TU Graz

SCIENCE
PASSION
TECHNOLOGY

TME-Box: Scalable In-Process Isolation through Intel TME-MK Memory Encryption

Martin Unterguggenberger[1] Lukas Lamster[1] David Schrammel[1] Martin Schwarzl[2] Stefan Mangard[1]

[1]Graz University of Technology | [2]Cloudflare, Inc.

NDSS 2025

› isec.tugraz.at

Problem Statement and Contributions

- **Lightweight and efficient isolation for modern cloud settings**

- **TME-Box repurposes Intel TME-MK for in-process isolation**
  - **Fine-grained** and **scalable isolation** on **off-the-shelf x86 machines**
  - **Sandboxes** use designated **encryption keys for memory interactions**
  - **Performance-optimized prototype** showcasing overheads of **5.2 % for data isolation** and **9.7 % for code and data isolation**

Problem Statement and Contributions

- **Lightweight and efficient isolation for modern cloud settings**

- **TME-Box repurposes Intel TME-MK for in-process isolation**
  - **Fine-grained** and **scalable isolation** on **off-the-shelf x86 machines**
  - **Sandboxes** use designated **encryption keys for memory interactions**
  - **Performance-optimized prototype** showcasing overheads of **5.2 % for data isolation** and **9.7 % for code and data isolation**

Motivation

- **Cloud computing is highly optimized for performance and efficiency**
  - Replace **process isolation** with **in-process sandboxes**

- **Exclusion of process isolation introduces security risks**
  - **Memory safety errors** allow to **leak private data**
    - **Heartbleed**[1] and **Cloudbleed**[2] vulnerabilities

Motivation

- **Cloud computing is highly optimized for performance and efficiency**
  - Replace **process isolation** with **in-process sandboxes**

- **Exclusion of process isolation introduces security risks**
  - **Memory safety errors** allow to **leak private data**
    - **Heartbleed**[1] and **Cloudbleed**[2] vulnerabilities

TME-Box Design

Design Overview

- Confidential computing **brings new hardware features** to **x86 servers**
- Repurpose **Intel memory encryption** for **in-process isolation**

Design Goals

- **Scalable isolation** and **flexible data relocation**
- **High number of sandboxes** through **encryption keys**
- **Hardware-assisted isolation** on commodity **x86 hardware**

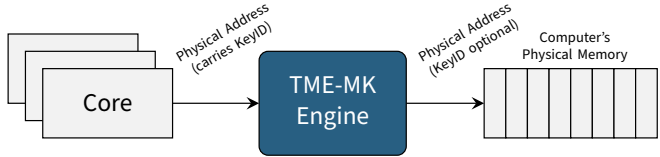
Design Overview

- Confidential computing **brings new hardware features** to **x86 servers**
- Repurpose **Intel memory encryption** for **in-process isolation**

- **Design Goals**
  - **Scalable isolation** and **flexible data relocation**
  - **High number of sandboxes** through **encryption keys**
  - **Hardware-assisted isolation** on commodity **x86 hardware**

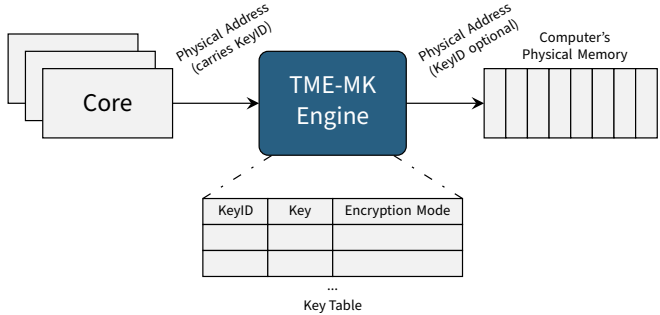
Intel Architectural Memory Encryption

- **Intel Total Memory Encryption Multi-Key (TME-MK)**[3]
  - **Encryption engine** located in memory controller



