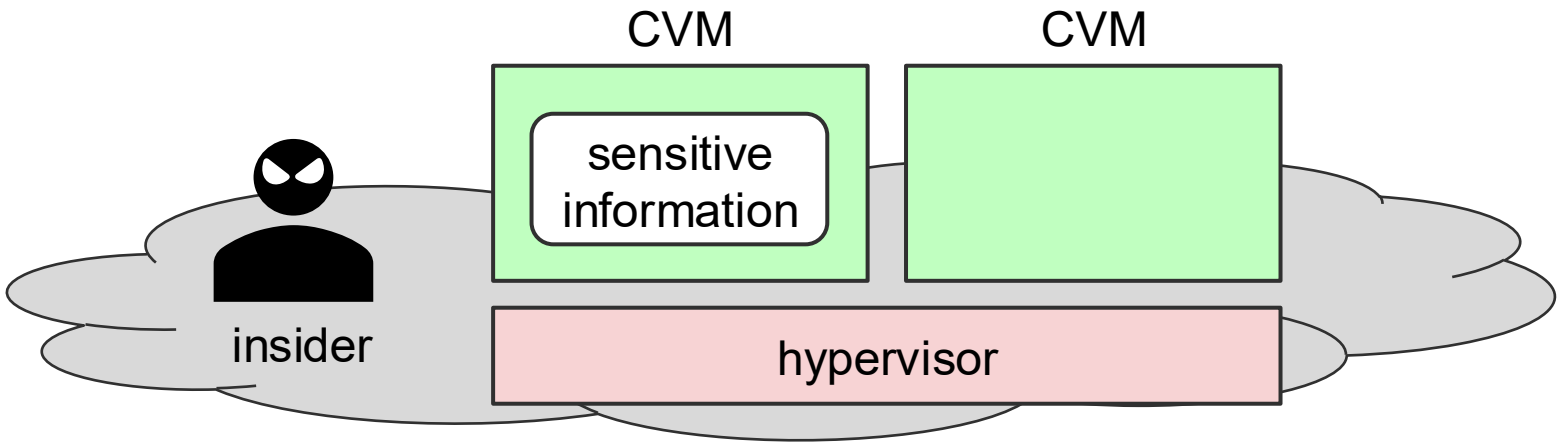
Software-defined SEV for Confidential VMs Based on the Delegation Architecture in RISC-V

Junpei Matsushita and Kenichi Kourai

Kyushu Institute of Technology, Japan

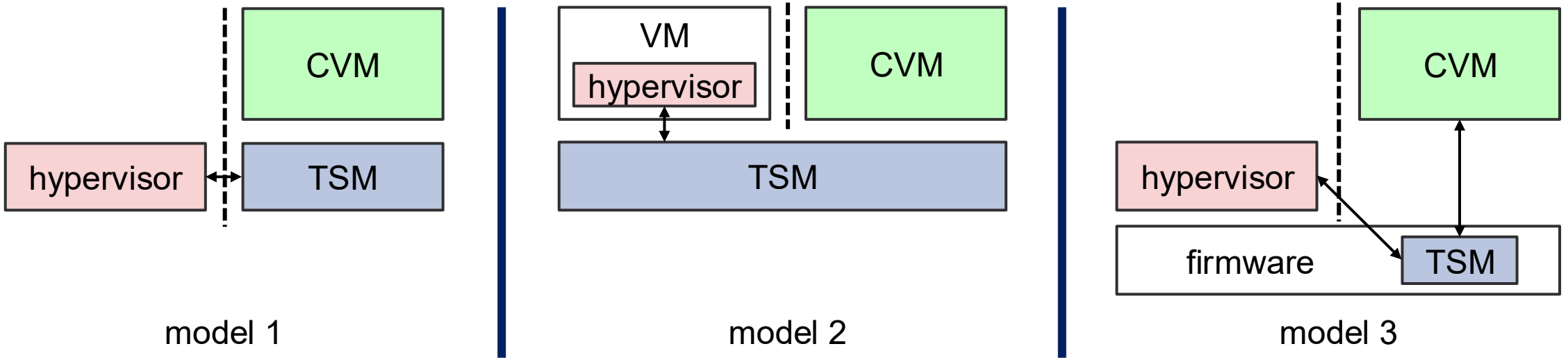
Confidential VMs (CVMs)

- **Users handle sensitive information in clouds**
  - The threat of cloud insiders is increasing
  - Eavesdrop on and tamper with sensitive information in users' VMs
- **Recent clouds offer confidential VMs**
  - The memory is protected by trusted execution environments (TEEs)
  - TEEs are isolated even from the hypervisor



