Confidential VM Extension I/O (CoVE-IO) for Confidential Computing on RISC-V platforms

Editor - Samuel Ortiz, Jiewen Yao, RISC-V AP-TEE-IO Task Group

Version 0.1, 2023/04: This document is in development. Assume everything can change. See http://riscv.org/spec-state for details.

Table of Contents

Preamble

This document is in the Development state

Assume everything can change. This draft specification will change before being accepted as standard, so implementations made to this draft specification will likely not conform to the future standard.

Copyright and license information

Contributors

This RISC-V specification has been contributed to directly or indirectly by:

- Samuel Ortiz <sameo@rivosinc.com>
- Jiewen Yao <jiewen.yao@intel.com>

Chapter 1. Introduction

The RISC-V Confidential VM Extension [CoVE] specification provides application and virtualized workloads with data confidentiality and integrity, addressing one of the major datacenter security challenge. By building a miminized Trusted Computing Base (TCB), the CoVE interfaces manage to isolate workloads from each others but also from untrusted domain host software components (e.g. the hypervisor). CoVE implementations create a Confidential Computing [CC] framework that allows for mitigating the risks that guest owners are exposed to when running their workloads on shared infrastructures like e.g. public clouds.

The value of such hardarwe-based Trusted Execution Environments (TEEs) is acknowledged across the industry, and Confidential Computing providers are looking for ways to improve the technology's scalability and performance. With the growing need for securing data processing, expanding Confidential Computing guests TCBs with trusted devices is of particular interest. Data analytics and transformation, artificial intelligence, financial transactions processing are only a few examples of confidential workloads for which a secure and performant I/O architecture is key to their operations. By extending guests TCBs with trusted accelerators, NICs or GPUs, such workloads would fully take advantage of their infrastructure providers capacities while keeping their data protected.

The CoVE interfaces provide TEE Virtual Machines (TVMs) memory with confidential attributes and allow TVM guests to share parts of their address space with an untrusted domain. Although this enables confidential guests to be assigned with devices, directly or not, through para-virtualized I/O implementation, it comes with a significant performance cost. Withouth additional protection, TVMs have no ways to trust hypervisor-assigned devices and must exclude them from their TCB by not allowing them to directly access confidential memory. Consequently, with the current CoVE specification, data flowing from a device to a TVM must first go through dedicated shared memory regions for the confidential guest to then move it over to its confidential address space. This systematic data copy between shared and confidential memory is called bounce-buffering and can have a major performance impact for confidential workloads.

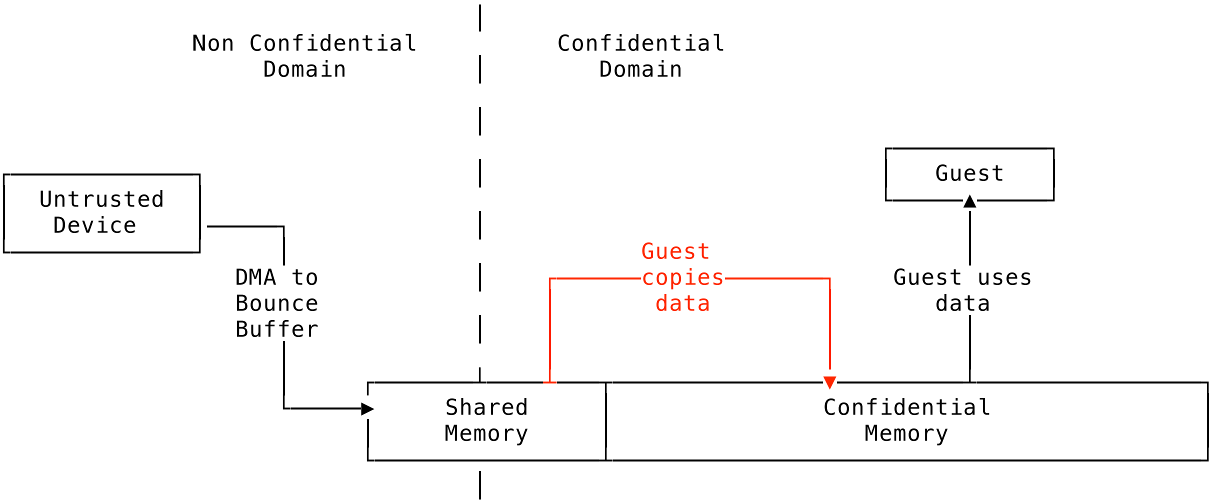

Figure 1. Bounce buffering between an untrusted device and a CoVE TVM

As devices typically expose their programming interfaces through memory mapped registers, using a shared memory buffer requires additional protection of the communication between the device and the TVM (e.g. transport-level data encryption). Such additional security layers can be impractical, intrusive and may also degrade I/O performance even further.

