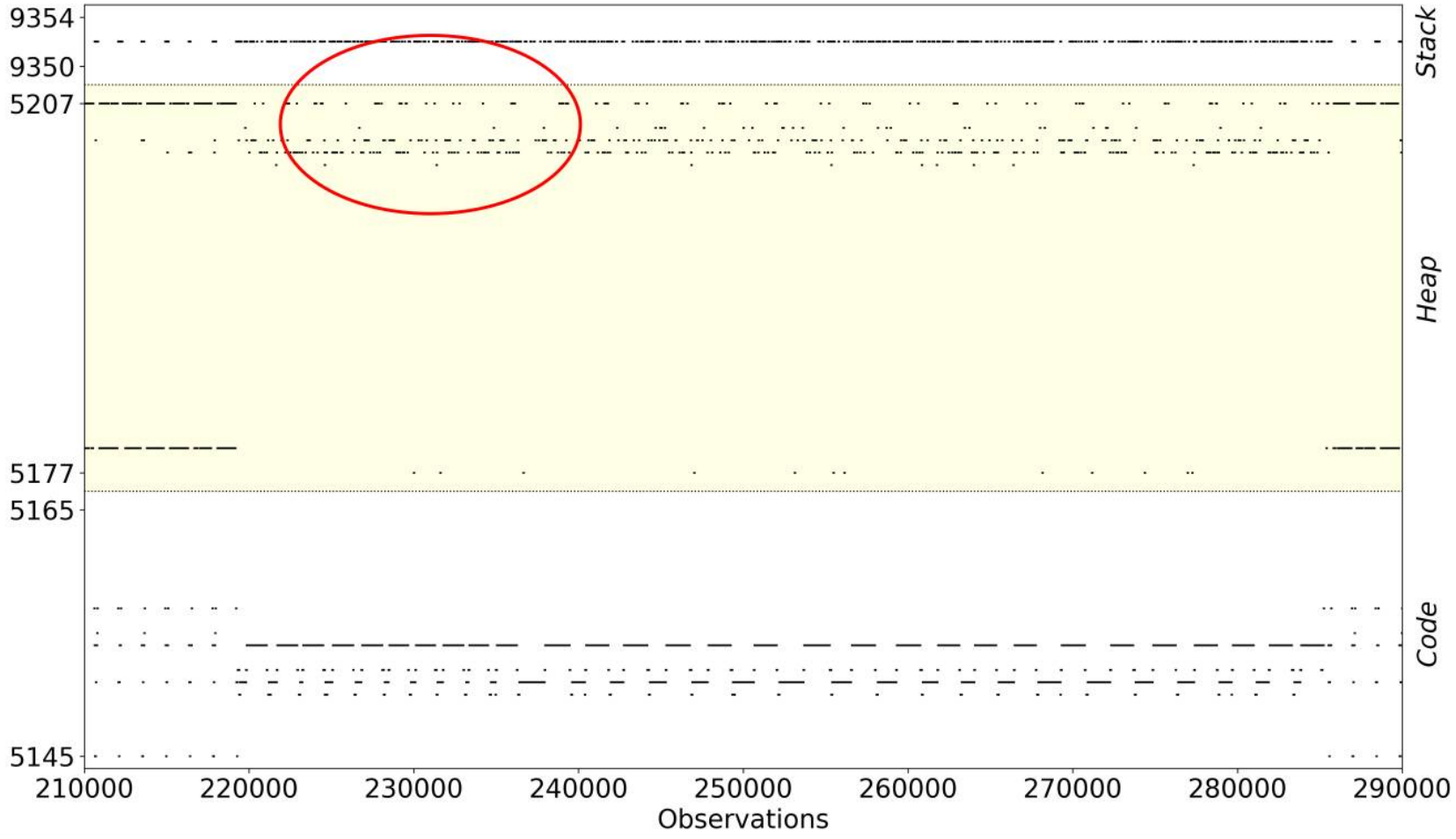
Vanoverloop et. al "TLBlur: Compiler-Assisted Automated Hardening against Controlled Channels on Off-the-Shelf Intel SGX Platforms", USENIX'25.

