

SHELTER: Extending Arm CCA with Isolation in User Space

Yiming Zhang^{1,2}, Yuxin Hu¹, Zhenyu Ning^{3,1}, <u>Fengwei Zhang</u>¹, Xiapu Luo², Haoyang Huang¹, Shoumeng Yan⁴, Zhengyu He⁴

1Southern University of Science and Technology, 2The Hong Kong Polytechnic University, 3Hunan University, 4Ant Group



COMPuter And System Security Lab

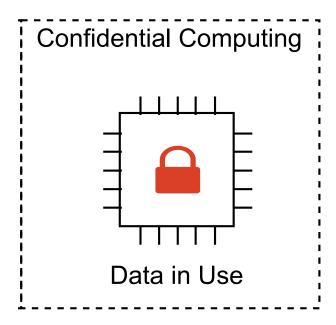




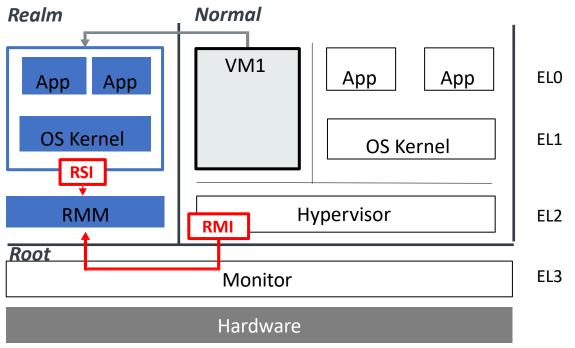


Confidential Computing

- Hardware-assisted security design
- Cloud and Edge devices
- Intel TDX, AMD SEV, Arm CCA



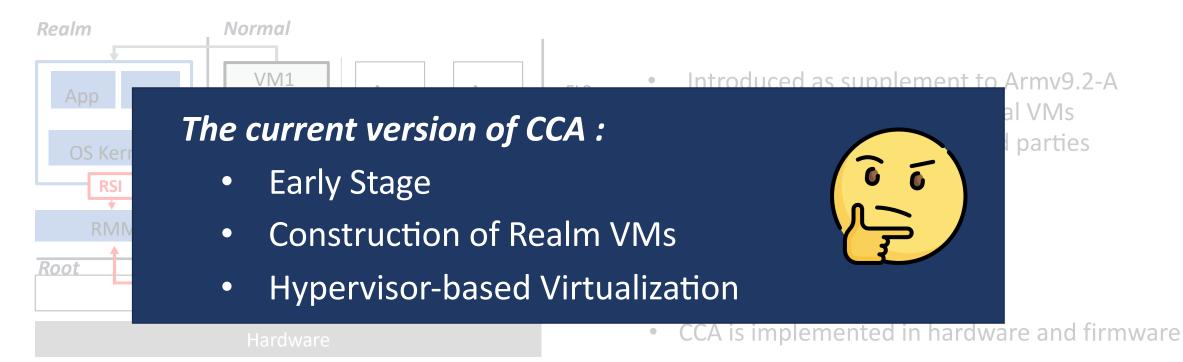
Arm Confidential Compute Architecture (CCA)



- Introduced as supplement to Armv9.2-A
- Two added additional worlds
 - Secure -> Secure & EL3 Root
 - Normal -> Normal & Realm
- CCA is implemented in hardware and firmware

RME: Realm Management Extension RMM: Realm Management Monitor RMI: Realm Management Interface RSI: Realm Services Interface

