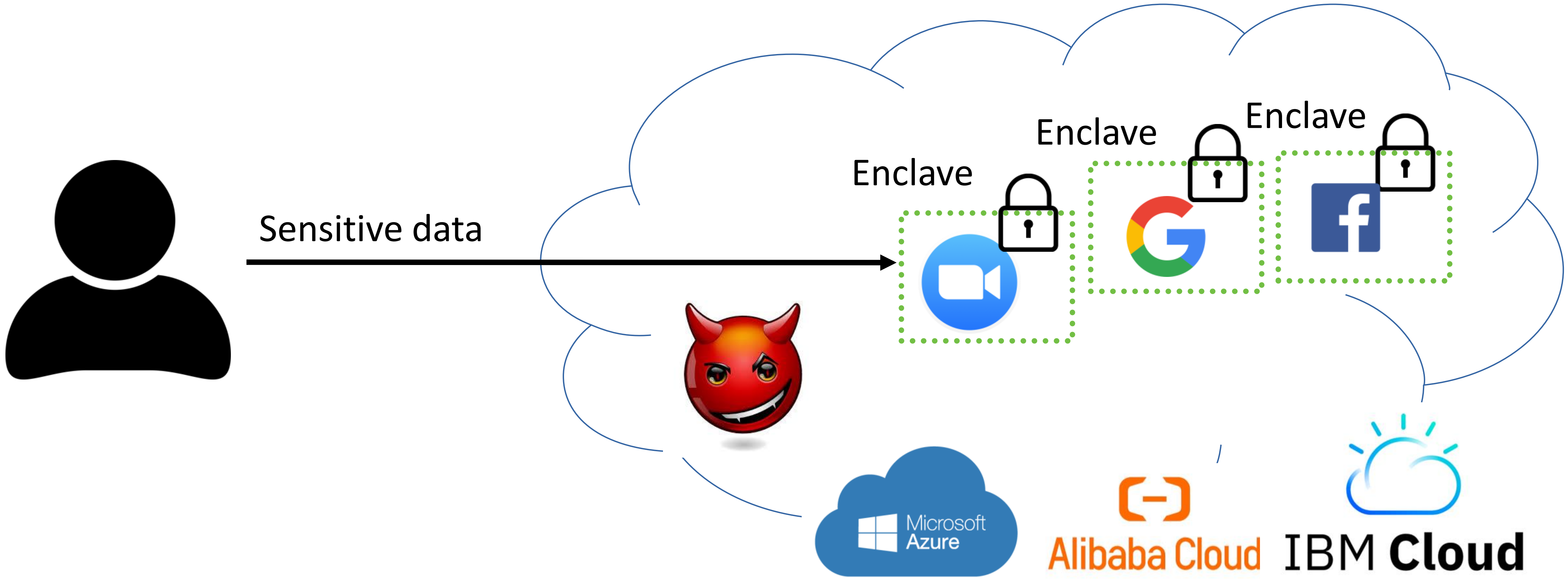
Autarky: Closing controlled channels with self-paging enclaves

Meni Orenbach, Technion

Andrew Baumann, Microsoft Research

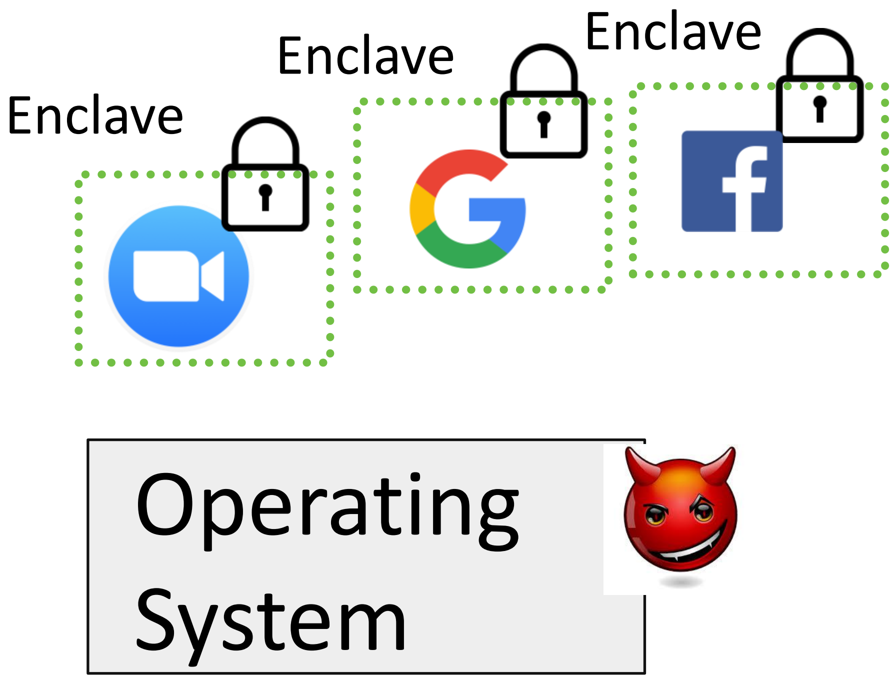
Mark Silberstein, Technion

Public cloud computing



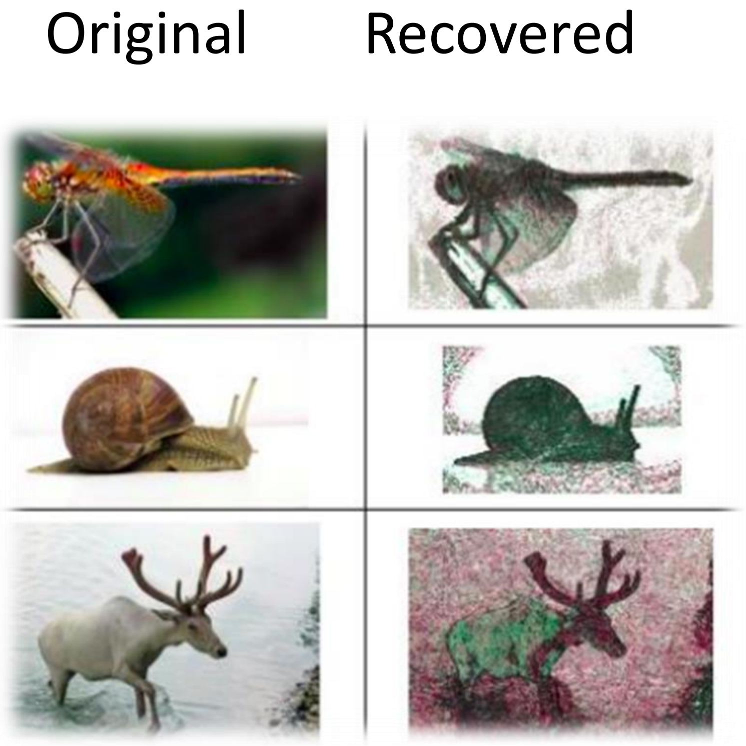
Intel SGX

- Isolated user-mode environment
- Commodity CPUs
- Small trusted computing base
  - CPU
  - Enclave’s code and data
    - Confidentiality
    - Integrity



Page fault side-channel attack

- OS-level attacker
  - **Induces** page faults
  - **Tracks** faulted address
  - **Infer secrets content** that depends on page access patterns
    - Control-dependent accesses
    - Data-dependent accesses



Xu, Y., Cui, W. and Peinado, M., 2015. **Controlled-Channel Attacks:** Deterministic Side Channels for Untrusted Operating Systems.