Ideally, a secure and performant CoVE I/O framework would rely on the ability for hypervisor-assigned devices to directly access TVMs confidential memory, while maintaining the guest confidentality and integrity protection already provided by the CoVE security model. Building such a framework requires enhancing both the host software stack and the assigned devices with new protection mechanisms. In addition to the existing CoVE defined capabilities, the host software must provide ways for TVM guests to establish trust with assigned devices before accepting them into their TCBs and giving them direct access to confidential memory regions. Devices, on the other side of the I/O link, must protect confidential guest workloads and their data from untrusted domain components controlling, accessing or tampering with them.

This document describes a proposal for extending the CoVE specification with I/O specific flows, interfaces and intrinsics with the goal of implementing the above-described framework. The CoVE I/O interfaces aim at giving CoVE TVM guests the ability to securily:

- Retrieve a device identity, configuration and security state in order for them to establish trust with the device.
- Verify that any untrusted domain component will not be able to intercept, modify or control data flowing between a guest and its assigned devices.
- Decide to accept devices into their TCB before allowing them to directly access their confidential memory, and before being able to control and configure said devices.

The CoVE I/O framework builds on top of the industry supported and ratified [PCI-SIG] TEE Device Interface Security Protocol [TDISP] specification, which itself relies on the [DMTF] Secure Protocol and Data Model [SPDM] protocol. TDISP compliant devices, also known as TEE-IO devices, implement security protections for isolating guest workloads and confidential data from domains to which a device interface is not assigned to. It also requires TEE-IO devices to provide secure means for confidential guests to attest of any device interface trustworthiness and verify its security configuration.

Chapter 2. Glossary

Term	Acronym	Definition
Application Processor	AP	APs can support commodity operating systems, hypervisors/VMMs and applications software workloads. The AP subsystem may contain several processing units, on-chip caches, and other controllers for interfacing with memory, accelerators, and other fixed-function logic. Multiple APs may be used within a logical system.
Application Processor- Trusted Execution Environment	AP-TEE	An execution mode that provides HW-isolation for workload assets when in use (user/supervisor code/data) and provides HW-attestable confidentiality and integrity protection against specific attack vectors per a specified adversary and threat model. The term CoVE, TEE and hardware-based TEE are also used as synonyms of AP-TEE in this document.
Attestation	N/A	The process by which a relying party can assess the security posture of the confidential workload based on verifying a set of HW-rooted cryptographically-protected evidence.
Confidential Computing	N/A	The protection of data in use by performing computation in a Hardware-based TEE.
Confidential VM Extension	CoVE	A set of non-ISA RISC-V extensions that enables confidential computing on RISC-V platforms.
Device Interface	DI	TDISP term, Device Interface. Same as a TDI. A DI can be a Virtual Function (VF) or a Physical Function (PF).
Device Security Manager	DSM	A DSM is a logical entity on a TEE-IO device that enforces the TDISP security policies, attributes and states.
Direct Memory Access	DMA	The ability for a device to directly read and write into the host physical memory.
Host platform	N/A	The combination of the host software stack and the host hardware components. The host hardware components include all SoC components (CPU, memory controller, power management IP blocks, etc) and all discrete ones (Root of Trust, IOMMU, physical memory, etc).
Host software	N/A	All software elements including type-1 or type-2 HS-mode VMM and OS; U mode user-space VMM tools; ordinary VMs hosted by the VMM that emulate devices. The hosting platform is typically a multi-tenant platform that hosts multiple mutually distrusting Tenants.