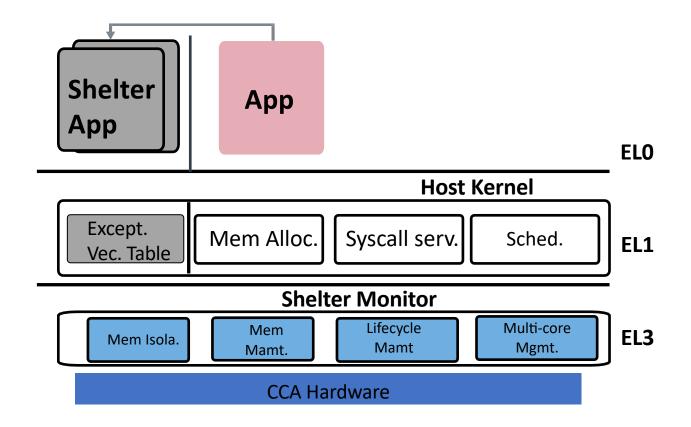
Arm Confidential Compute Architecture (CCA)



RME: Realm Management Extension RMM: Realm Management Monitor RMI: Realm Management Interface RSI: Realm Services Interface

Motivation

- Cooperating with CCA hardware to provide user-level isolation
- Complement to CCA's Realm VM architecture

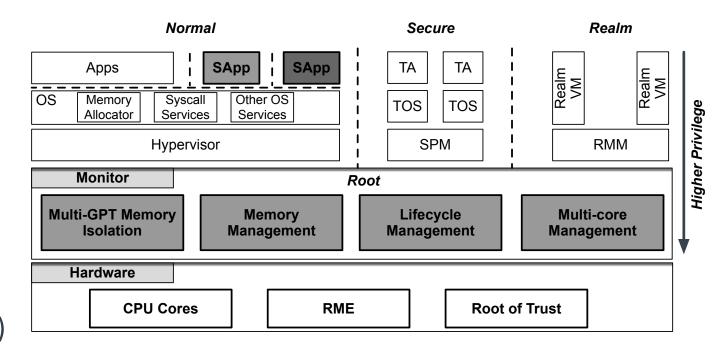


Threat Model & Assumptions

- An attacker can compromise Host OS, hypervisor, or privileged software in Secure, and Realm world (e.g., SPM or RMM)
- The Monitor is trusted and the hardware is correctly implemented
- Physical/Side-channel/denial-of-service attacks are out of scope
- Assuming remote attestation support and secure boot

Shelter

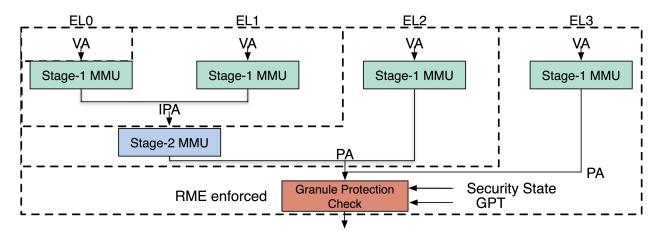
- SHELTER App (SApp)
 - Running on Normal World ELO
- Host OS
 - Non-security responsibilities
- Shelter Monitor
 - In Root world
 - Security responsibilities
- CCA hardware feature
 - Realm Management Extension (RME)



Granule Protection Check (GPC)

- RME enforced isolation is managed through a new Granule Protection Table (GPT)
- GPT is controlled by the Monitor in EL3
- GPT specifies what physical address spaces (PAS) a memory page belongs to

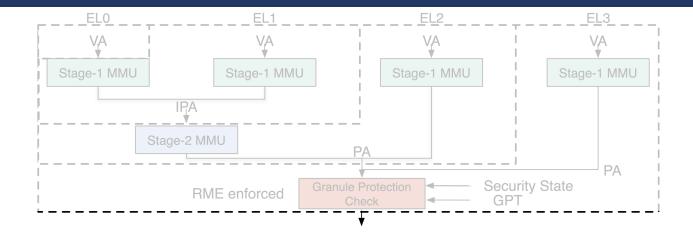
Security state	Normal PAS	Secure PAS	Realm PAS	Root PAS
Normal	✓	X	×	×
Secure	\checkmark	\checkmark	×	×
Realm	\checkmark	×	\checkmark	×
Root	\checkmark	\checkmark	\checkmark	\checkmark