Instrumentation to Self-Monitor Page Accesses at Runtime

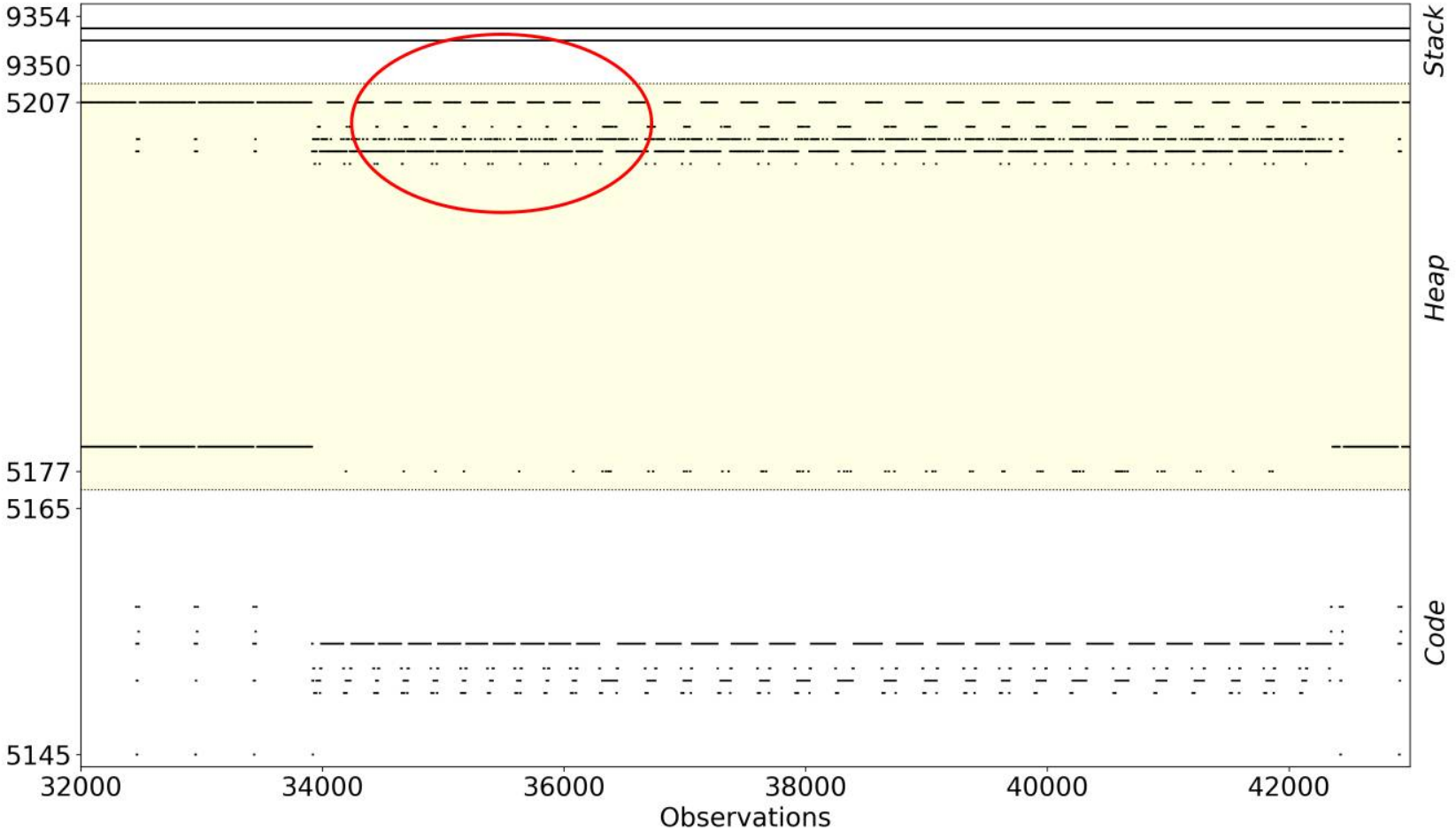


Vanoverloop et. al “TLBlur: Compiler-Assisted Automated Hardening against Controlled Channels on Off-the-Shelf Intel SGX Platforms”, USENIX’25.