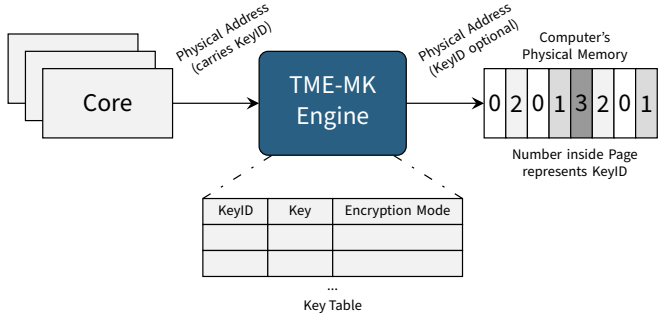
Intel Architectural Memory Encryption

- **Intel Total Memory Encryption Multi-Key (TME-MK)**[3]
  - Specified for **up to 32K encryption keys**



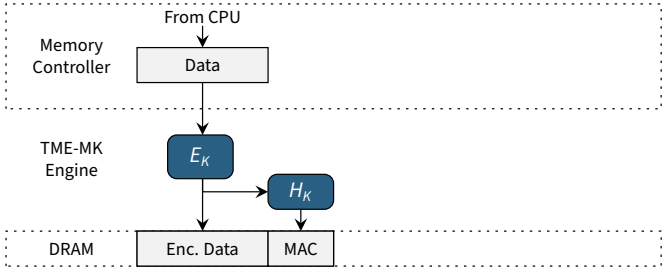
Intel Architectural Memory Encryption

- **Intel Total Memory Encryption Multi-Key (TME-MK)**[3]
  - **Page-granular encryption** of physical memory



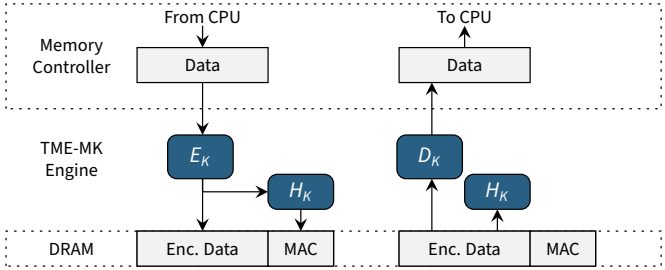
Authenticated Encryption

- **Intel TME-MK with cryptographic-integrity**[4]
  - **Intel TDX** integrates support for **authenticated encryption**



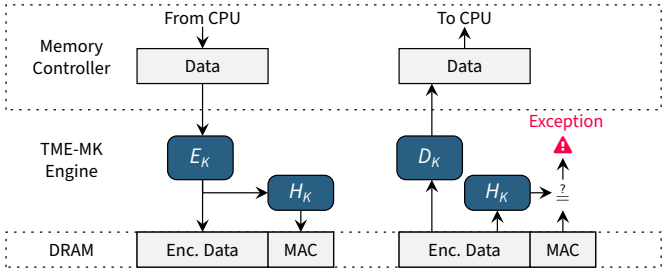
Authenticated Encryption

- **Intel TME-MK with cryptographic-integrity**[4]
  - Data encryption utilizes **AES XTS** with **SHA-3 MAC**



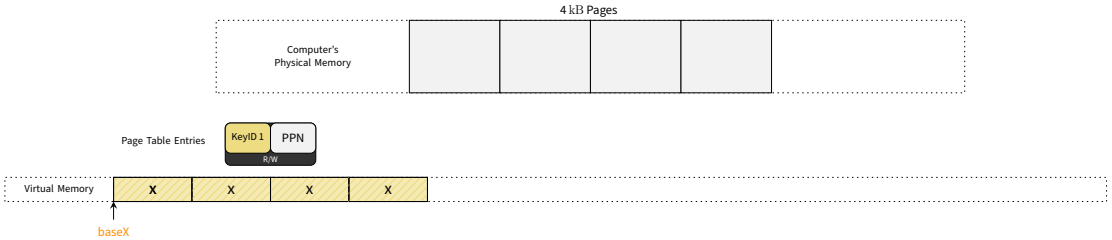
Authenticated Encryption

- **Intel TME-MK with cryptographic-integrity**[4]
  - ⚠ **Integrity enforcement** through a **28-bit MAC**