Granule Protection Check (GPC)

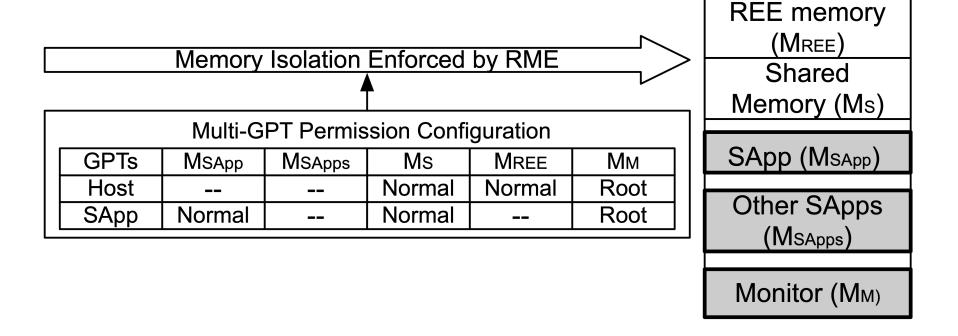
- RME enforced isolation is managed through a new Granule Protection Table (GPT)
- GPT is controlled by the Monitor in EL3
- GPT specifies what physical address spaces (PAS) a memory page belongs to

It is not satisfied with the goal of isolating memory between SApps and other privileged software in Normal, Secure, and Realm world.



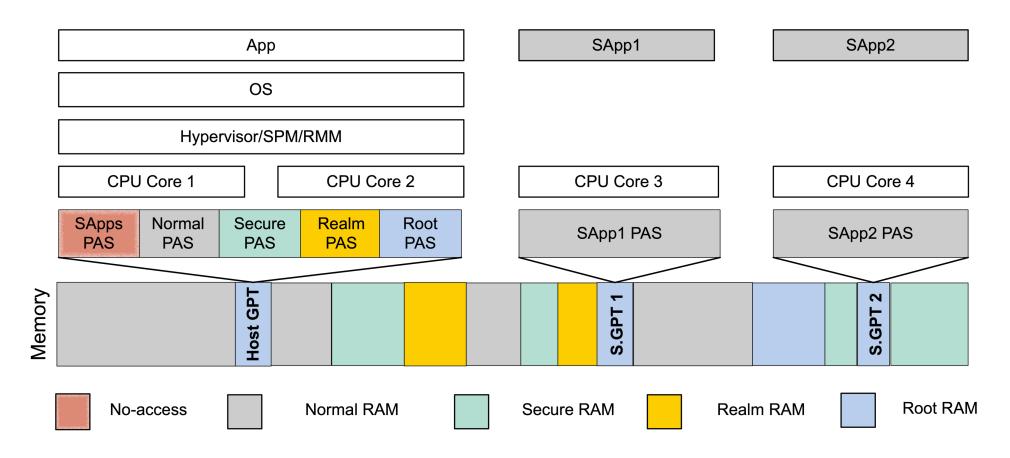
Multi-GPT Memory Isolation

- Maintain multiple GPTs in EL3 Monitor
- Divide the physical address space (PAS) for different programs



Multi-GPT Memory Isolation

Establishing address-space-per-core for each SApp and other code region



Multi-GPT Memory Isolation

• The Monitor dynamically controls the access permissions of different programs

Performance Optimization

- New GPT construction causes long startup latency for SApps
 - Root cause: Shelter needs to add granule information containing a layout of the entire main memory for the new GPT and measure each GPT entry