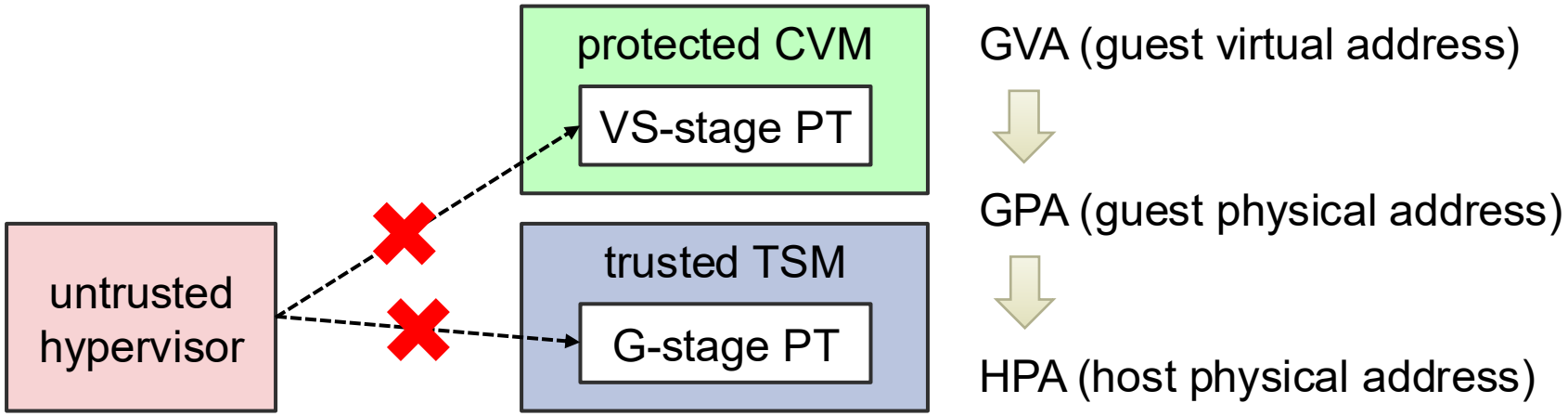
RISC-V CoVE

- **Confidential VM Extension (CoVE) is provided in RISC-V**
  - Use the trusted TEE security manager (TSM) to run CVMs
- **Three deployment models are defined**
  - Model 1: Run the TSM on confidential memory divided by hardware
  - Model 2: Run the hypervisor on the TSM using nested virtualization
  - Model 3: Run the TSM inside the firmware running in M-mode



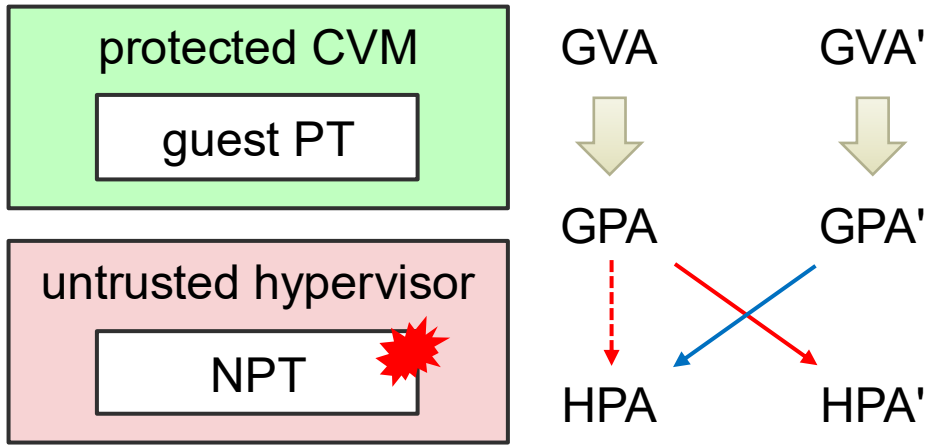
Direct Management Architecture

- **Use two-stage address translation to isolate CVMs**
  - Translate a GVA into a GPA using the VS-stage page tables (PT)
  - Translate the GPA into an HPA using the G-stage PT
- **The two PTs are protected to guarantee its correctness**
  - The VS-stage PT is managed by the guest OS in a protected CVM
  - The G-stage PT is managed by the trusted TSM



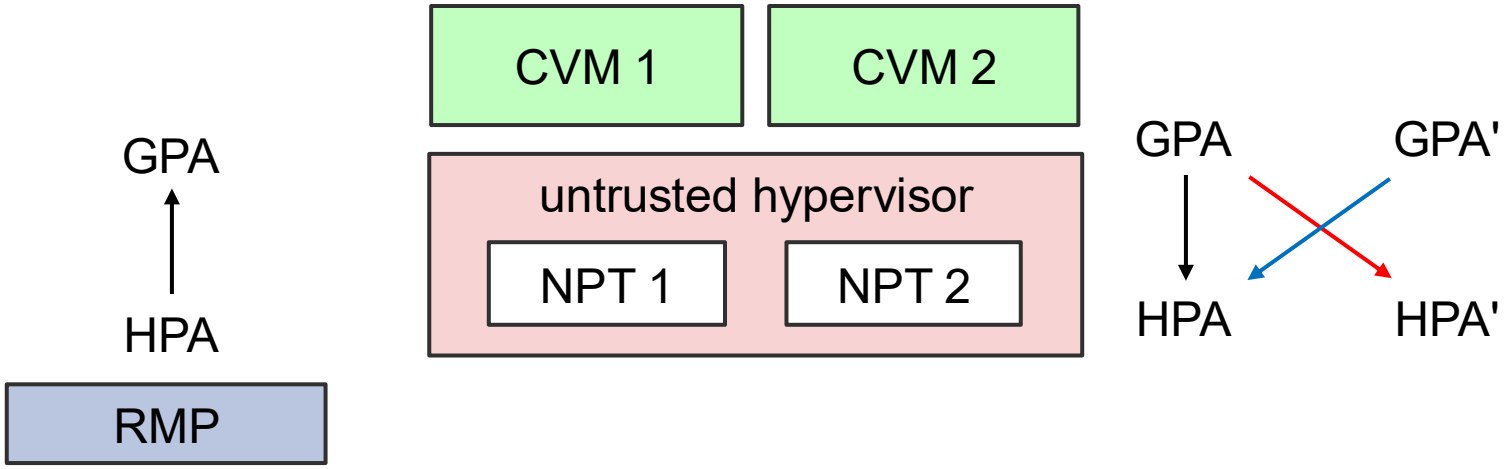
AMD SEV-SNP

- **Use no such trusted component as the TSM**
  - Could reduce the TCB size
  - The NPT (= G-stage PT) is managed by an untrusted hypervisor
- **Not guarantee the correctness of address translation**
  - Can change memory assignment to a CVM (remapping attack)
  - Can map the same memory to multiple locations (aliasing attack)