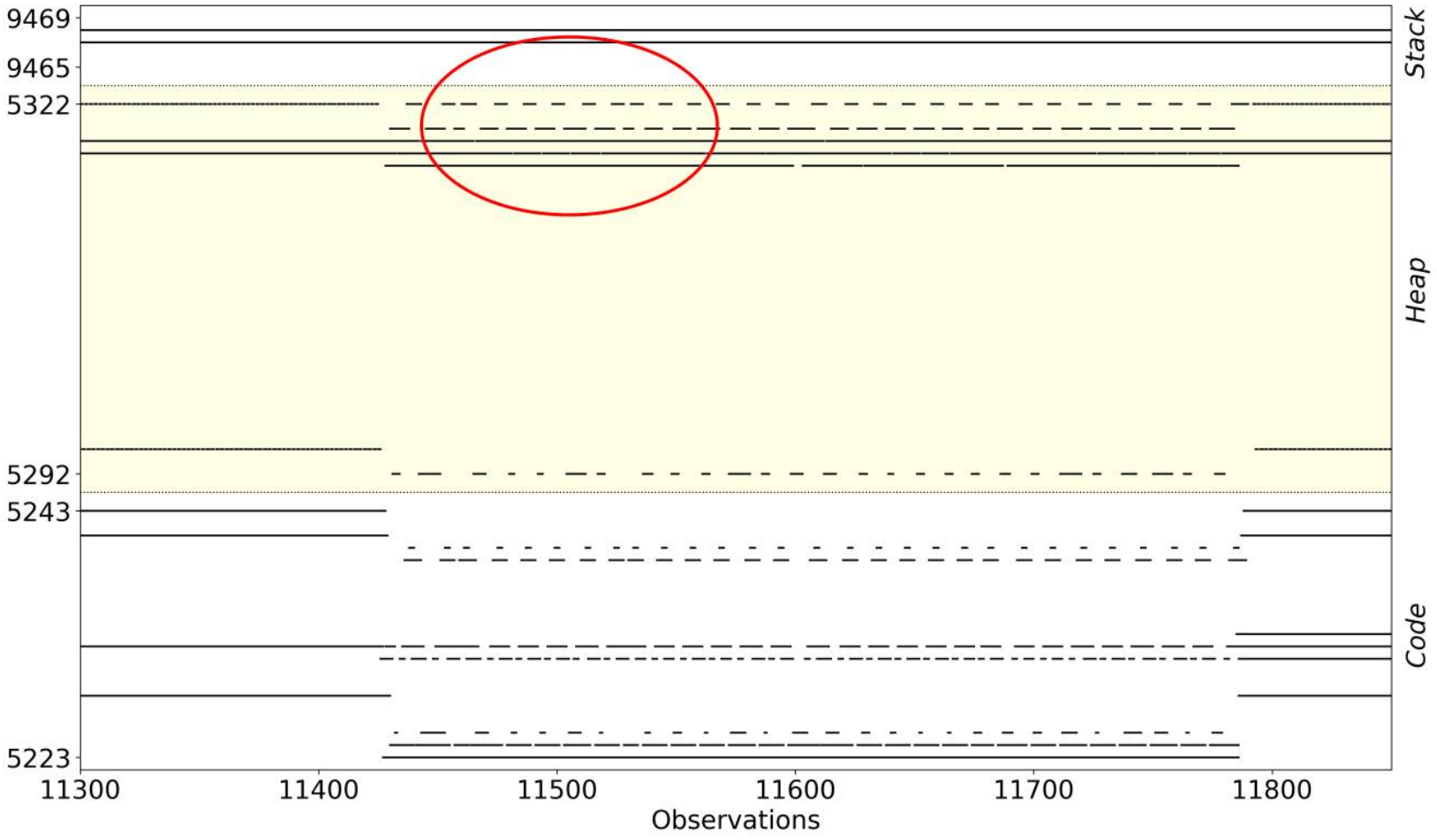
Leakage Reduction in Practice: Libjpeg Single-Stepping