Performance Optimization

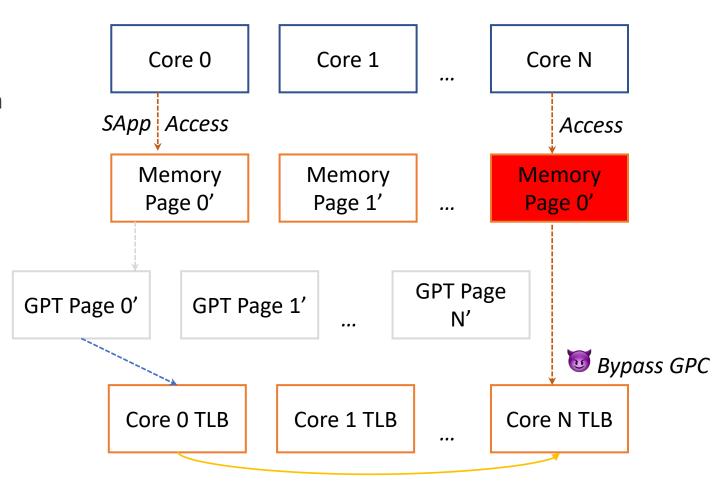
- New GPT construction causes long startup latency for SApps
 - Root cause: Shelter needs to add granule information containing a layout of the entire main memory for the new GPT and measure each GPT entry

Multi-GPT Management							
GPTs	MSApp	MSApps	Ms	MREE	Мм		
Host		-	Normal	Normal	Root		
SApp	Normal		Normal		Root		
Copy to create new SApp GPT							
<u> </u>							
Template GPT							

^{*}Using shadow GPT, a template with copy and update to speed up SApp creation

TLB-based GPT attack

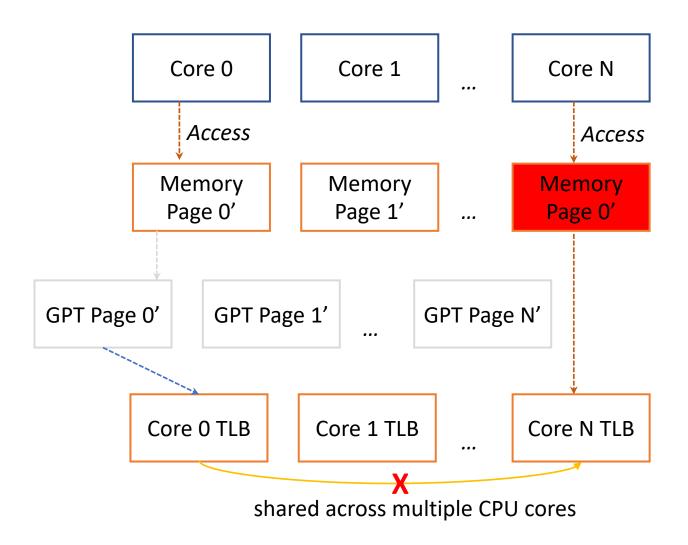
- GPT entries are permitted to be cached in TLB as part of TLB entry
- GPT information in a TLB is permitted to be shared across multiple CPU cores



Different GPTs are shared across CPU cores

Defend against TLB-based GPT attacks

- TLB invalidation during switches and GPT modifications
- Disable the shareable property of TLB



Some Execution Features

- Memory management
 - Contiguous physical memory pool
 - Ensure multiple SApps do not have memory overlap
 - SApp Page table is isolated
- Syscall & lago attack checks
- Interrupt & Signal
- Multi-threaded synchronization primitive

SHELTER: Extending Arm CCA with Isolation in User Space

Yiming Zhang^{1,2,3,*}, Yuxin Hu^{1,2,*}, Zhenyu Ning^{4,1}, Fengwei Zhang^{2,1,†}, Xiapu Luo³, Haoyang Huang^{1,2}, Shoumeng Yan⁵, Zhengyu He⁵

¹Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology ²Department of Computer Science and Engineering, Southern University of Science and Technology ³Department of Computing, The Hong Kong Polytechnic University ⁴College of Computer Science and Electronic Engineering, Hunan University ⁵Aut Group