Delegation Architecture

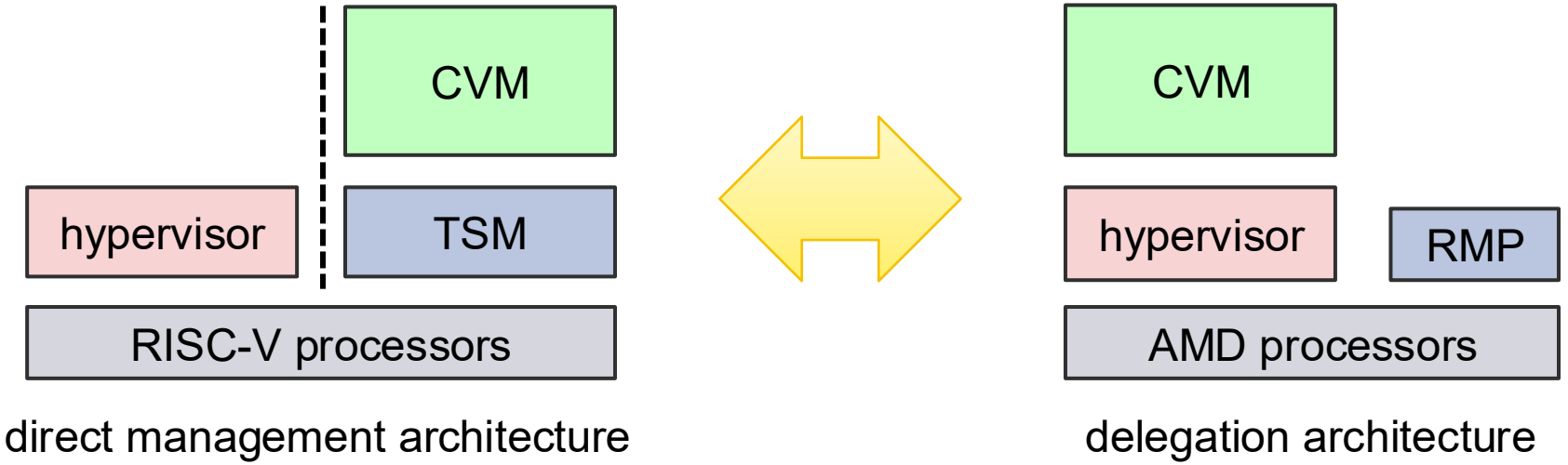
- **Use the hardware-protected Reverse Map Table (RMP)**
  - Maintain a reverse mapping from an HPA to a GPA
  - Maintain a page owner (CVM or the hypervisor)
- **Perform an RMP check on every address translation**
  - Verify the consistency between forward and reverse mappings
  - Verify the correctness of page ownership



Research Question

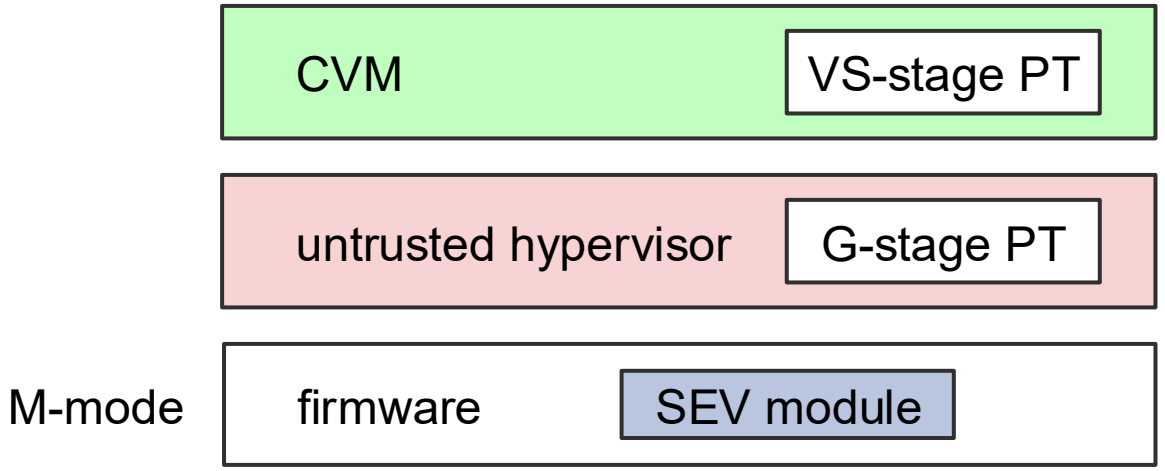
- **It is unclear whether this delegation architecture of SEV-SNP is implementable in RISC-V**

- **Goals**
  - Design the delegation architecture for RISC-V
  - Clarify the runtime overhead of its implementation



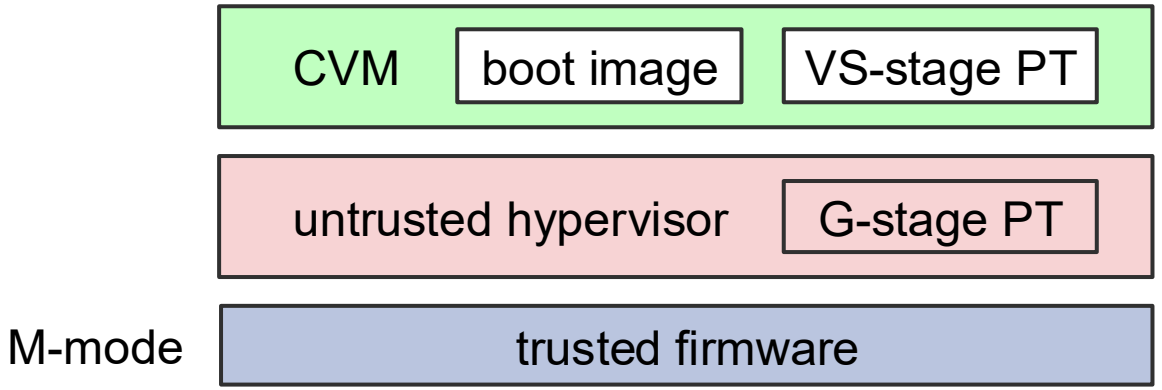
Software-defined SEV (SD-SEV)

- **Realize the delegation architecture mostly in software**
  - Use a small trusted software module (SEV module)
  - Rely on a minimal hardware extension
- **The SEV module runs inside the firmware in M-mode**
  - Isolated from an untrusted hypervisor running on the firmware
  - Emulate SEV-SNP features, e.g., memory integrity and authenticity