Controlled-channel attack


ars TECHNICA
BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE STORE

SPECULATIVE EXECUTION STRIKES AGAIN —

Intel SGX is vulnerable to an unfixable flaw that can steal crypto keys and more

Just when you thought it was secure again, Intel's digital vault falls to a new attack.

DAN GOODIN - 3/11/2020, 12:40 AM

- Precursor to other attacks
  - Foreshadow [Usenix Security'18]
  - Sgxspectre [arXiv'18]
  - LVI [IEEE S&P'20]
  - Microscope [ISCA'19]
  - Zombieload [CCS'19]

Why?

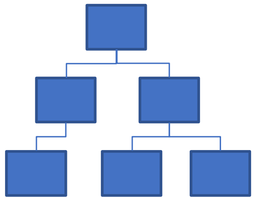
- **Attacker controls the channel**
- **Precise**
- **No noise**

Agenda

- Background
- Controlled-Channel Attack
- Self-Paging Enclaves
- Evaluation

SGX virtual memory protection

- SGX **validates** the OS does not insert **spurious mappings**



Page table
(maintained by OS)

VA	PA
10000	f0000

SGX Reverse page table

(Inaccessible by OS)

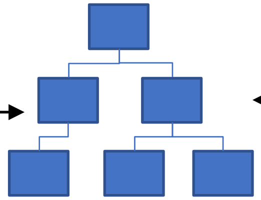
PA	VA
f0000	10000

- SGX **does not** validate the **prescence** of expected mappings

The missing component

?

Side-channel attacks defense



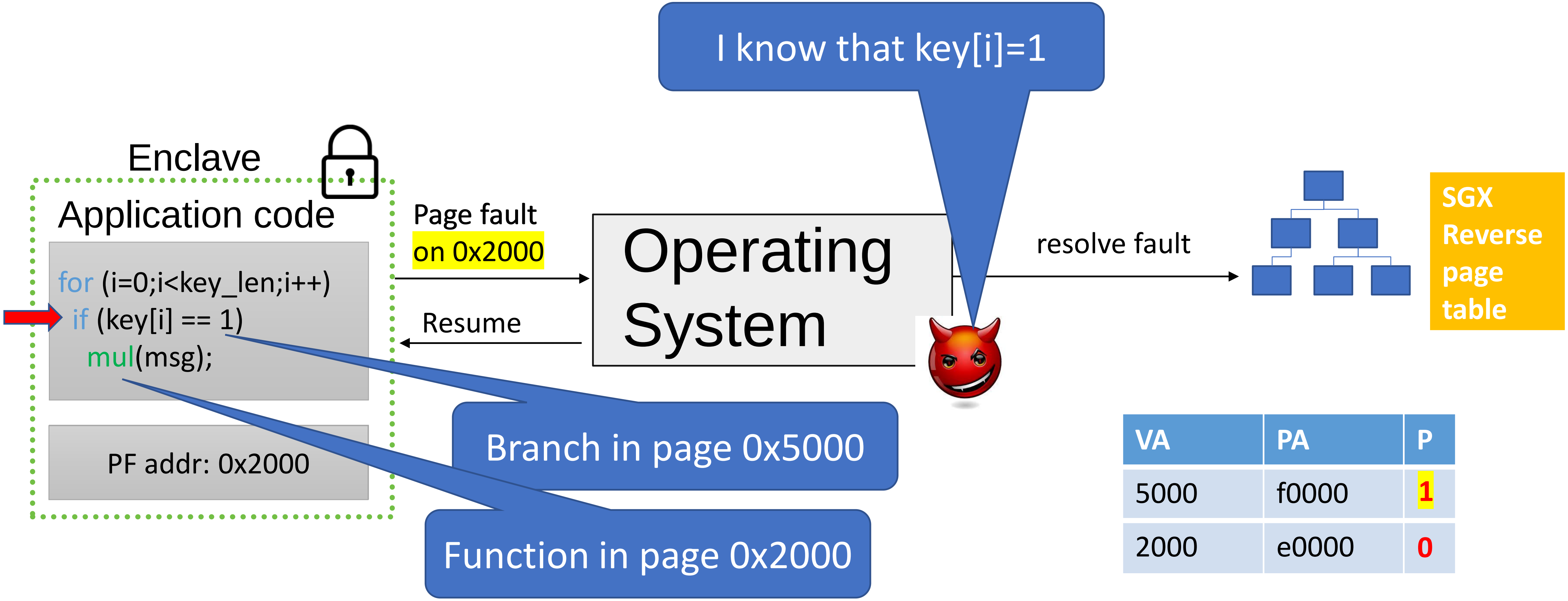
Active mapping attacks defense

SGX Reverse page table

Validate presence of expected mappings

Validate mapping

Implication: Controlled channel attack



VA	PA	P
5000	f0000	1
2000	e0000	0

Existing Software Mitigations

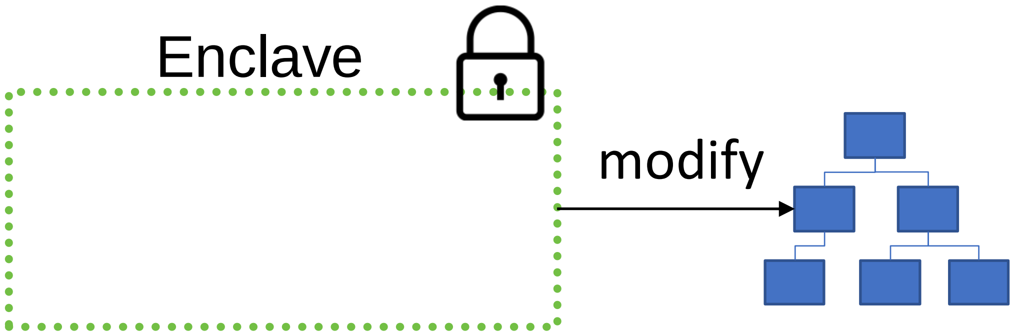
- Detect attack due to high frequency of exceptions
  - **Restrict** demand-paging
  - False positive occurrence
- Provably obfuscate all memory accesses
  - Orders of magnitude **performance impact**

Software mitigations are limited

[1] Ming-Wei Shih, San ... rams. In NDSS'2017.
[2] Oleksii Oleksenko, B ... al side-channel attacks. In USENIX ATC'2018.
[3] Sanchuan Chen, Xia ... ution with Déjá Vu. In Asia CCS'2017.
[4] Sajin Sasy, Sergey Gorbunov, and Christopher W. Fletcher. ZeroTrace: Oblivious memory primitives from Intel SGX. In NDSS'2018.