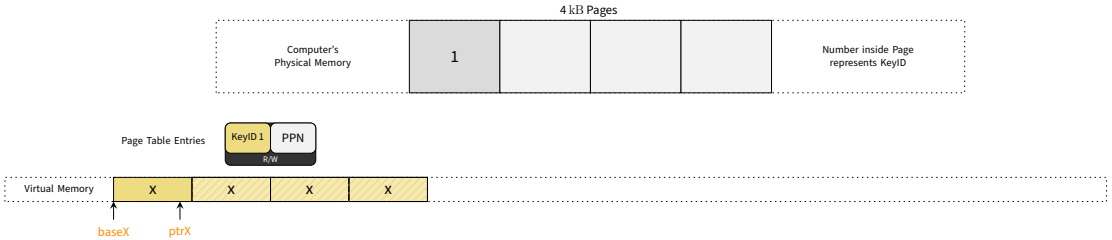
TME-Box Sandbox

- **Memory interactions use sandbox's designated encryption key**
  - **Sandbox-specific keyID** maps to **sandbox's encryption key**



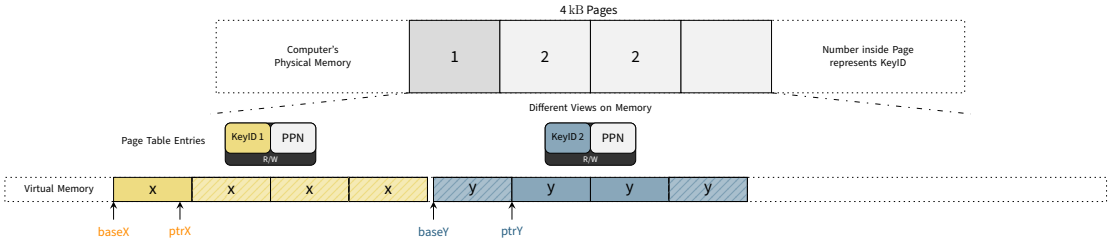
TME-Box Sandbox

- **Memory interactions use sandbox's designated encryption key**
  - Compiler controls **base address and index** of memory operations



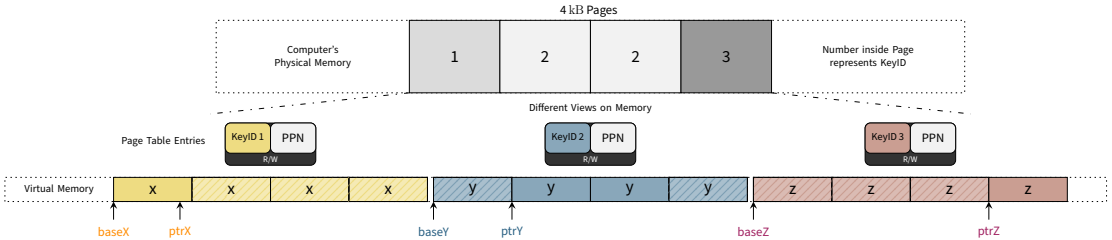
TME-Box Sandbox

- **Memory interactions use sandbox's designated encryption key**
  - Sandboxes **share underlying physical memory**



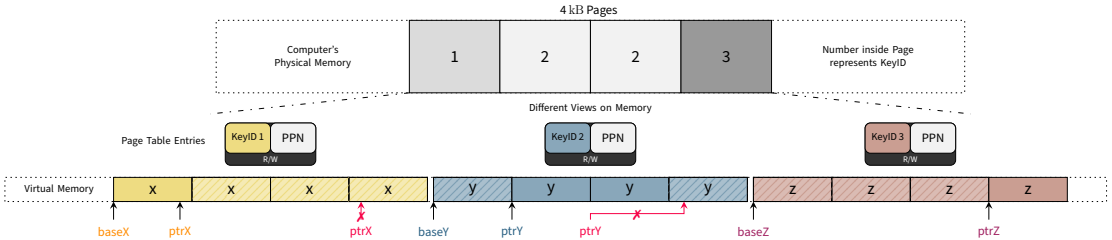
TME-Box Sandbox

- **Memory interactions use sandbox's designated encryption key**
  - **Three sandboxes** mapped to **four physical pages**