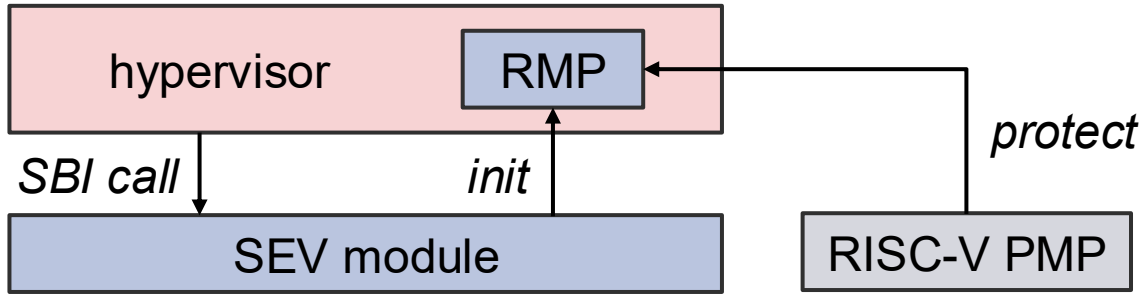
Threat Model

- **Trust assumptions**
  - Software in M-mode, including the SEV module, is trusted
  - The hypervisor and the G-stage PT are untrusted
- **Attacker model**
  - Remapping and aliasing attacks by modifying the G-stage PT
  - Direct attacks against the memory of CVMs by the hypervisor
  - Attacks against the boot images loaded into CVMs



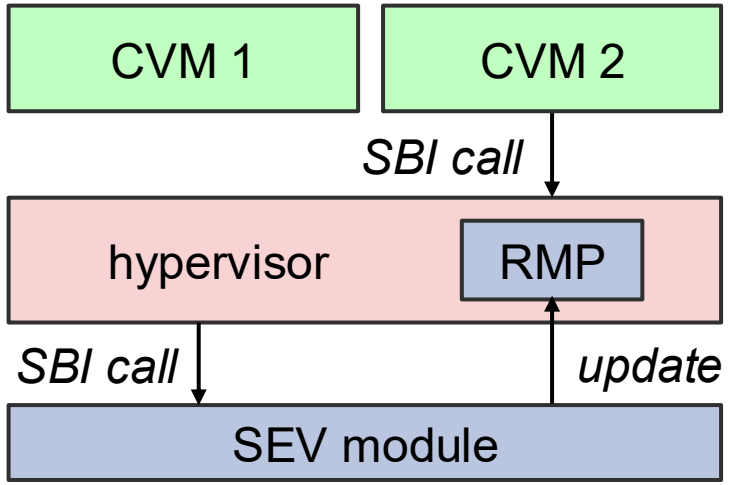
RMP Initialization

- **The hypervisor invokes the SEV module using an SBI call**
  - Instead of issuing a command of AMD Secure Processor (AMD-SP)
  - Allocate and pass the memory region used for the RMP
- **The SEV module protects the RMP using RISC-V PMP**
  - Configure the memory region to be read-only for the hypervisor
  - Create entries for 4-KB physical pages in the RMP



RMP Update

- **The hypervisor updates an RMP entry using an SBI call**
  - The SEV module sets the VM ID and the GPA in the RMP entry
  - It marks the RMP entry as unvalidated to delect illegal assignment
- **A CVM updates an RMP entry via the hypervisor**
  - Mark a page as unassigned to a CVM
  - Share the memory accessed by virtual devices with the hypervisor

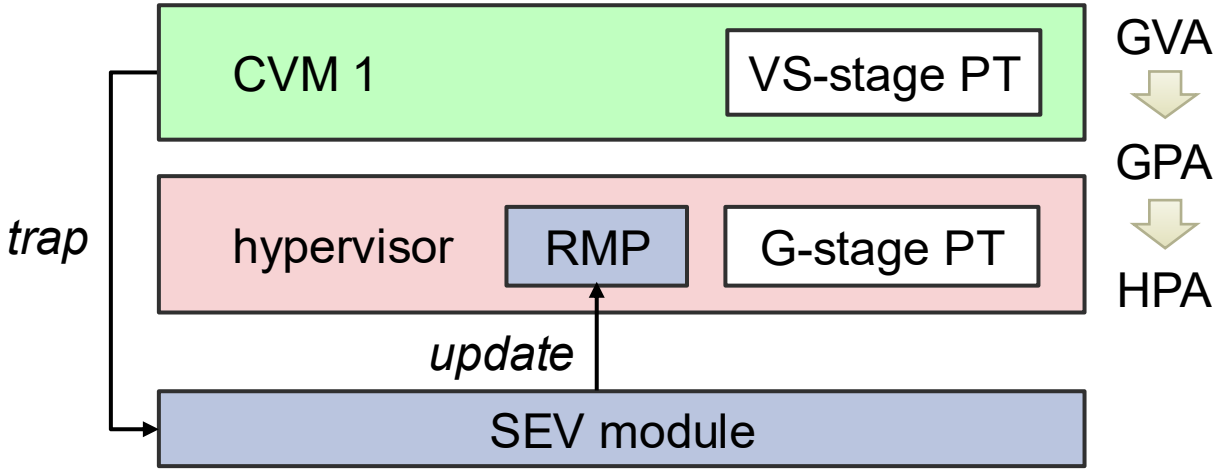


RMP

HPA	VM ID	GPA	Assigned	Validated
10	1	50	1	0
11	2	51	0	0

Validation of an RMP Entry

- **A CVM validates an RMP entry by specifying a GVA**
  - The SEV module traps an undefined instruction used for this purpose
  - It marks the RMP entry as validated to accept current assignment
- **The SEV module finds the RMP entry from the GVA**
  - Translate the GVA into an GPA using the VS-stage PT manually
  - Translate the GPA into an HPA using the G-state PT manually