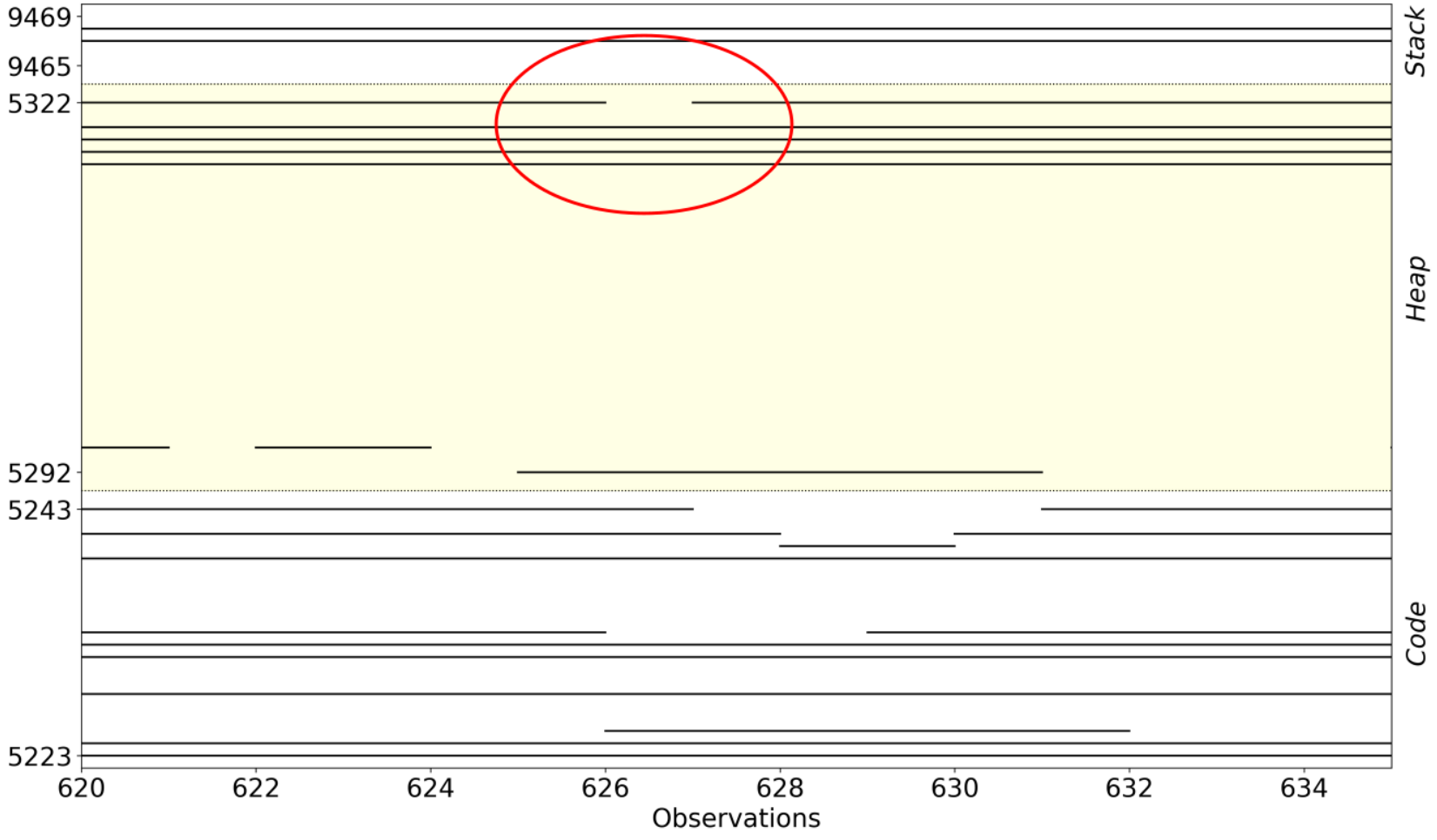
Leakage Reduction in Practice: Libjpeg Page Faults