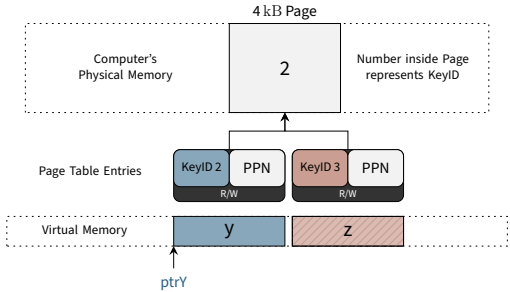
TME-Box Sandbox

- Memory interactions use sandbox's designated encryption key
  - Data integrity detects unauthorized sandbox access



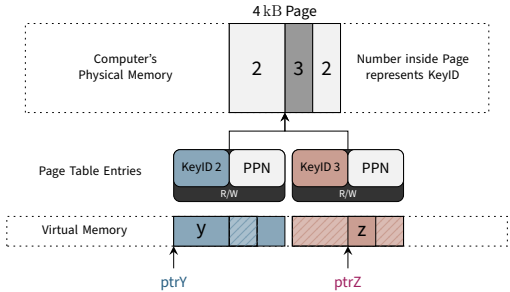
Scalable Memory Isolation

- **Sub-page granular encryption**
  - **Page aliasing** enables **fine-grained encryption**



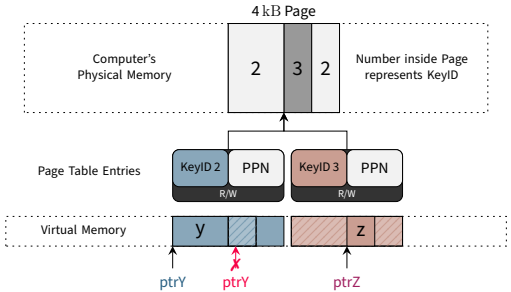
Scalable Memory Isolation

- **Sub-page granular encryption**
  - Parts of **physical page encrypted differently**
  - **TME-MK integrity** mode is **cache line-sized**



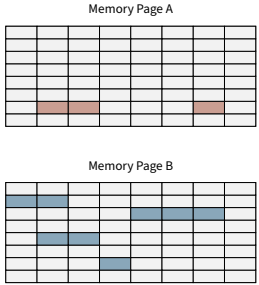
Scalable Memory Isolation

- **Sub-page granular encryption**
  - **Memory initialized** with **keyID** (encryption key)
  - Detect **unauthorized access** through **integrity exception**



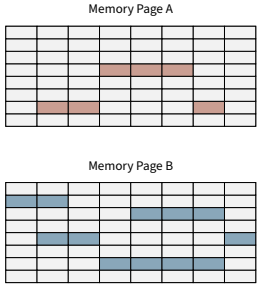
Flexible Memory Management

- **Flexible memory management for the allocator**
  - **Scalable isolation** from single **cache lines** to **full pages**



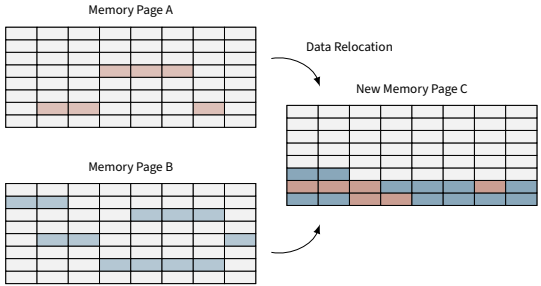
Flexible Memory Management

- **Flexible memory management for the allocator**
  - **Efficient management** of **fragmented memory resources**