Abstrac

The increasing adoption of confidential computing is providing individual users with a more seamless interaction with numerous mobile and server devices. TrustZone is a promising security technology for the use of partitioning sensitive private data into a trusted execution case. THE, Unfortunately, third-party develop to TrustZone. This is becauch security applicatif Moreover, TrustZo ties affecting Truthe entire system.

the entire system Advanced ' cently new o (CCA) creat world for co dentiality a ily targets provide use we present S mary Realm party develope in userspace. SH CCA hardware pi hardware-based isol ware workloads to trust privileged software (e.g., tru sor). We have implemented and eva results demonstrated that SHELTER guarantees the security of applications with a modest performance overhead (<15%) on real-world workloads

1 Introduction

The increasing adoption of confidential computing is provid-

devices [14]. Meanwhile, as vast numbers of devices are being widely deployed and connected, a host of new security unterabilities and attacks are breaking out [33]. It is critical that these devices provide a high level of security and privacy to protect sensitive data. On Arm platforms, TrustZone [26] supports such an ability that enforces system-wide isolation using two different physical address spaces (PAS) named Normal world and Secure world for untrusted and trusted software, expectively.

ough TrustZone enables systems to protect sensitive the TEE, there still exist two major limitations to i) Third-party developers have limited accessibility ne. This is because TEE vendors need to rigorte such security applications to prevent the de-Trusted Applications (TA) that may import exnerabilities [11]. These processes increase the for deploying new TAs, conflicting with the et trend of computing services [46]. (ii) The e for commercial TrustZone-based systems is e there are increasing vulnerabilities affecting sted OSes, according to recent studies [33, 34]. defense mechanism based on privilege division rchitecture called Exception Levels (EL0-EL3). For e, in the Secure world, Secure Exception Level 0 (i.e., runs TAs, S.EL1 runs the trusted OS, and S.EL3 runs e. However, once a vulnerability affecting ploited, the entire TrustZone-based sysbe compromised [33].

An d a new system called Confidential Compu. 123] to protect data in use on armv9.2. 123] to protect data in use on tation in a new PAS named code and data from access or modification, 124] like a instantiate multiple Received by a new hardware primits. 123 to 2 alled Real Management Ex-

Security Analysis

Adversary Subject	Main Attacks	Defense
OS/Hypervisor	Unauthorized memory access Invalid mapping or return value Illegal GPT modification GPC circumvention	0 2 36 5
SApp	SApp abuse	00
TLB/Cache Untended GPT sharing in Unauthorized cache access EL3 code cache injection		0 0 0
Peripherals Malicious DMA		0

• Multi-GPT isolation enforced by GPC; • Monitor checks (e.g., ensuring no memory overlap between SApps, checking syscall return value, verifying validity of the runtime); • Multi-core synchronization; • Microarchitectural Maintenance; • Monitor Maintaining in the highest privilege.

Shelter Implementation

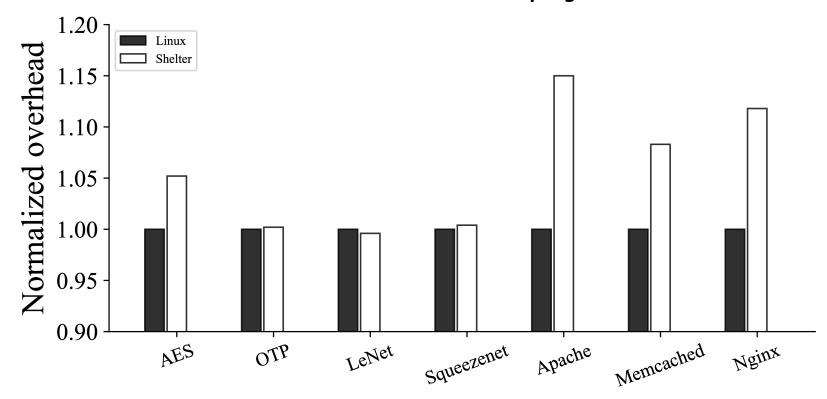
- Functional prototype implementation
 - FVP Base RevC-2xAEMvA with RME-enabled features
 - TCB: ATF with 2k SLoCs additions
- Official CCA software stacks
 - TCB: ATF + TF-RMM (released date 2022/11/09)
 - TF-RMM(v0.2.0) is around 8.2k SLoCs
- TCB comparison with CCA
 - 2k vs 8k SLoCs