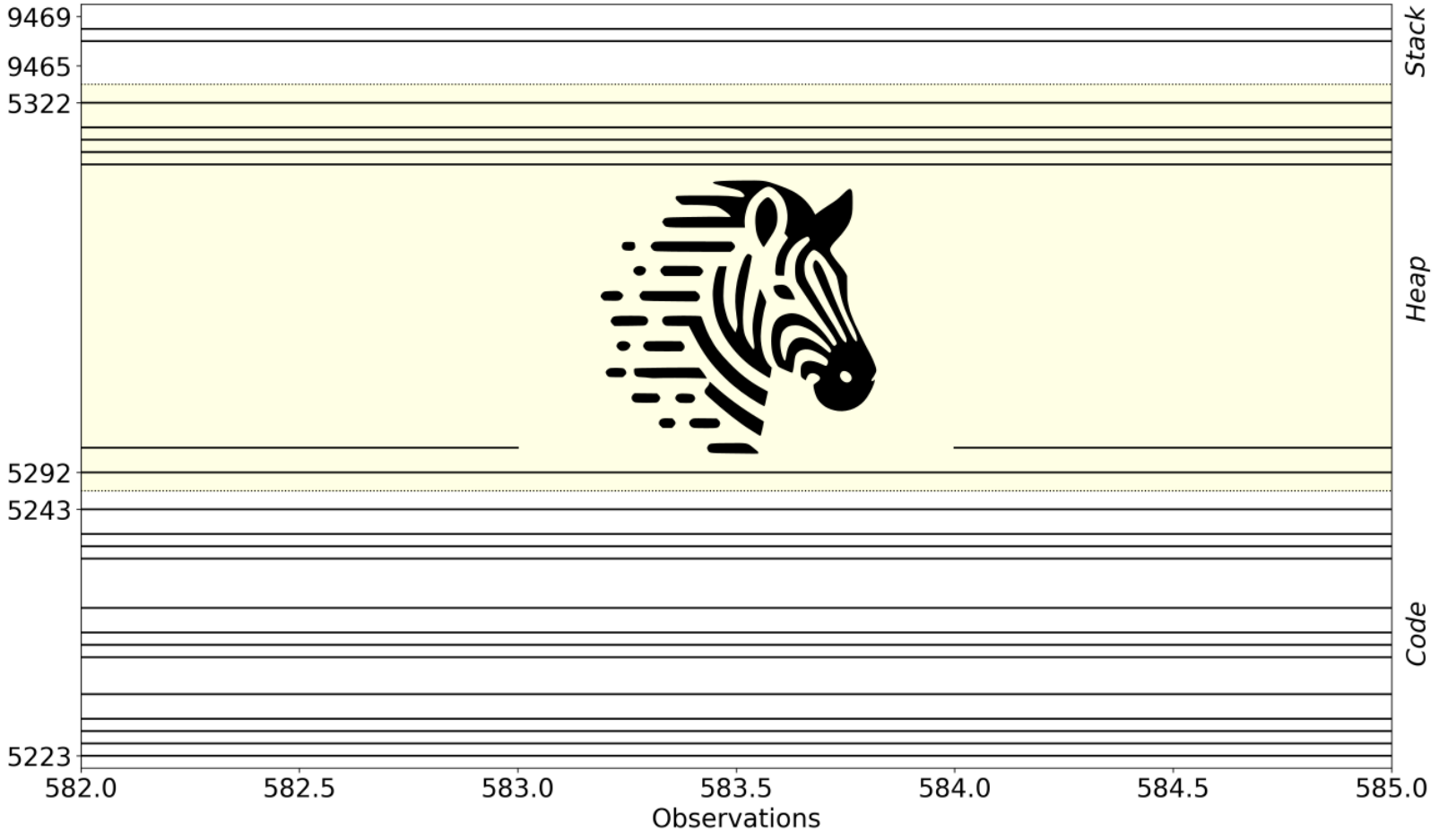
Leakage Reduction in Practice: Libjpeg TLBlur (N=10)