Flexible Memory Management

- Flexible memory management for the allocator
  - Relocation of data in memory



Implementation and Evaluation

TME-Box Framework

LLVM extension

Memory allocator

Kernel support

TME-Box Framework

 **LLVM extension**

 **Memory allocator**

 **Kernel support**

Reserve **CPU register** (r15; gs register)

Instrument **memory operations**

Instrument **control-flow transfers**

TME-Box Framework

LLVM extension

- CPU register
- Memory operations
- Control-flow transfers

Memory allocator

Kernel support

Initialize and **manage memory** of runtime data

TME-Box Framework

LLVM extension

- **CPU register**
- **Memory operations**
- **Control-flow transfers**

Memory allocator

- **Initialize memory**
- **Memory management**

Kernel support

Syscall interface to assign **keyIDs to pages**

TME-Box Framework

LLVM extension

- CPU register
- Memory operations
- Control-flow transfers

Memory allocator

- Initialize memory
- Memory management

Kernel support

- Syscall interface

Performance Evaluation

Evaluate SPEC CPU2017 benchmark suite

- **Performance overhead**
  - Compiler instrumentation
  - Memory initialization
  - Memory encryption
  - Memory aliasing

Performance Evaluation

Evaluate SPEC CPU2017 benchmark suite

- **Performance overhead**
  - **Compiler instrumentation**
  - Memory initialization
  - Memory encryption
  - Memory aliasing

control **base and index**

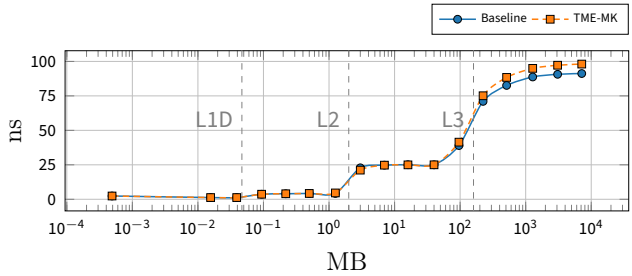
x86 segment-based addressing

restrict **control-flow transfers**

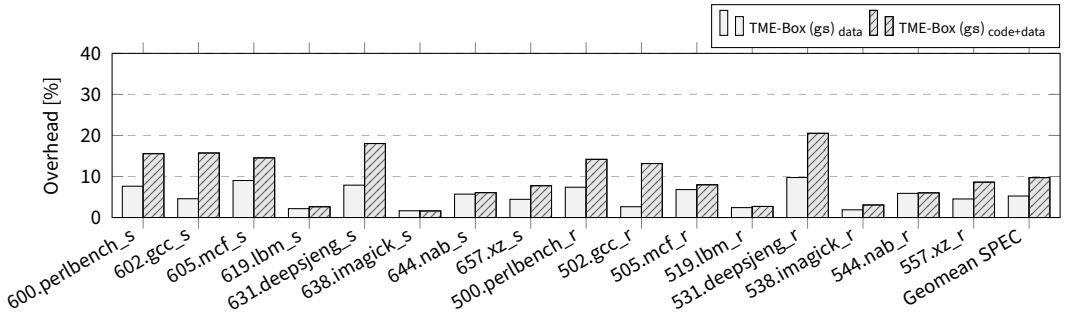
Performance Evaluation

Evaluate SPEC CPU2017 benchmark suite

- **Performance overhead**
  - Compiler instrumentation
  - Memory initialization
  - **Memory encryption**
  - Memory aliasing