Performance Evaluation

- No commercial hardware supporting CCA is available on the market
 - FVP Simulator is not cycle accurate
- GPT-analogue in Armv8-A Juno Board
 - Mimic all GPT in-memory operations
 - Replace the GPT-related registers with idle EL3 registers
 - Invalidate all TLBs instead of TLB GPT invalidation instructions (e.g., TLBI PAALLOS)
 - The other functionality are the same as those on the FVP

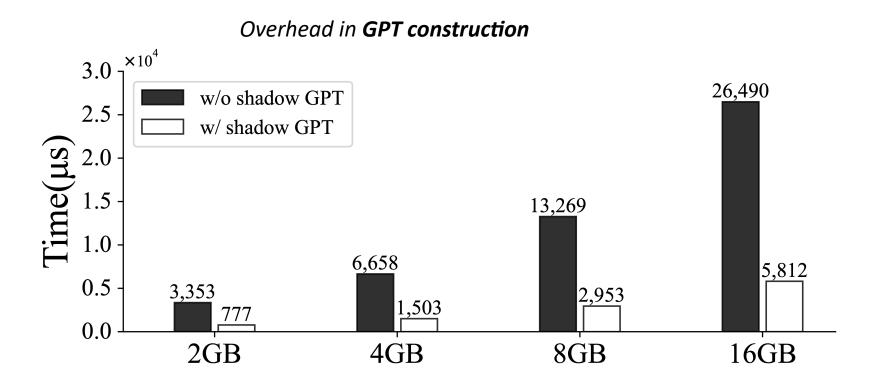
Application Benchmarks

Runtime Overhead on real-world programs



SHELTER incurs <15% runtime-overhead on real-world workloads compared with Linux

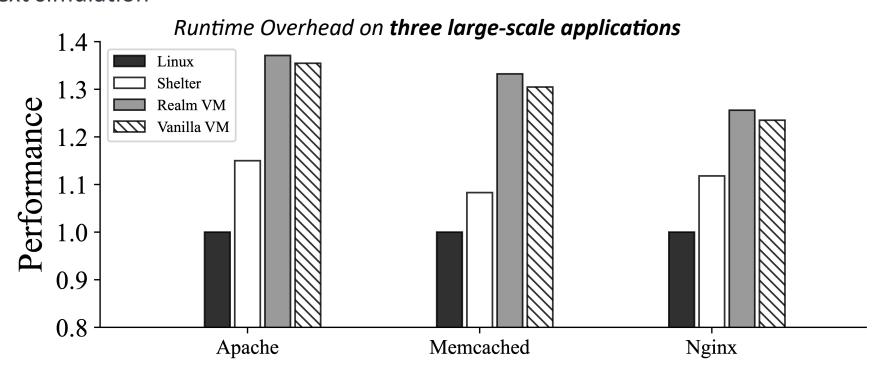
Performance Optimization



✓ With shadow GPT, reducing overhead on average of 77.5% in SApp Creation

Comparison with CCA's VM-based approach

 A basic CCA VM-based performance prototype with same GPT-analogue methodology and a Realm-context simulation



✓ Avg. SHELTER 11.7% vs CCA Realm VM 32.0%

Conclusion

- Shelter leverages CCA hardware for a new creation of user-level isolated environment
 - complementary to CCA's primary Realm VM-style architecture
 - A smaller TCB
 - Lower performance overhead
 - No hardware modification for compatible platforms, including mobile and server
- Open Source
 - https://github.com/Compass-All/Shelter





Thanks for listening! Q & A zhangfw@sustech.edu.cn