Term	Acronym	Definition
Hypervisor or Virtual Machine Monitor	VMM	HS mode software that manages Virtual Machines by virtualizing hart, guest physical memory and IO resources. This document uses the term VMM and hypervisor interchangeably for this software entity.
Input/Output Memory Management Unit	IOMMU	A system agent that translates device virtual addresses to physical addresses.
Integrity and Data Encryption	IDE	Extended PCIe capability for integrity, confidentiality and replay protection of PCIe Transport Layer Packets (TLP).
I/O Translation Agent	N/A	Same as an IOMMU.
Platform TCB	TCB	Platform Trusted Computing Base, same as TCB.
Physical Device	N/A	The physical device to which DIs belong to. This is the actual physical PCIe device.
Root of Trust	ROT	Isolated HW/SW subsystem with an immutable ROM firmware and isolated compute and memory elements that form the Trusted Compute Base of a TEE system. The RoT manages cryptographic keys and other security critical functions such as system lifecycle and debug authorization. The RoT provides trusted services to other software on the platform such as verified boot, key provisioning, and management, security lifecycle management, sealed storage, device management, crypto services, attestation etc. The RoT may be an integrated or discrete element, and may take on the role of a Device Identification Composition Engine (DICE).
Secure Protocol and Data Model	SPDM	A DMTF defined specification for exchanging messages between devices over a variety of transports and physical media. In the CoVE-IO context, SPDM is used to exchange TDISP and IDE Key Management messages over PCIe DOE mailboxes.
System TCB	TCB	System Trusted Computing Base, same as TCB.
TEE Device Interface Security Protocol	TDISP	An architecture for trusted I/O virtualization.
TEE I/O Device Interface	TDI	The unit of assignment for a trusted I/O capable device. For example, a TDI can be a Virtual Function (VF) or a Physical Function (PF).
TEE Security Manager	TSM	HS-mode software module that acts as the trusted (in TCB) intermediary between the VMM and the TVM. This module extends the TCB chain on the CoVE platform.

Term	Acronym	Definition
Trusted Computing Base	TCB	The hardware, software and firmware elements that are trusted by a relying party to protect the confidentiality and integrity of the relying parties' workload data and execution against a defined adversary model. In a system with separate processing elements within a package on a socket, the TCB boundary is the package. In a multi-socket system the TCB extends across the socket-to-socket interface, and is managed as one system TCB.
Trusted Device Manager	TDM	A CoVE-IO device manager, responsible for verifying, attesting and accepting CoVE-IO devices into a TVM TCB. This is a TVM guest software stack component.
Trusted Memory Mapped Input Output	Trusted MMIO	A TDI memory mapped I/O region that can only be accessed by a TVM that accepted the TDI in its TCB. TDIs describe their trusted MMIO regions through TDISP. The TVM, with the TSM support, is responsible for verifying that trusted MMIO ranges are correctly mapped into its address space.
TEE VM	TVM	A VM instantiation of an confidential workload.
Virtual Machine	VM	Virtual Machines hosted by a VMM

Chapter 3. Requirements

In order to extend TVM TCBs with external and untrusted devices, both the host platform and the assigned devices, must meet a specific set of requirements.

3.1. Device

CoVE-IO compliant implementations support extending TVMs TCBs with devices if and only if they meet the following requirements:

3.1.1. PCIe or CXL.io

A CoVE-IO supported device must be a PCIe or CXL.io compliant device with optional SR-IOV capabilities. When supporting SR-IOV, the device can be split into multiple Virtual Functions (VFs). A TVM assigned interface can be the whole device, also known as the Physiscal Device (PF) or any of its VFs. In both cases, this unit of assignment is referred to as a TEE I/O Device Interface (TDI).

A host enumerated TDI must provide the following PCIe capabilities:

TEE Device Interface Protocol (TDISP)

In order to interoperate with a RISC-V CoVE-IO implementation, PCIe devices must support the [TDISP] protocol version 1.0 or above, as defined in the latest TDISP ECN. CoVE-IO devices must support all required TDISP requests and may support the optional ones, as defined by the TDISP request code table.

Moreover, CoVE-IO compatible device firmwares must run a Device Security Manager (DSM) to enforce the TDISP defined security attributes and policies. The DSM must support both full PCIe devices (PFs) and Virtual Functions (VFs) assignment to TVMs.

Enumeration

TDIs PCIe Device Capabilities Registers must have the "TEE-IO Supported" bit in "Device Capabilities Register" set in order for the host to discover their TDISP capability.

Data Object Exchange (DOE)

The TDI implements the optional PCIe Data Object Exchange mechanism. The minimal supported version is 1.0 defined in [PCIE].

A CoVE-IO supported device must support the following DOE object types:

Vendor ID	Object Type	Description
0001	00	DOE Discovery
0001	01	CMA/SPDM
0001	02	Secure CMA/SPDM

Secure Protocol and Data Model (SPDM)

RISC-V CoVE-IO compliant hosts use the Secure Protocol and Data Model ([SPDM]) protocol to exchange TDISP and IDE Key Management (IDE_KM) messages with physical devices, over DOE mailboxes. CoVE-IO also relies on SPDM for gathering devices certificates and measurements.