Existing Hardware Mitigations

- Private enclave page tables

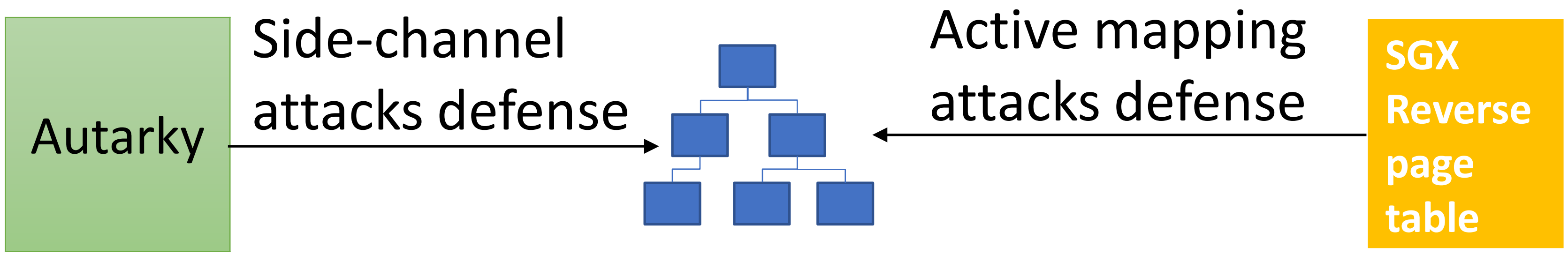


Requires major changes to SGX internals
since SGX is entangled with the x86 architecture

[1] Victor Costan … 2016.
[2] Dayeol Lee, D… .
[3] Shaizeen Aga …

Our solution: Autarky

- Minimal extension to SGX OS-hardware interface
  - **Backward-compatible** with SGX
  - **Validate presence** of expected mappings



Agenda

- Background
- Controlled-Channel Attack
- Self-Paging Enclaves
- Evaluation

Design principles

Force the OS to call the enclave on every page fault

Secure demand-paging

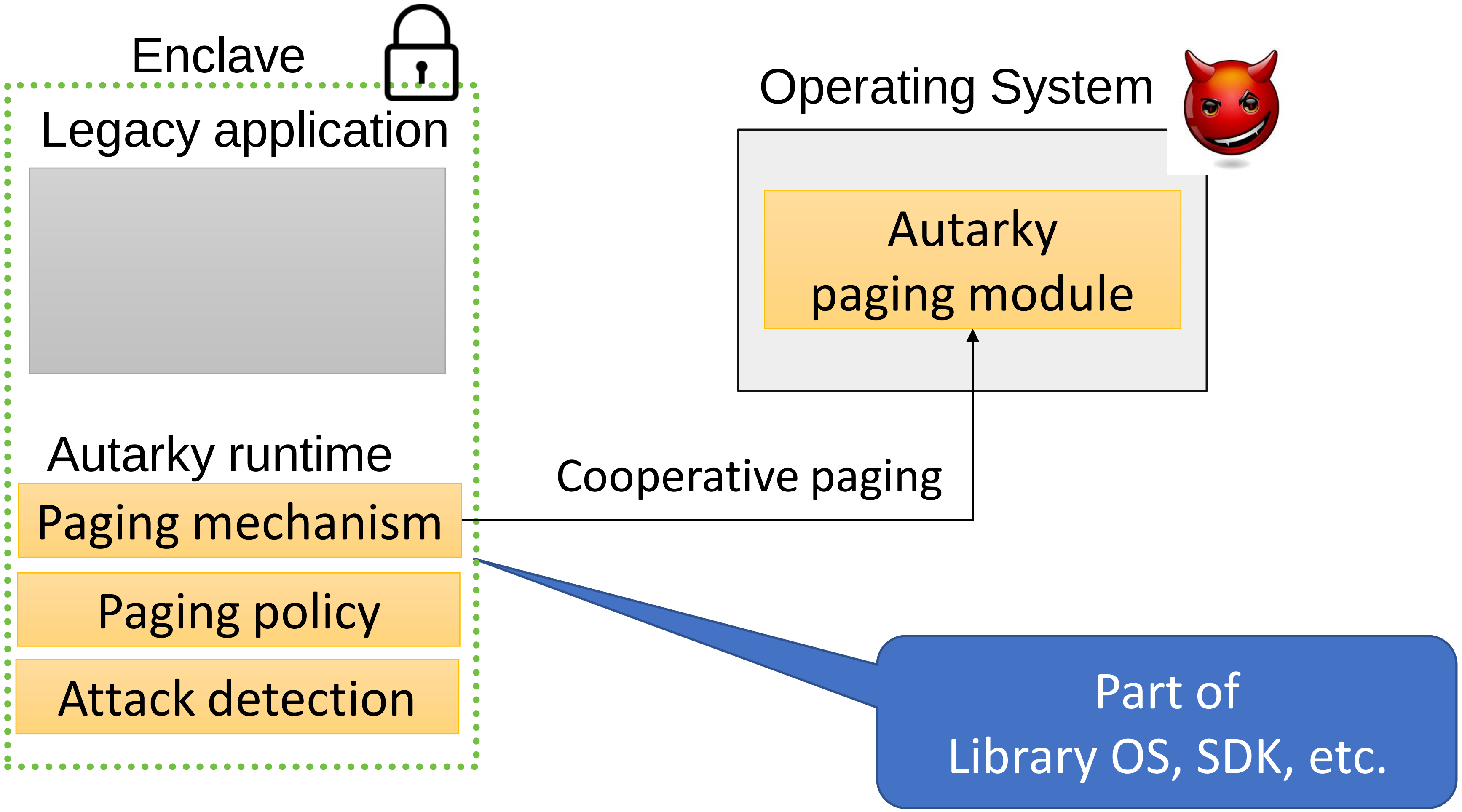
Give enclave power to control all page faults