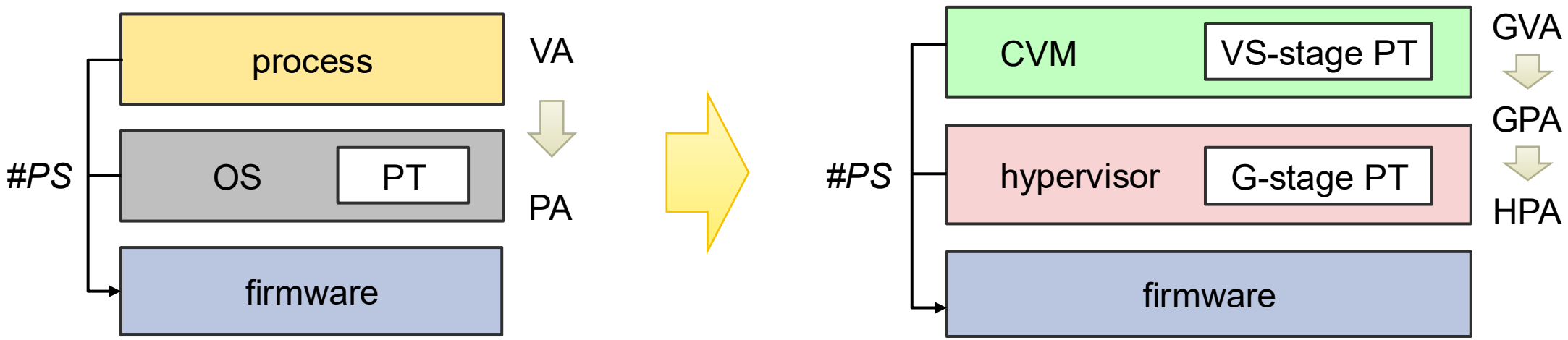


RMP

HPA	VM ID	GPA	Assigned	Validated
10	1	50	1	1

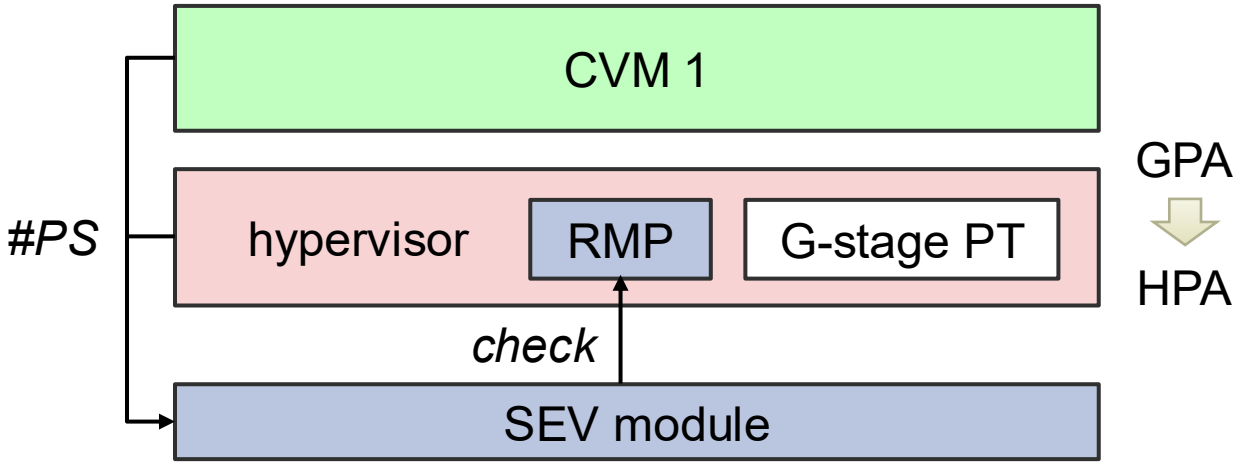
Extention of Page Success Exception (#PS)

- **#PS is a new exception added to RISC-V [Yamamoto+'25]**
  - Raised when the success of address translation on a TLB miss
    - Opposite of the page fault exception
  - The firmware can perform access control using the VA and PA
- **Add support for the two-stage address translation of VMs**
  - The SEV module performs an RMP check using the GPA and HPA



RMP Check

- **Verify page ownership and mapping consistency**
  - Find the RMP entry corresponding to the HPA
  - Fail if the hypervisor attempts to access the page assigned to a CVM
  - Fail if the VM ID and the GPA do not match the RMP entry
- **Verify page validation status**
  - Fail if a CVM attempts to access an unvalidated page

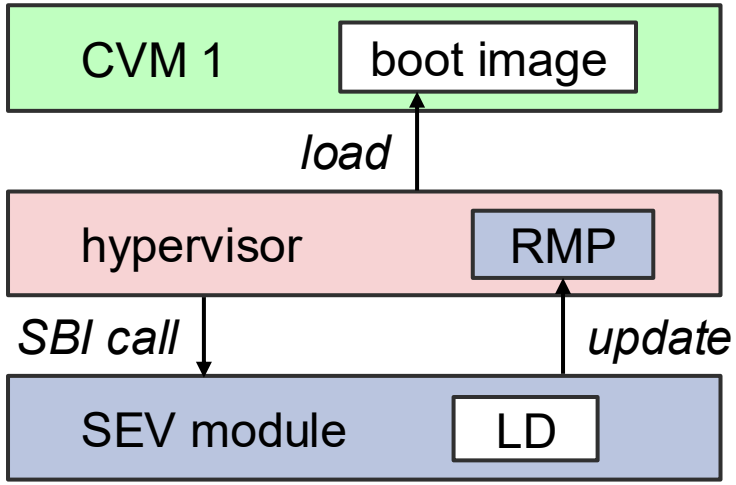


RMP

HPA	VM ID	GPA	Assigned	Validated
10	1	50	1	1

Measurement of a Boot Image

- **SD-SEV measures the initial memory pages of a CVM**
  - The target regions consist of a boot image including the OS
- **The hypervisor measures each page using an SBI call**
  - The SEV module updates a launch digest (LD) of the CVM
  - It computes a hash value of the page, its GPA, and the current LD
  - It marks the corresponding RMP entry as validated to bootstrap