As a consequence, a CoVE-IO supported DSM must support the SPDM specification version 1.2 or above. It must also support sending and receiving SPDM Secured Messages as defined in the [SecuredSPDM] specification version 1.1 or above.

CoVE-IO compatible DSMs must support the following SPDM responder capabilities: `CERT_CAP`, `MEAS_CAP` (10b), `ENCRYPT_CAP`, `MAC_CAP`, `KEY_EX_CAP`.

Moreover, CoVE-IO compatible DSMs must support below algorithms: 1. For `BaseAsymAlgo`, one or more of the following ones: `TPM_ALG_RSASSA_3072`, `TPM_ALG_ECDSA_ECC_NIST_P256`, `TPM_ALG_ECDSA_ECC_NIST_P384`. 2. For `BaseHashAlgo`, one or more of the following ones: `TPM_ALG_SHA_256`, `TPM_ALG_SHA_384`. 3. For `MeasurementHashAlgo`, one or more of the following ones: `TPM_ALG_SHA_256`, `TPM_ALG_SHA_384`. 4. For `DHE Group`, one or more of the following ones: `secp256r1`, `secp384r1`. 5. For `AEAD` Cipher Suite, `AES-256-GCM` with 16 Byte MAC.

Integrity and Data Exchange (IDE)

The data that flows between a TDI and a TVM must be confidential and both integrity and replay attack protected. PCIe IDE provides this protection for all Transport Layer Packets (TLPs) moving between those two endpoints. As a consequence, CoVE-IO compatible devices must implement IDE and expose this PCIe capability defined in [PCIE].

Selective Streams

As all PCIe switches are excluded from a TVM's TCB, CoVE-IO compliant hosts TSMs exclusively establish selective IDE streams between the host PCIe Root Port and TEE-IO devices.

Key Management

To fully support IDE, CoVE-IO compatible devices must implement the IDE Key Management (IDE_KM) protocol. IDE_KM messages are initiated by the TSM and sent over SPDM to the DSM.

CoVE-IO compatible devices must keep track of the IDE key invocation field, which is in bit 63:0 of the IDE sub-stream specific AES-GCM initialization vector (IV). If the IDE key invocation field overflows, the IDE stream must enter Insecure state. Before overflowing, the device may notify the host to let the host software perform the IDE_KM defined key refresh process.

3.2. Host

In order to support TEE-IO devices with the above described capabilities, a CoVE implementation must meet software and hardware requirements, as described in the next sections.

3.2.1. Ownership

In a CoVE-IO context, the non-confidential host software stack is the sole platform resources owner. In particular, the physical devices to which TDIs belong are owned and managed by the host, typically through the device PF. Host software stack components, e.g. the VMM, assign, unassign and expose TDIs resp. to, from and to TVMs.

3.2.2. Platform Hardware Components

PCIe

All CoVE-IO-compliant devices in a CoVE platform must be connected to the CoVE platform through a PCIe Root Port (RP) that is part of a PCIe Root Complex (RC).

A CoVE-IO PCIe link can be direct or go via any topology or PCIe switches and bridges. Since those are excluded from a TVM's TCB, only selective IDE streams must be used between a CoVE-IO device and its corresponding PCIe Root Port.

The TSM establishes one single selective IDE stream for each physical device from which TDIs may be attached to TVMs. All TDIs within a CoVE-IO device share the same IDE stream.

For a given selective IDE stream, the TSM generates, owns and distributes the IDE keys to both the CoVE-IO device and its upstream PCIe Root Port.