Enclave-OS cooperative paging

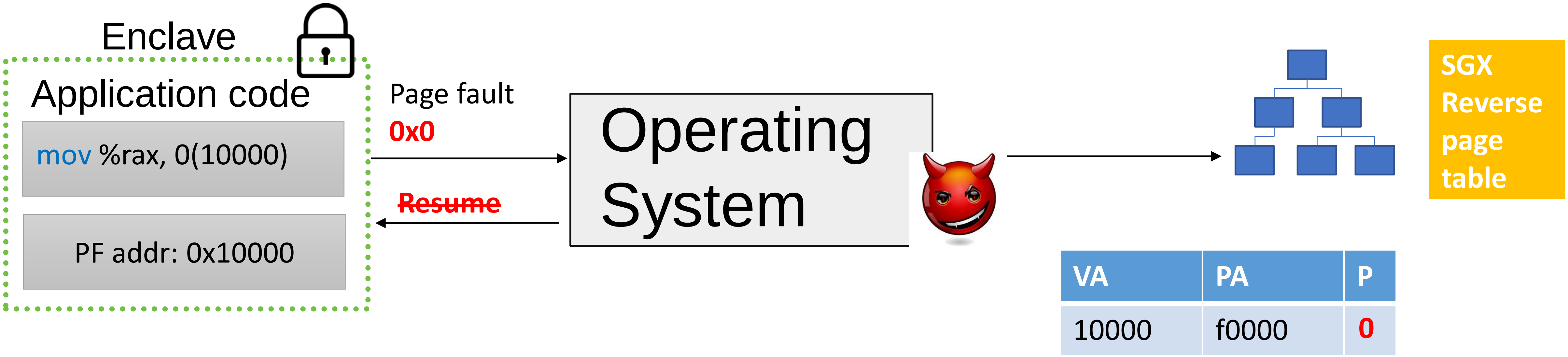
Hide fault information from the OS

Enclave can enforce its own paging policy

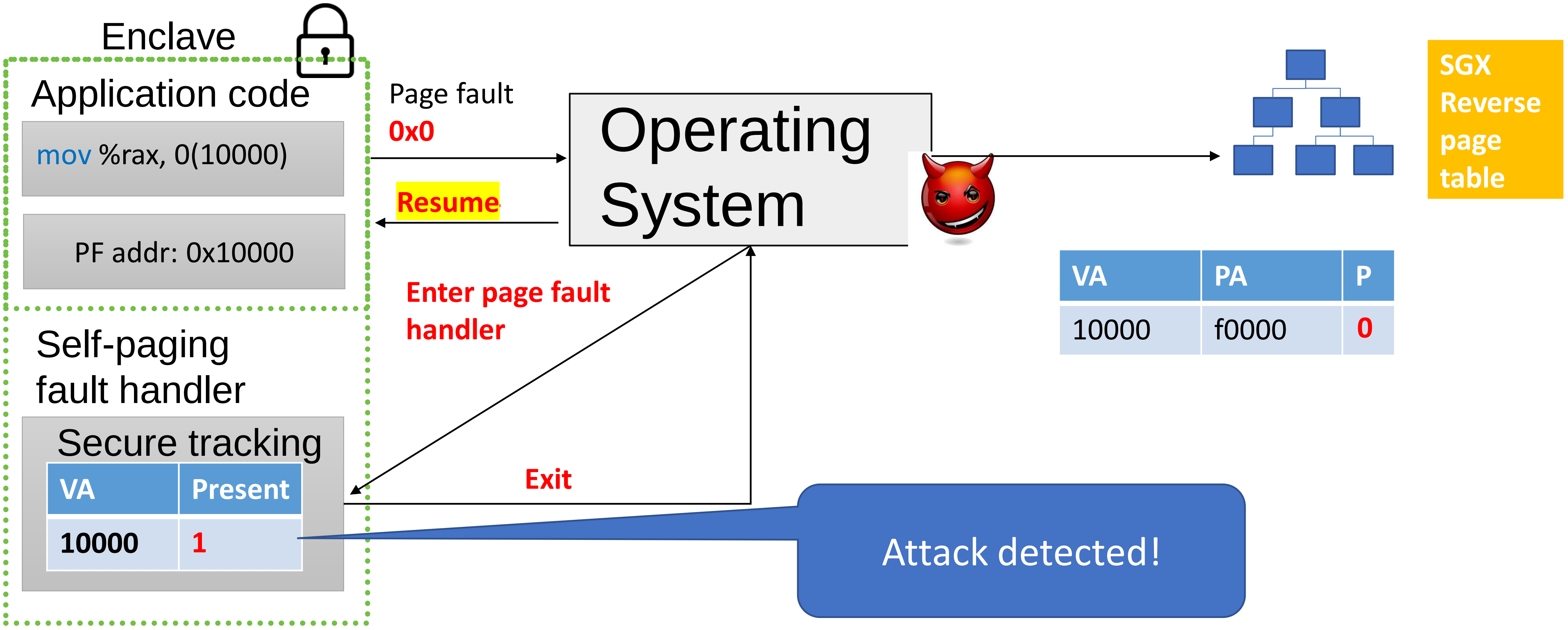
Design overview



Self-Paging Enclaves