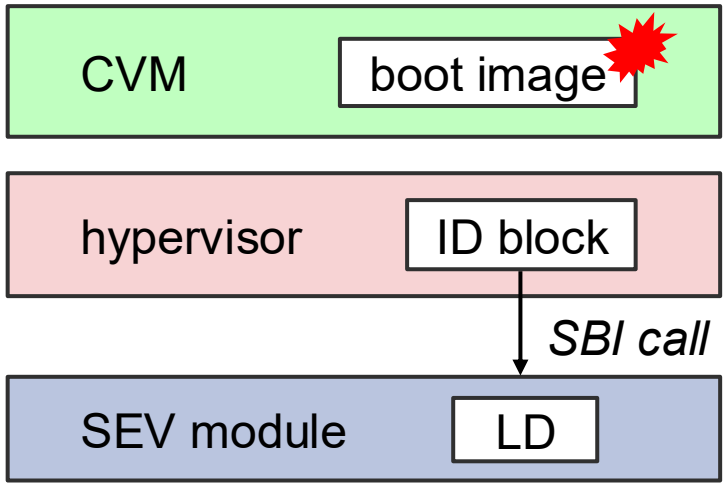


RMP

HPA	VM ID	GPA	Assigned	Validated
12	1	52	1	1

LD Verification

- **SD-SEV can verify the LD at the boot time of a CVM**
  - The VM owner can pass an ID block to the SEV module
  - The ID block contains the pre-computed LD of the CVM
- **The SEV module checks the LD on finishing measurement**
  - Fail if the computed LD does not match the ID block
  - Detect the tampering of the boot image by the hypervisor



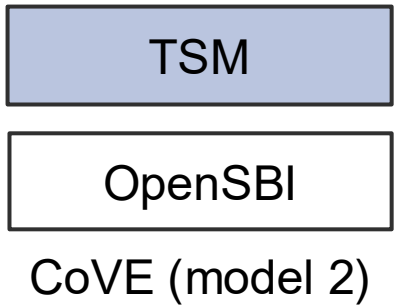
Experiments

- **We conducted several experiments**
  - Compare the TCB size with CoVE's
  - Confirm the memory integrity and authenticity of a CVM
  - Exmine the overhead of SD-SEV

	host	emulation	VM
CPU	Intel Core i7-14700	4	2
memory	32 GB	4 GB	1 GB
OS	Linux 6.14	-	Linux 6.15
hypervisor	-	Xvisor 0.3.2	
firmware	-	OpenSBI 1.6	Basic Firmware
emulator	QEMU 10.0.50	-	-

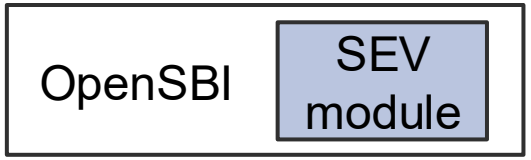
TCB Size

- **We measured the binary sizes of the TCB**
  - SD-SEV, CoVE (model 2), and CoVE (model 3)
  - Compared with the original TCB (OpenSBI in M-mode)
- **SD-SEV's TCB is only 4% larger than the baseline**
  - CoVE's TCB is 2-19x larger

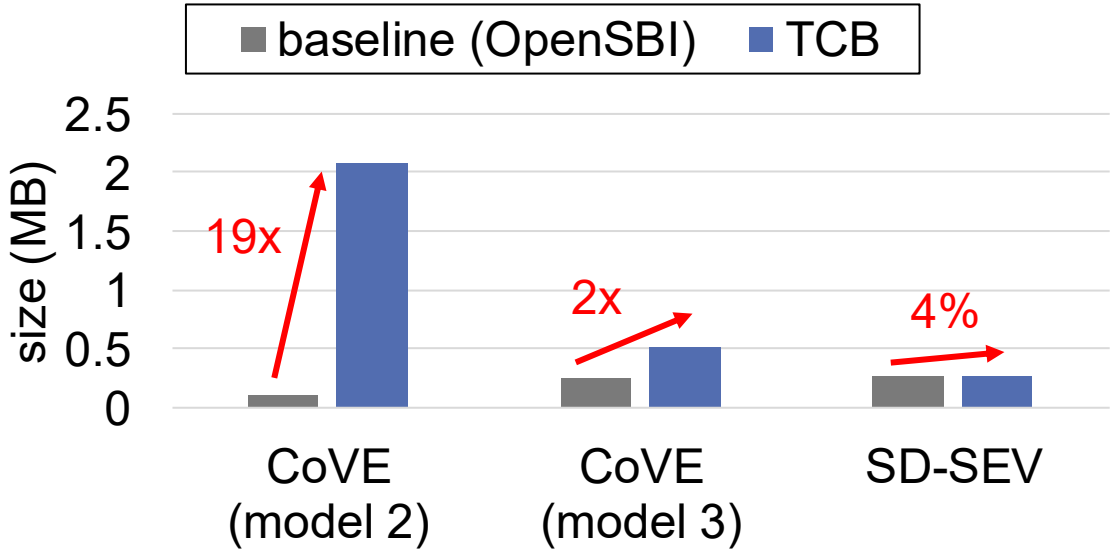


CoVE (model 2)

CoVE (model 3)