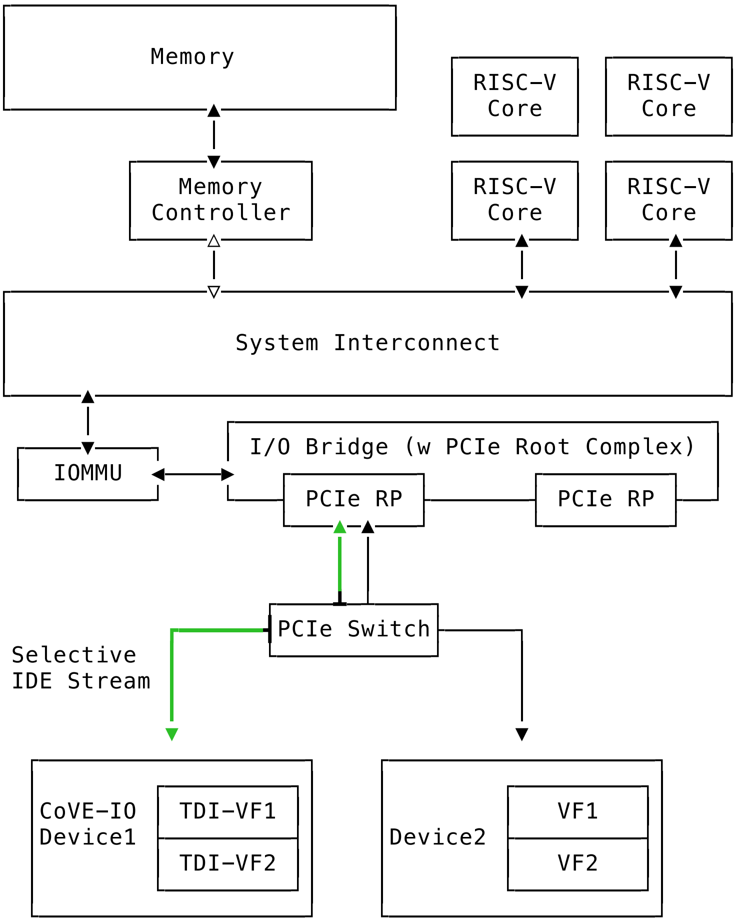

Figure 2. CoVE-IO PCIe Topology

IO Translation Agent

CoVE-IO-compliant platforms must have at least one IO translation agent, also known as an IOMMU. Platform IOMMUs must follow the RISC-V [IOMMU] architecture specification v1.0.

In order to protect direct access to confidential memory, a CoVE-IO device must be attached to a PCIe

Root Port that is bound to a platform IOMMU. All inbound traffic from a TDI must then be translated by the upstream IOMMU.

As the confidential Domain Security Manager (DoSM), the TSM is responsible for setting TDI-specific DMA mapping, MSI page tables and also for configuring platform IOMMUs own MSIs. As a consequence, all IOMMUs on a CoVE-IO platform must provide a domain isolated Register Programming Interface (RPI) that is exclusively accessible to the TSM.

Hardware Root-of-Trust

As described in Section 3.2.2.1, the TSM generates and sets the IDE keys into both the CoVE-IO PCIe endpoint and its upstream Root Port, for all maintained selective IDE streams.

When setting IDE keys into a CoVE-IO device, the TSM relies on the DSM IDE Key Management (`IDE_KM`) support, and its ability to receive IDE_KM messages over a Secured SPDM session. However, there are no architecturally-defined PCIe protocol for managing Root Port IDE keys.

Instead of adding multiple vendor-specific `IDE_KM` implementations to the TSM, the TSM relies on the platform hardware Root-of-Trust (HROT) to implement the `IDE_KM` protocol and abstract the platform specific PCIe RP implementation away from the TSM. The TSM establishes a Secured SPDM session with the HROT over a host accessible DOE mailbox, and then sets platform RP IDE keys over that session.

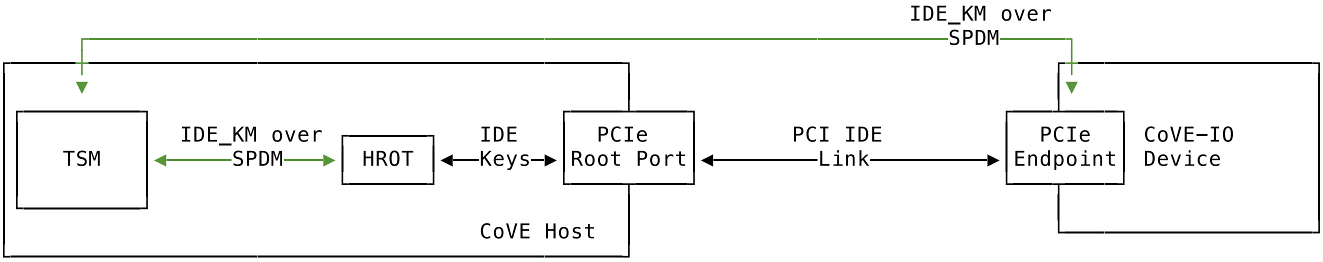

Figure 3. PCIe Root Port IDE Key Management through Hardware Root-of-Trust

As a consequence, a CoVE-IO-compliant platform must have at least one PCIe accessible HROT, with the following requirements:

1. The HROT must support the DOE mechanism
2. The HROT must support Secured SPDM sessions
3. The HROT must support the IDE Key Management protocol

CoVE-IO Manifest

The TSM must be provisioned with a trusted piece of data describing the required CoVE-IO platform components. The hardware Root-of-Trust provides the TSM with a CoVE-IO manifest containing the following pieces of information:

Trust anchor

A list of root certificates that the TSM uses to verify DSM certificates received through SPDM.