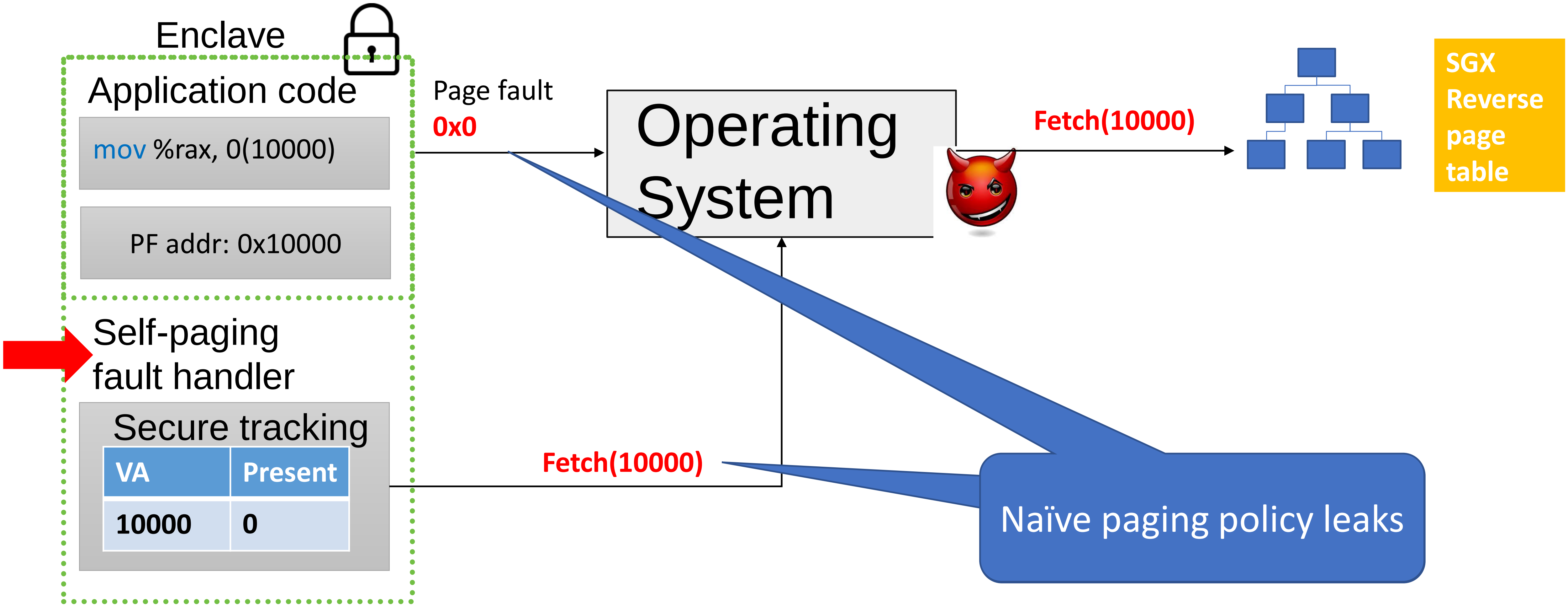
Self-Paging Enclaves



Enclave can protect against spurious page faults

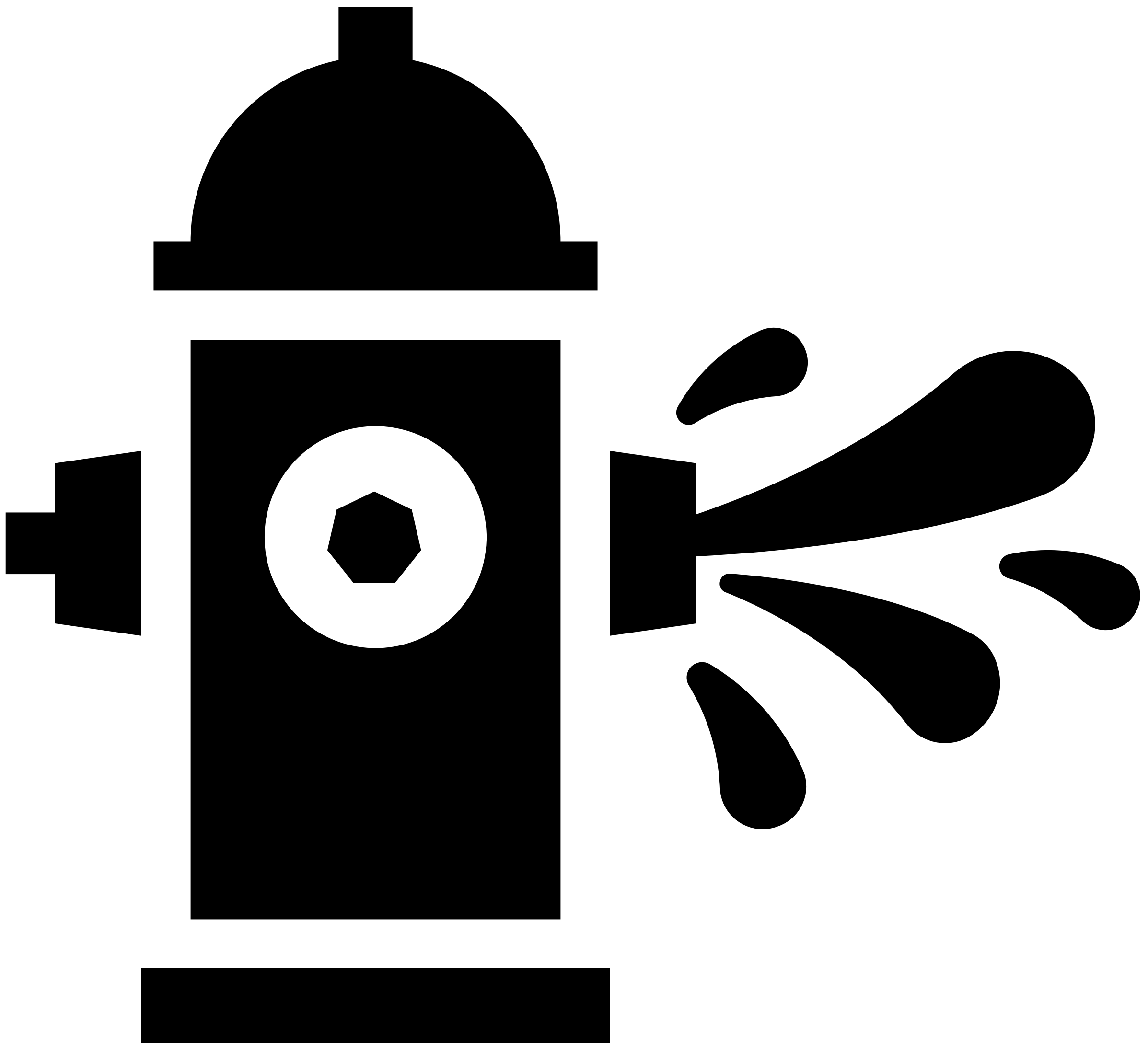
Original attack required millions of page faults.
Removing control is a huge improvement