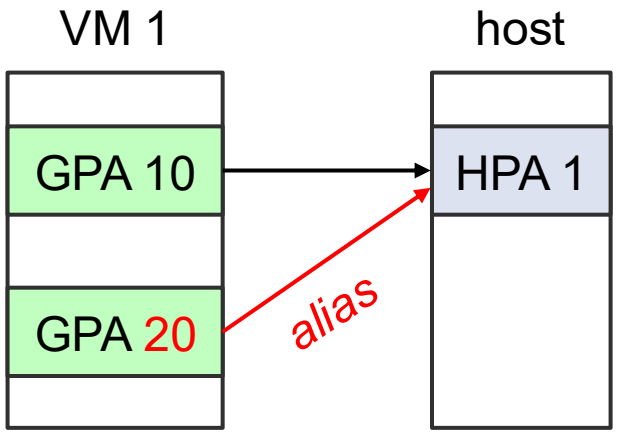


SD-SEV



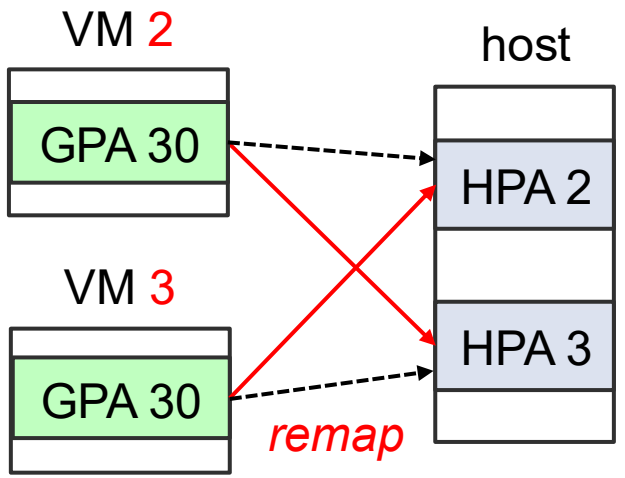
Attack Detection

- **We mapped two GPAs onto the same HPA in a CVM**
  - Detected this attack because one GPA did not match the RMP entry
- **We exchanged physical pages between two CVMs**
  - Detected this remapping attack by verifying page ownership
- **We loaded a modified OS into a CVM**
  - Detected this attack against the boot image by LD verification



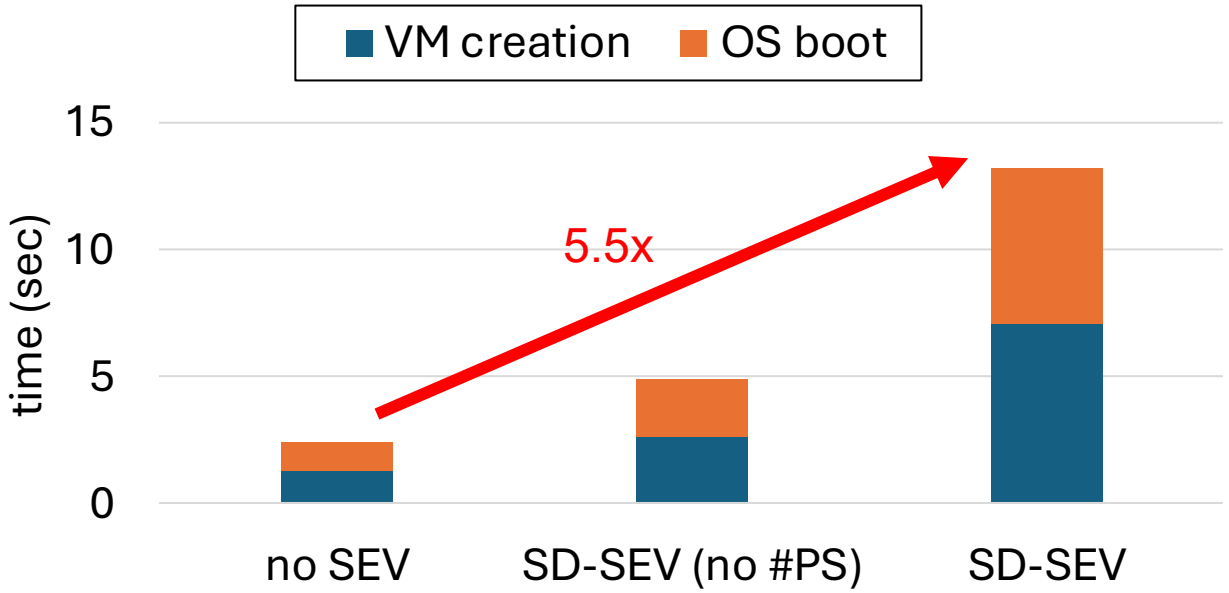
RMP

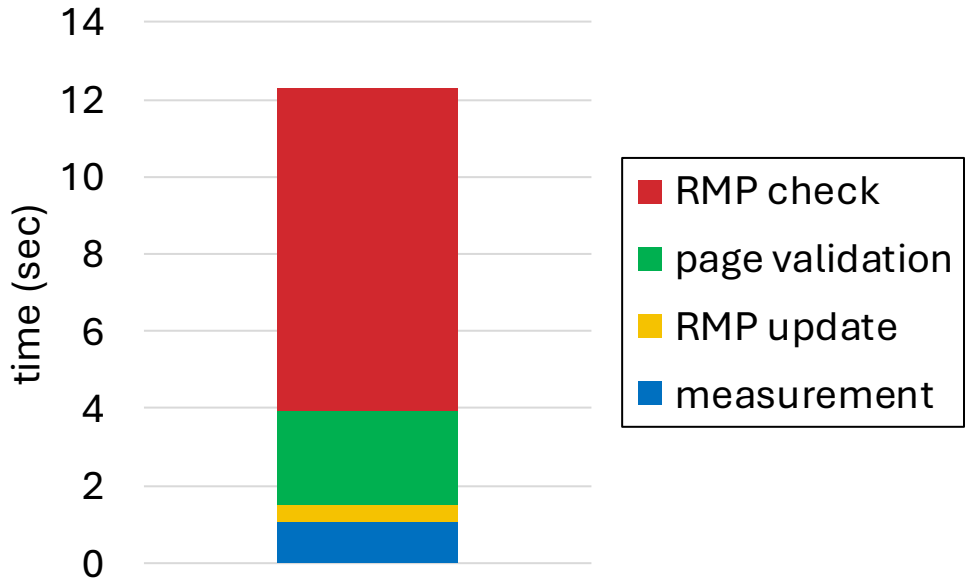
HPA	VM ID	GPA	Validated
1	1	10	1
2	2	30	1
3	3	30	1