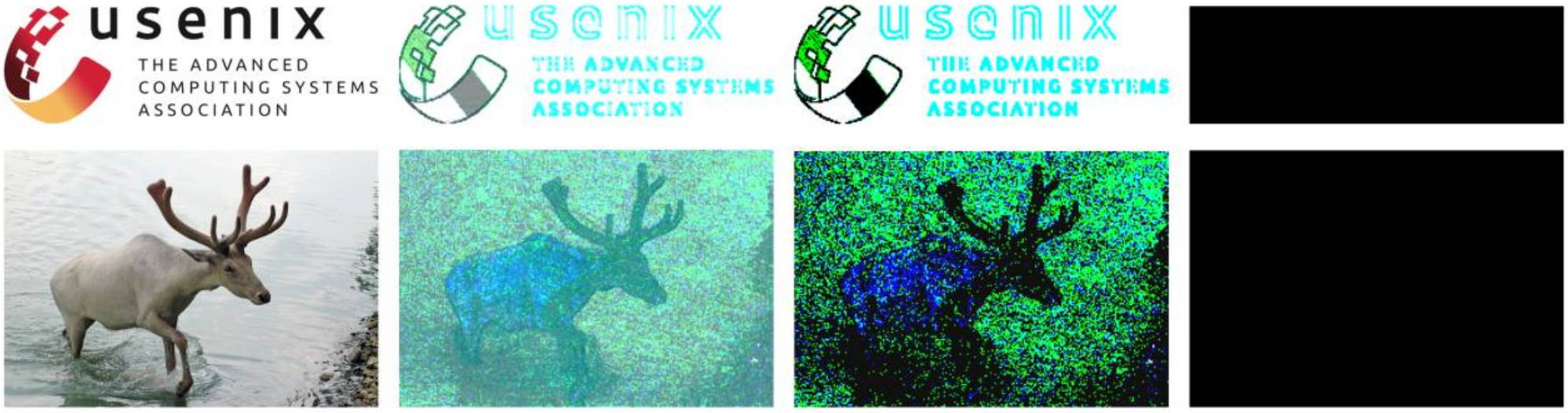
Leakage Reduction in Practice: Libjpeg TLBlur (N=20)



Leakage Reduction in Practice: Libjpeg TLBlur (N=30)



TLBlur: Compiler-Assisted Leakage Reduction in Practice

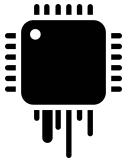


Automated “blurring” of page-access traces in space and time

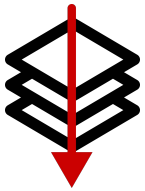
Vanoverloop et. al “TLBlur: Compiler-Assisted Automated Hardening against Controlled Channels on Off-the-Shelf Intel SGX Platforms”, USENIX’25.

Conclusions and Take-Away

New era of **confidential computing** for the cloud



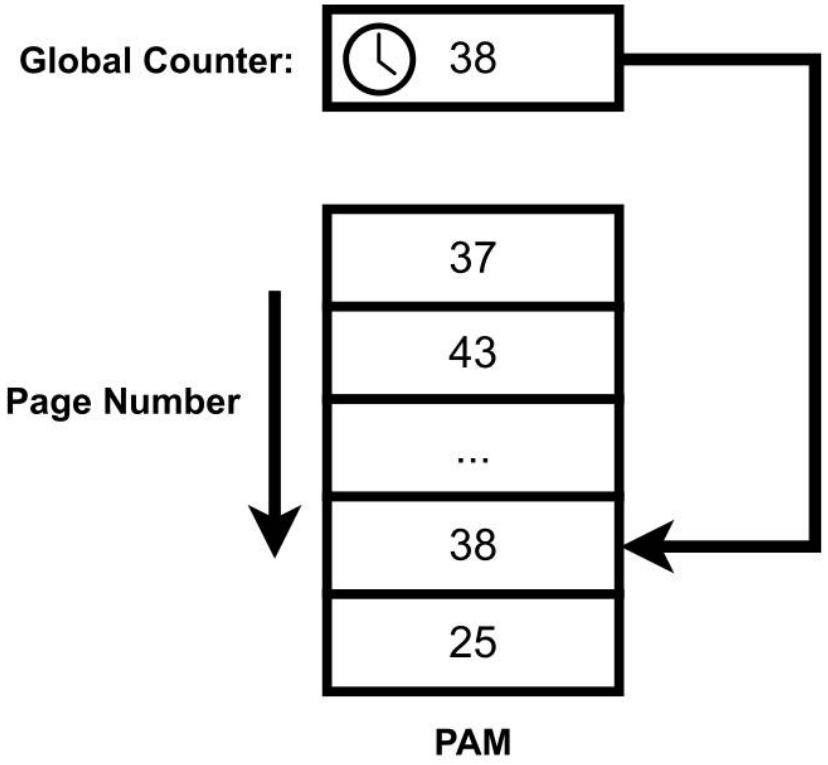
→ Side-channel amplification for **privileged adversaries**



Vertically integrated **hardware-software co-design** mitigations

Thank you! Questions?