Support for legitimate page faults



Paging policy: part of the enclave's runtime

Control the leakage

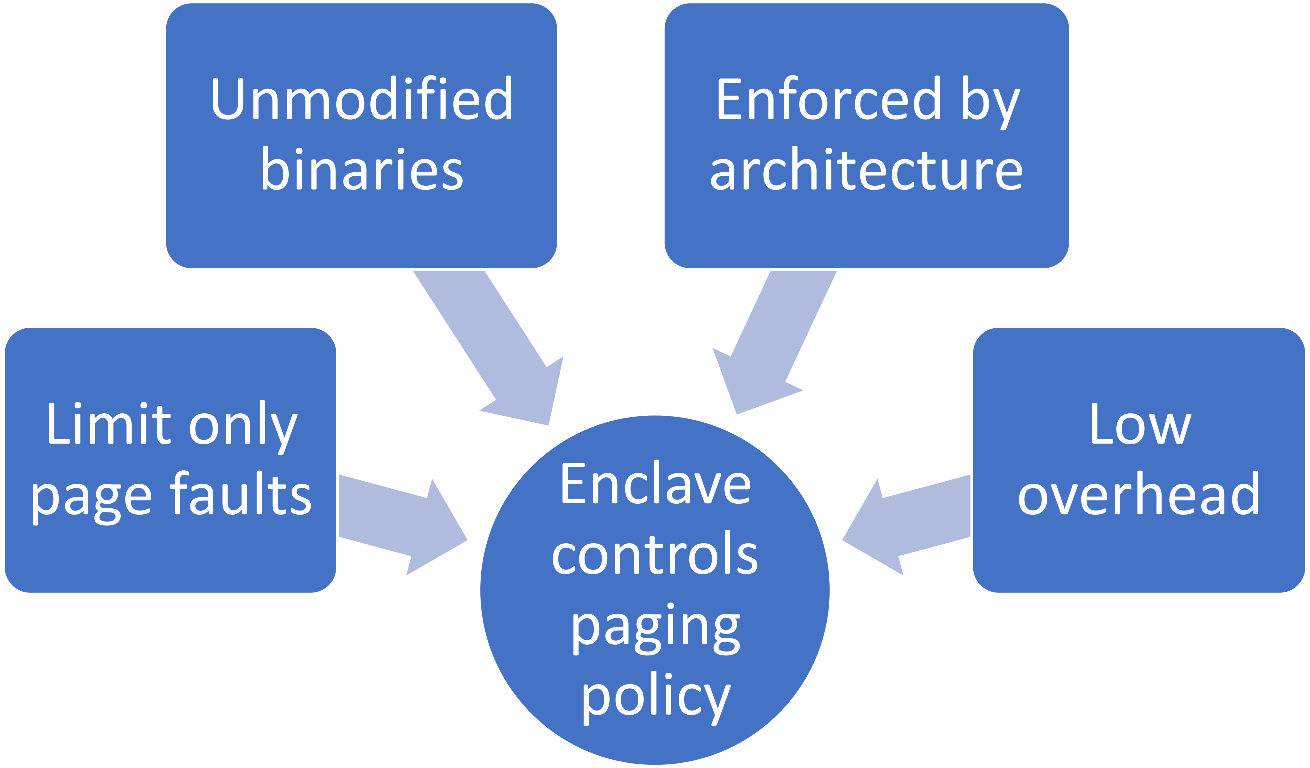


Agenda

- Background
- Controlled-Channel Attack
- Self-Paging Enclaves
- Paging policies
- Evaluation

Rate-limiting policy

- Used by state-of-the-art software mitigations
  - Put a **limit** on the rate of exceptions
  - **Low** security guarantees



[1] Ming-Wei Shih, Sangho Lee, Taesoo Kim, and Marcus Peinado. T-SGX: eradicating controlled-channel attacks against enclave programs. In NDSS'2017.
[2] Oleksii Oleksenko, Bohdan Trach, Robert Krahn, Mark Silberstein, and Christof Fetzer. Varys: Protecting SGX enclaves from practical side-channel attacks. In USENIX ATC'2018.
[3] Sanchuan Chen, Xiaokuan Zhang, Michael K. Reiter, and Yinqian Zhang. Detecting privileged side-channel attacks in shielded execution with Déjá Vu. In Asia CCS'2017.

ORAM policy

- **Provably obfuscates** distribution of memory accesses
- Prior solutions show substantial **performance cost**
- Autarky is order-of-magnitude **faster** and makes it practical
  - Invoke ORAM only for paging

See paper for more details

[1] Sajin Sasy, Sergey Gorbunov, and Christopher W. Fletcher. ZeroTrace: Oblivious memory primitives from Intel SGX. In NDSS'2018.
[2] Meni Orenbach, Yan Michalevsky, Christof Fetzer, and Mark Silberstein. CoSMIX: A compiler-based system for secure memory instrumentation and execution in enclaves. In Usenix ATC'2019.