Boot Time

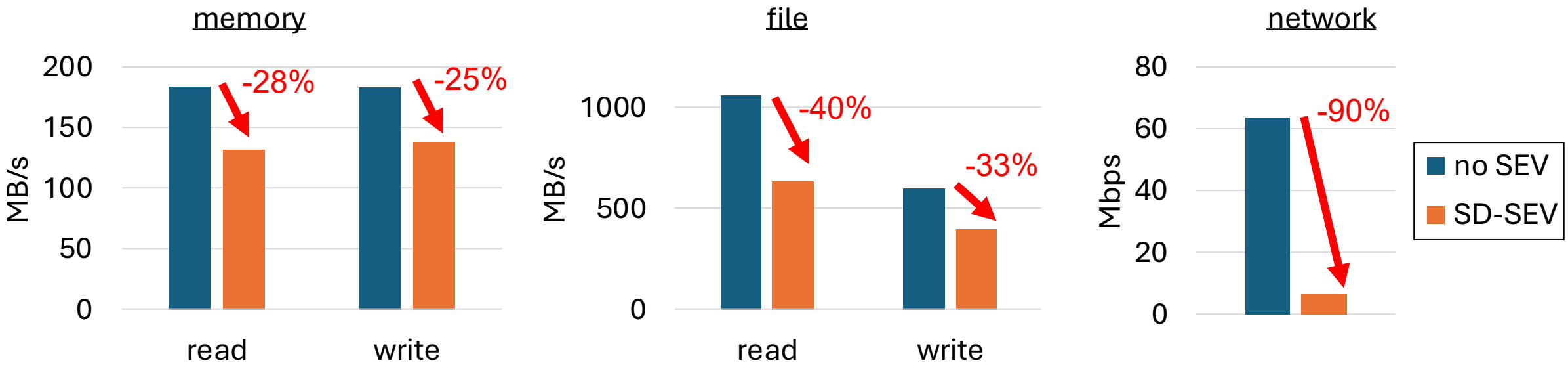
- **The boot time of a CVM was 5.5x longer**
  - Increase the time of both the VM creation and the OS boot
  - Enabling #PS increased the boot time significantly
- **RMP checks took much longer due to too many #PS**
  - Only 0.85 us per RMP check





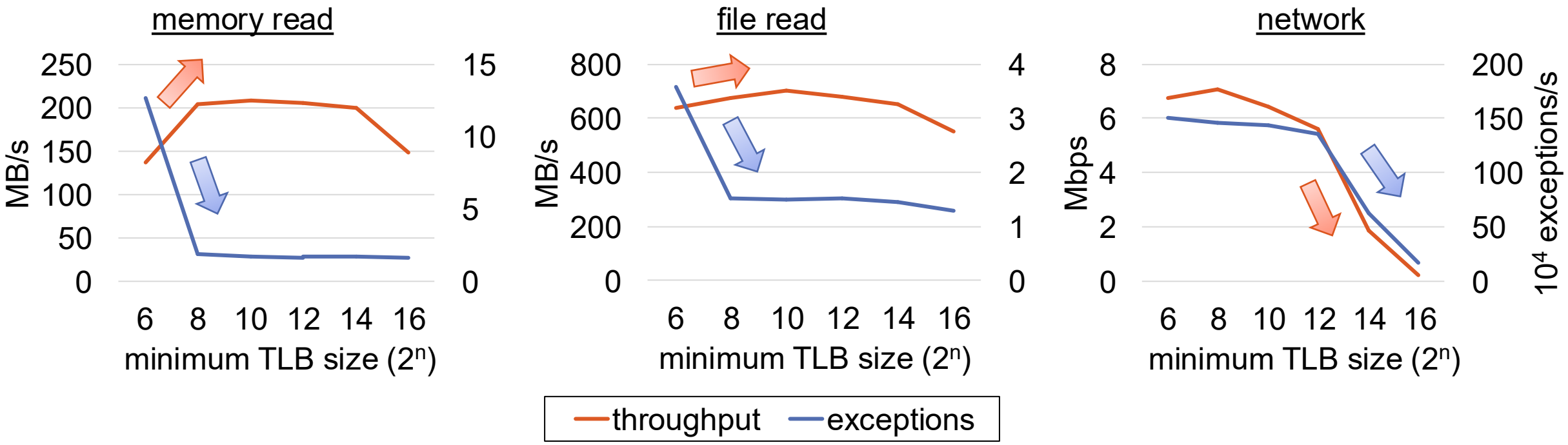
VM Performance

- **Memory access performance degraded by 25-28%**
  - The VM raised 50-75% of #PS
- **File access performance degraded by 33-40%**
  - The virtual device in the hypervisor raised 70-90% of #PS
- **Network throughput degraded by 90% due to 7x #PS**



Impact of #PS

- **We increased the minimum TLB size to reduce #PS**
  - Increased memory read performance by 50%
  - Slightly increased file read performance by 5%
  - Not effective for network throughput



Related Work

- **ACE [Ozga+, HASP'23]**
  - Guarantee exclusive memory assignment by Rust's ownership
  - Fomally verify the memory safety of the core part with RefinedRust
- **Other commercial TEEs for CVMs**
  - The secure EPT is managed by the trusted TDX module in Intel TDX
  - The RTT is managed by the trusted RMM in Arm CCA
- **AnyTEE [Cerdeira+, Access'25]**
  - Software-defined TEE framework with hardware virtualization
  - Implementing TEEs for CVMs is future work

Conclusion

- **We proposed SD-SEV for CVMs based on the delegation architecture for RISC-V**
  - Run the small, trusted SEV module in M-mode
  - Use a minimal hardware extension called a page success exception
  - The SEV module performs RMP checks using #PS
  - Showed that the performance of CVMs was significantly affected
- **Future work**
  - Improve performance by reducing the overhead of #PS