Efficiently Updating the Page-Access Map

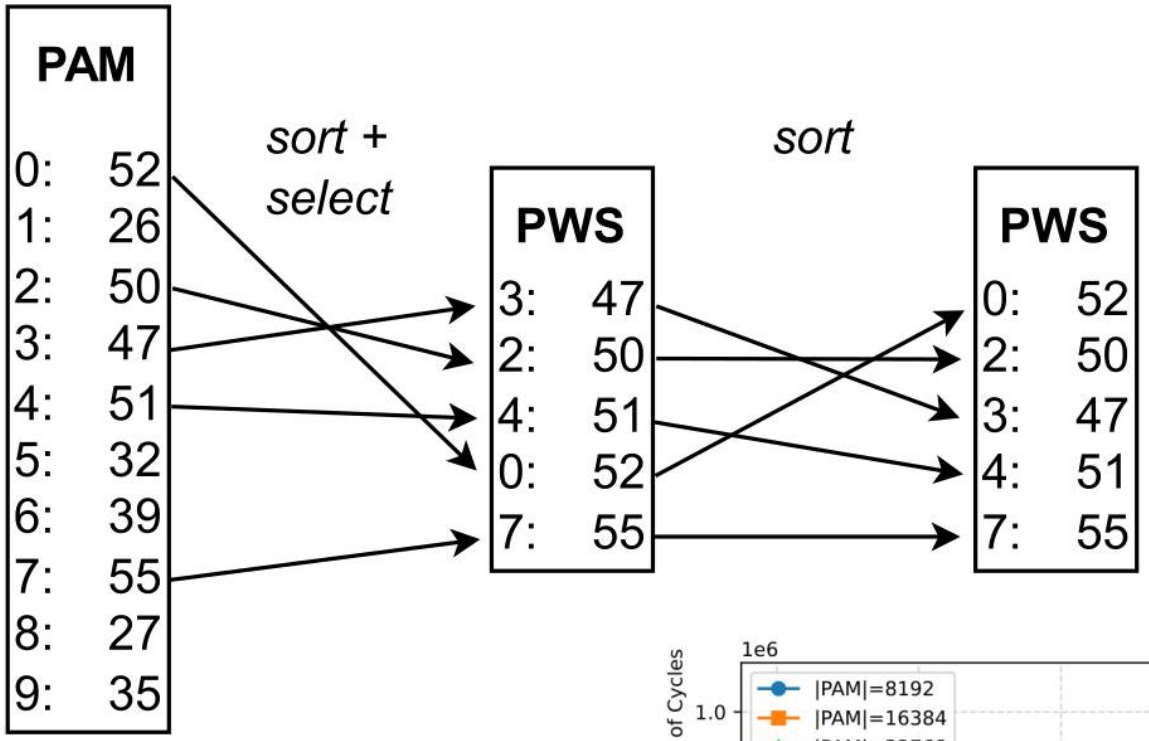


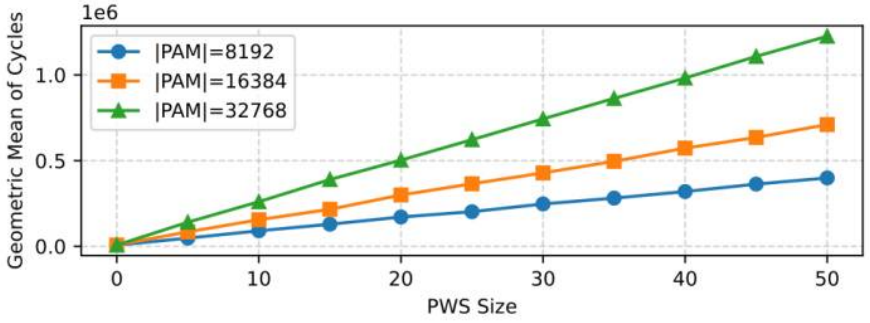

```
tlblur_pam_update:
    page_num = compute_index(addr)

    counter = global_counter
    counter++

    global_counter = counter
    pam[page_num] = counter
```

Constant-Time Page-Access Map Sorting





Performance Overhead