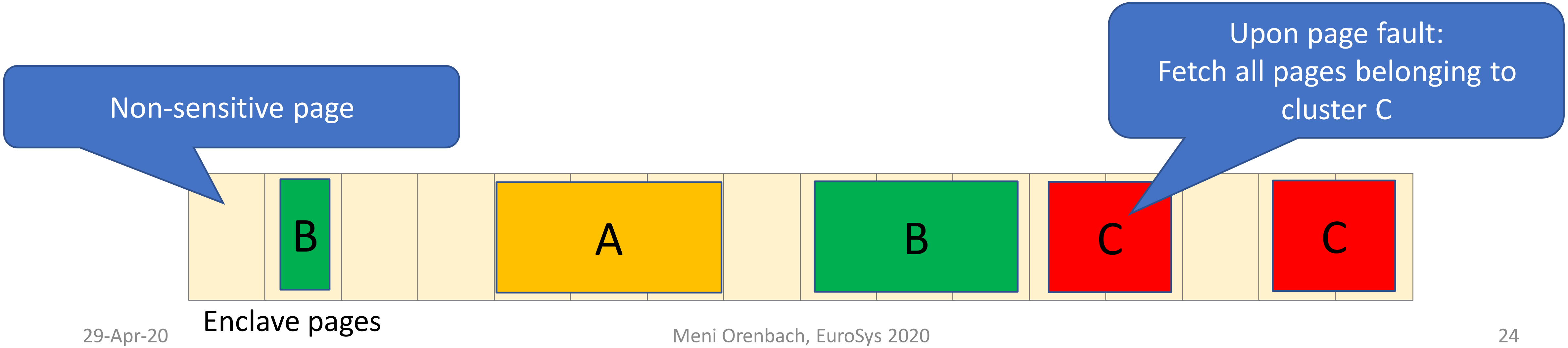
Novel page clusters policy

Some applications **do not** need oblivious paging across **all** pages

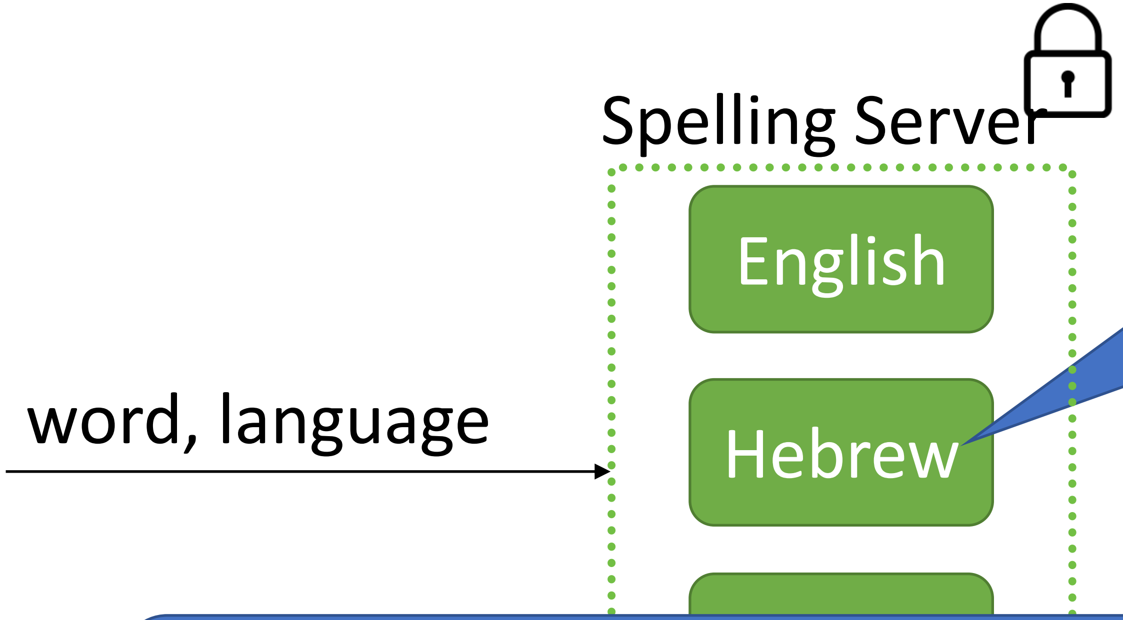
Page clusters: cooperative paging for **all pages** in the **cluster**

Actual faulted address is **hidden** from the OS

Actual page access is **not leaked**



Page clusters policy use cases

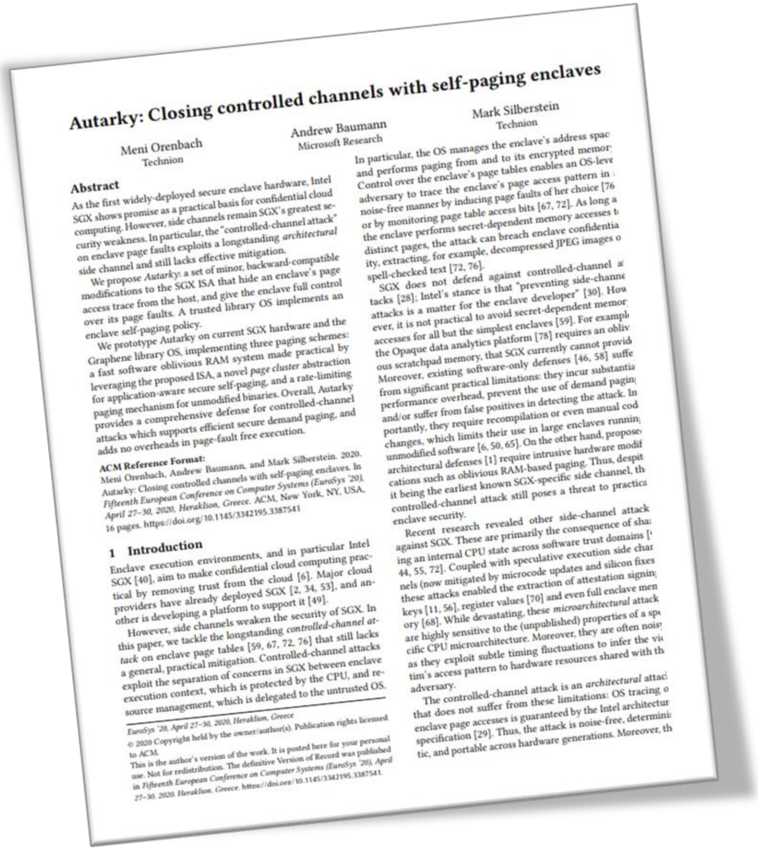


Attacker learns victim access to a dictionary. Not which word queried

Similarly for libraries:
Attacker learns library access, not which function executed.

More details

- SGX1 and SGX2 cooperative paging mechanisms
- Eliminate accessed, dirty bit leakage
- Practical optimizations
-   - Remove extra enclave crossing on page faults
-   - Remove all enclave crossings on page faults



Autarky: Closing controlled channels with self-paging enclaves

Meni Orenbach
Technion

Andrew Baumann
Microsoft Research

Mark Silberstein
Technion

Abstract

As the first widely-deployed secure enclave hardware, Intel SGX shows promise as a practical basis for confidential cloud computing. However, side channels remain SGX's greatest security weakness. In particular, the "controlled-channel attack" on enclave page faults exploits a longstanding *architectural* side channel and still lacks effective mitigation.

We propose *Autarky*: a set of minor, backward-compatible modifications to the SGX ISA that hide an enclave's page access trace from the host, and give the enclave full control over its page faults. A trusted library OS implements an enclave self-paging policy.

We prototype Autarky on current SGX hardware and the Graphene library OS, implementing three paging schemes: a fast software oblivious RAM system made practical by leveraging the proposed ISA, a novel *page cluster* abstraction for application-aware secure self-paging, and a rate-limiting paging mechanism for unmodified binaries. Overall, Autarky provides a comprehensive defense for controlled-channel attacks which supports efficient secure demand paging, and adds no overheads in page-fault free execution.

ACM Reference Format:
Meni Orenbach, Andrew Baumann, and Mark Silberstein. 2020. Autarky: Closing controlled channels with self-paging enclaves. In *Fifteenth European Conference on Computer Systems (EuroSys '20), April 27–30, 2020, Heraklion, Greece.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3342195.3387541

1 Introduction

Enclave execution environments, and in particular Intel SGX [40], aim to make confidential cloud computing practical by removing trust from the cloud [6]. Major cloud providers have already deployed SGX [2, 34, 53], and another is developing a platform to support it [49].

However, side channels weaken the security of SGX. In this paper, we tackle the longstanding *controlled-channel attack* on enclave page tables [59, 67, 72, 76] that still lacks a general, practical mitigation. Controlled-channel attacks exploit the separation of concerns in SGX between enclave execution context, which is protected by the CPU, and resource management, which is delegated to the untrusted OS.

EuroSys '20, April 27–30, 2020, Heraklion, Greece

In particular, the OS manages the enclave's address spac
and performs paging from and to its encrypted memor
Control over the enclave's page tables enables an OS-leve
adversary to trace the enclave's page access pattern in
noise-free manner by inducing page faults of her choice [76
or by monitoring page table access bits [67, 72]. As long a
the enclave performs secret-dependent memory accesses t
distinct pages, the attack can breach enclave confidentia
ity, extracting, for example, decompressed JPEG images o
spell-checked text [72, 76].

