Advancing Confidential Computing on RISC-V with the Memory Protection Table

Stefano Mercogliano, Valerio Di Domenico and Alessandro Cilardo

Department of Electrical Engineering and Information Technologies
University of Naples Federico II

Abstract

Cloud computing presents significant security challenges, especially in multi-tenant environments where ensuring data confidentiality and integrity is crucial. To address this, commercial architectures like Intel TDX, AMD SEV, and ARM RME have introduced Confidential Virtual Machine (CVM) technologies to enable trusted execution. Similarly, RISC-V is developing its own confidential computing framework, the Confidential Virtualization Extension (CoVE). To accelerate the growth of a trusted computing ecosystem on RISC-V, we introduce an open-source implementation of the Memory Protection Table (MPT); a key hardware component that enforces secure supervisor domain isolation. Designed as a stand-alone module, the MPT has been tested both in isolation and integrated into the CVA6 processor's MMU.

Introduction

In the era of big data, cloud computing has become increasingly popular, allowing users to offload computations to remote machines for tasks like financial analysis, machine learning, and bioinformatics. However, shared infrastructure raises security and trust concerns, as tenants may require strict data protection. *Trusted Computing* (TC) addresses these issues through hardware-based security mechanisms, including hardware roots of trust, system-on-chips, and processor extensions, ensuring integrity, confidentiality, and authenticity in cloud environments. Trusted Computing is not a new concept, as it has been evolving for years with technologies like Intel SGX and ARM TrustZone. Recently, a major advancement has been the *Confidential Virtual Machines* (CVMs) paradigm, pioneered by AMD's Secure Encrypted Virtualization (SEV) extension family. CVMs create isolated, cryptographically protected environments as *Trusted Execution Environments* (TEEs), managed by hypervisors that may or may not be trusted. Intel responded with Trust Domain Extensions (TDX), introducing a dedicated module that secures CVMs against untrusted hypervisors. ARM's Realm Management Extension (RME) in Armv9-A, combined with the Confidential Compute Architecture (CCA), establishes isolated "realms" to enhance security in multi-tenant environments. While these technologies represent the cutting edge of cloud security, they are still evolving and not yet fully stable, making it difficult to determine which will ultimately prove to be the most effective. In the meanwhile, the RISC-V community has been actively working toward enhancing security through CVM support. The current baseline specification, known as the *Confidential Virtualization Extension* (CoVE) for the RISC-V application profile [1], aims to establish core APIs that a trusted codebase comprising the *Trusted Security Monitor* (TSM) and its driver (the *TSM driver*), must support for managing CVMs. Unlike other architectures, CoVE does not require modifications to the instruction set architecture (ISA), making it a flexible and scalable approach. It defines multiple deployment models, catering to a wide range of security needs, from embedded systems with strict security requirements to high-assurance cloud computing environments. However, to fully realize CoVE's potential, additional orthogonal extensions are necessary to reinforce isolation, confidentiality, and integrity for CVMs. While the CoVE specification is in an advanced stage, practical implementation and technical development for confidential computing on RISC-V remain limited, highlighting the need for further research and engineering efforts in this area.

Our Goal is to advance RISC-V trusted computing by open-sourcing the fundamental physical support for CoVE deployment models: the *Memory Protection Table* (MPT), as specified in the Smmpt document. The MPT enables RISC-V CPUs to support both trusted and untrusted hypervisors, thereby reducing the trusted computing base (TCB). Additionally, it allows for multiple hypervisors to coexist while ensuring strong isolation by protecting physical memory pages. In this paper, we briefly discuss the SystemVerilog design and implementation of the MPT, along with its ongoing integration into the H-extension capable CVA6 processor [2]. Our work aims to provide a secure and efficient foundation for deploying Confidential Virtual Machines (CVMs) in RISC-V-based architectures, further strengthening the ecosystem of open-source trusted computing.