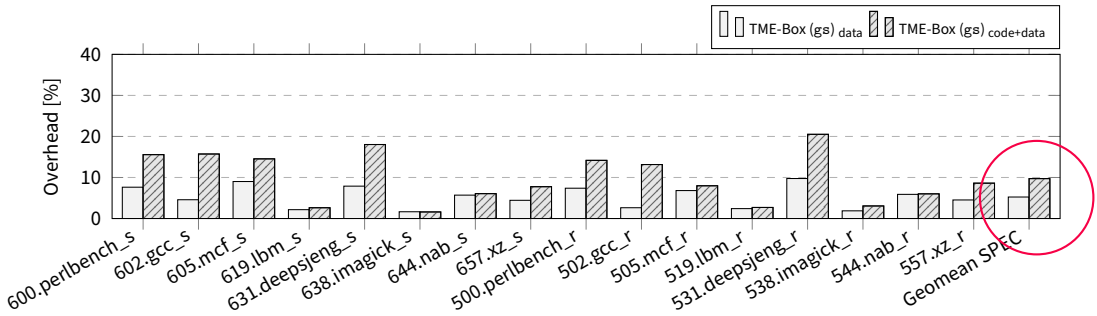


Performance Overhead of TME-Box in gs-Mode



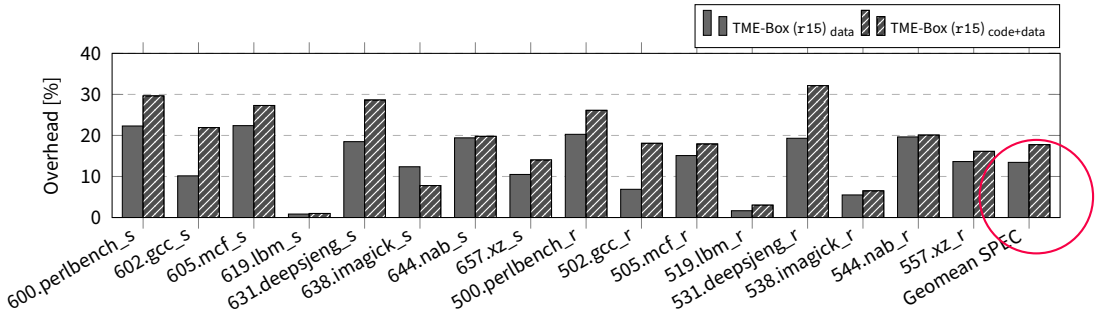
Geomean overhead of **5.2 % (data)** and **9.7 % (code+data)**

Performance Overhead of TME-Box in gs-Mode



- Geomean overhead of **5.2 % (data)** and **9.7 % (code+data)**

Performance Overhead of TME-Box in r15-Mode



- Geomean overhead of **13.4 % (data)** and **17.7 % (code+data)**

Conclusion

TME-Box repurposes Intel TME-MK for in-process isolation

- **Hardware-assisted sandboxing** through **encryption**
- **Scalable isolation** from individual **cache lines** to **full pages**
- **Flexible memory management** for the allocator
- **Isolation** of **up to 32K sandboxes** (encryption keys)
- Available on **off-the-shelf Intel x86 machines**

TU Graz

SCIENCE
PASSION
TECHNOLOGY

TME-Box: Scalable In-Process Isolation through Intel TME-MK Memory Encryption

Martin Unterguggenberger[1] Lukas Lamster[1] David Schrammel[1] Martin Schwarzl[2] Stefan Mangard[1]

[1]Graz University of Technology | [2]Cloudflare, Inc.

NDSS 2025

› isec.tugraz.at

Bibliography I

[1] Zakir Durumeric, Frank Li, James Kasten, Johanna Amann, Jethro Beekman, Mathias Payer, Nicolas Weaver, David Adrian, Vern Paxson, Michael Bailey, and J. Alex Halderman. **The Matter of Heartbleed**. IMC. 2014.

[2] John Graham-Cumming. **Incident report on memory leak caused by Cloudflare parser bug**. https://blog.cloudflare.com/incident-report-on-memory-leak-caused-by-cloudflare-parser-bug. Accessed: 2024-06-10. 2017.

[3] Intel. **Intel Architecture Memory Encryption Technologies**. https://www.intel.com/content/www/us/en/content-details/679154/intel-architecture-memory-encryption-technologies-specification.html. Revision 1.4, Accessed: 2023-01-31. 2022.

[4] Intel. **Intel Trust Domain Extensions**. https://cdrdv2-public.intel.com/690419/TDX-Whitepaper-February2022.pdf. Accessed: 2024-05-27. 2022.