SGX does not defend against controlled-channel a
tacks [28]; Intel's stance is that "preventing side-channe
attacks is a matter for the enclave developer" [30]. How
ever, it is not practical to avoid secret-dependent memor
accesses for all but the simplest enclaves [59]. For exampl
the Opaque data analytics platform [78] requires an obliv
ous scratchpad memory, that SGX currently cannot provid
Moreover, existing software-only defenses [46, 58] suffe
from significant practical limitations: they incur substantia
performance overhead, prevent the use of demand pagin
and/or suffer from false positives in detecting the attack. In
portantly, they require recompilation or even manual cod
changes, which limits their use in large enclaves runnin
unmodified software [6, 50, 65]. On the other hand, propose
architectural defenses [1] require intrusive hardware modif
cations such as oblivious RAM-based paging. Thus, despit
it being the earliest known SGX-specific side channel, th
controlled-channel attack still poses a threat to practica
enclave security.

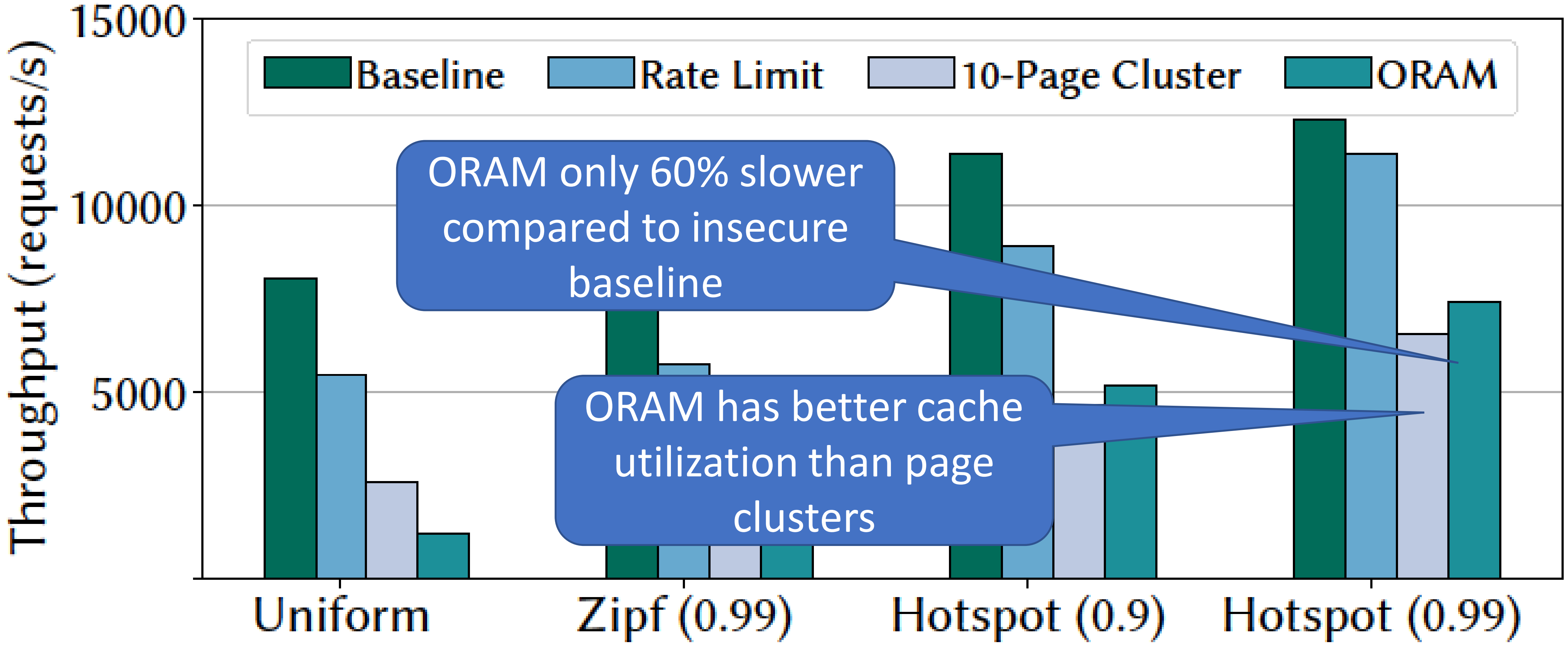
Recent research revealed other side-channel attack
against SGX. These are primarily the consequence of sha
ing an internal CPU state across software trust domains [
44, 55, 72]. Coupled with speculative execution side cha
nels (now mitigated by microcode updates and silicon fixes
these attacks enabled the extraction of attestation signin
keys [11, 56], register values [70] and even full enclave men
ory [68]. While devastating, these *microarchitectural* attack
are highly sensitive to the (unpublished) properties of a sp
cific CPU microarchitecture. Moreover, they are often nois
as they exploit subtle timing fluctuations to infer the vi
tim's access pattern to hardware resources shared with th
adversary.

The controlled-channel attack is an *architectural* attac
that does not suffer from these limitations: OS tracing o
enclave page accesses is guaranteed by the Intel architectur
specification [29]. Thus, the attack is noise-free, determini
tic, and portable across hardware generations. Moreover, th

Agenda

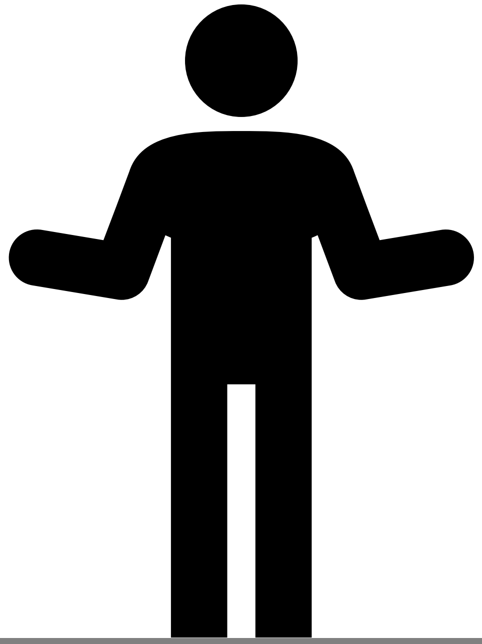
- Background
- Controlled-Channel Attack
- Self-Paging Enclaves
- Evaluation

Memcached stores > 2x available memory
Issuing random 1KB GET requests



Conclusion

- Autarky **mitigates** the controlled-channel attack
  - **Practical modifications** to the architecture
  - Runtime with a secure paging policy
- Maintains **backward compatibility**
  - Operating system
  - Demand-paging
- Attack is not unique to SGX enclaves
  - Retrofit Autarky for other enclave environments!



Thank you!