The Memory Protection Table

The RISC-V privileged architecture defines a supervisor mode (S-mode) capable of running an operating system or a hypervisor with unrestricted access to physical memory; such a software, along with its stack, forms a *supervisor domain*. However, in multi-tenant environments, consolidating the entire TCB within a single supervisor domain significantly expands the attack surface. To mitigate this risk, it is preferable to allow tenants to selectively access or share physical memory pages, ensuring that a CVM and its TSM use only the memory they require. This approach (a) enforces the principle of minimal privilege and (b) reduces the TCB for each CVM. Supervisor domain isolation can be achieved using the Physical Memory Protection (PMP) mechanism. However, PMP is designed for simple memory layouts and does not scale well in complex systems. Instead, the MPT specification introduces a more scalable solution. It utilizes a new CSR, `mmpt`, which stores both the Supervisor Domain ID (SDID) assigned to a hart and the base address of a trie-based structure defining access permissions for physical pages. The MPT intercepts address translations performed by standard page tables or G-stage translations, introducing an additional walking process, starting at the `mmpt` address. The MPT uses the most significant bits of the physical address to locate the corresponding MPT entry and uses two or three walking levels. Permissions can be assigned at different page granularities, from 4KB pages up to Gigapages. We implemented the MPT as a stand-alone hardware walker in SystemVerilog, designed for integration within an MMU. While functionally similar to a conventional MMU page walker, the MPT enforces specific access checks to prevent non-valid phyisical page formats and access violations at level one of the trie structure. To properly support the MPT, the target MMU's TLBs must be extended to store the SDID alongside the ASID and VMID, ensuring valid translations across different supervisor domains. To improve performance, the MPT implementation includes a caching mechanism, akin to TLBs, which stores frequently accessed physical pages along with their associated permissions. The MPT has been verified and is integrated into the CVA6 MMU as a replacement for PMP. In order to validate its functional behaviour in the CVA6 processor, we adopted the MPT APIs implemented in the OpenSBI codebase [3] running in M-mode. While the Salus TSM is an open implementation running in HS-mode [4], it currently supports only a single deployment model, where the TSM constitutes the entire TCB, limiting the system to a single supervisor domain, hence making MPT not directly usable in the current Salus version.

Future Works

While the RISC-V trusted computing specifications are steadily progressing toward stability, the development of a fully open-source processor capable of supporting these security features remains a long-term endeavor. Achieving this goal requires a combination of robust hardware and software mechanisms to establish a truly secure and open RISC-V ecosystem. CoVE APIs are now supported in the Linux kernel, potentially paving the way for KVM-based CVMs. However, there has been little progress at the hypervisor level, particularly with the Salus project, while more mature hypervisors such as Jailhouse, Xvisor, Bao, and seL4 remain largely unexplored. At a deeper level, there is currently no available TSM driver or system-on-chip with native CoVE support. In this context, our open-source implementation of the MPT [5] marks a significant milestone for a CoVE-based ecosystem, albeit an initial one. Our immediate next steps include (a) integrating OpenTitan as a root of trust and (b) developing an open-source TSM driver. To that end, we are actively working on Shadowfax [6], a Rust-based extension to OpenSBI that runs on our MPT-enabled CVA6 platform, laying the foundation for a RISC-V confidential computing ecosystem.

Acknowledgments

This work has been partially supported by the *Spoke 1 "FutureHPC & BigData"* of ICSC - Centro Nazionale di Ricerca in High-Performance-Computing, Big Data and Quantum Computing, funded by European Union - NextGenerationEU.

References

[1] Ravi Sahita et al. "CoVE: Towards confidential computing on RISC-V platforms". In: *Proceedings of the 20th ACM International Conference on Computing Frontiers*. 2023, pp. 315–321.

[2] Bruno Sá et al. "CVA6 RISC-V virtualization: Architecture, microarchitecture, and design space exploration". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2023).

[3] *GitHub - grg-haas/smmtt — github.com*. `https://github.com/grg-haas/smmtt`.

[4] *GitHub - rivosinc/salus: Risc-V hypervisor for TEE development — github.com*. `https://github.com/rivosinc/salus`.

[5] Valerio Di Domenico Stefano Mercogliano. *RISC-V Memory ProtectionTable*. `https://github.com/Granp4sso/RISC-V-Memory-Protection-Table`. 2025.

[6] Giuseppe Capasso Stefano Mercogliano. *Shadowfax: a CoVE-compliant TSM-driver written in rust*. `https://github.com/Granp4sso/shadowfax`. 2025.