IOMMUs

For each IOMMU present in the platform:

- The IOMMU RPI MMIO base address. This is used as the IOMMU identifier.

PCIe Root Ports

For each PCIe Root Ports present in the platform:

- A PCIe Segment:Bus:Device:Function identifier.
- The IOMMU identifier the RP is bound to.
- The list of all MMIO ranges routed throught that RP.
- The RP ECAM base address.
- All downstream PCIe Endpoints linked to that RP, identified by their PCIe RID (i.e. the device PCIe Bus:Device:Function triplet).

TODO: More precise CoVE-IO manifest format.

3.2.3. Software

Host

To support extending TVMs with CoVE-IO devices, the untrusted domain software stack must:

- Implement the [CoVE] Host Extension (`COVH`).
- Support the RISC-V [IOMMU] programming interface with an IOMMU driver.
- Implement the CoVE-IO host ABI, as described in Chapter 8 of this document.

TSM

The trusted Domain Security Manager, i.e. the TSM, is the trusted intermediary between the untrusted domain and the TVM. To allow for securely assigning TDIs into TVMs, it must:

- Support the [CoVE] Host Extension (`COVH`).
- Implement the [CoVE] Guest Extension (`COVH` and `COVG`).
- Support the RISC-V [IOMMU] programming interface with an IOMMU driver.
- Support the CoVE-IO host ABI, as described in Chapter 8 of this document:
  - Implement the SPDM requester protocol and flows.
  - Implement the TDISP requester protocol and flows.
  - Implement the PCIe IDE Key Management protocol.
- Implement the CoVE-IO guest ABI, as described in Chapter 8 of this document.

3.3. Guest

A TVM guest must verify and explictly accept any TDI into their TCBs. The TSM prevents both TDIs from directly accessing the TVM confidential memory and the TVM from doing memory mapped I/O with TDIs, unless the TVM guest accepts the TDI.

By implementing the CoVE-IO guest ABI, the TSM allows for a TVM guest to verify the trustworthiness of an assigned TDI. The TVM also uses the same ABI to notify the TSM about its TDI acceptance decision.

The TDI verification process from the TVM guest not only requires support from the TSM through the CoVE-IO guest ABI but may also include running local or remote attestation of the physical device the assigned TDI belongs to. In order to minimize the TVM guest software stack changes needed to support the CoVE-IO TDI verification, attestation and acceptance flows, the CoVE-IO guest must run a Trusted Device Manager (TDM) as a separate TVM guest process. Although the TDM can be architectured in a TEE-agnostic fashion, it must support the CoVE-IO guest ABI.

Chapter 4. Architectural Overview

Chapter 5. Security Model

The CoVE-IO securiy model is built on the following assumptions:

- The host platform physical devices are owned by untrusted domain software components (e.g. the host VMM or the hypervisor) that are not part of any TVM TCB and are thus untrusted by TVMs.
- Only the TVM owner can assess of a TDI trustworthiness. Based on that assessment, it explicitly accepts or rejects a TDI into its TCB.
- A TDI may access a TVM confidential memory through DMA only when all the following conditions are met:
    - The TVM owner has explicitly allowed the TDI to access its confidential memory by accepting it.
    - The TDI is exclusively assigned to the TVM, i.e. it must not be shared other TVMs or any host software component.
- A TVM may access a TDI trusted MMIO space only when all the following conditions are met:
    - The TVM owner has explicitly accepted the TDI.
    - The TDI is exclusively assigned to the TVM, i.e. it must not be shared other TVMs or any host software component.
- Until a TDI is accepted by the TVM:
    - The TDI is not allowed to DMA into the TVM confidential memory.
    - Trusted MMIO access to the TDI is blocked by TSM.
    - Untrusted MMIO access to the TDI may still be allowed by the VMM for ordinary VMs (if and only if the TDI is in the TDISP unlocked state).

By means of the CoVE-IO guest ABI, TVMs are required to explicitly accept TDIs into their TCBs. Accepting or rejecting a TDI is a TVM specific decision, based on TVM specific set of verification policies and criteria. A TVM accepting a TDI does not imply that other TVMs on the same host platform would accept it as well.

Shareable CoVE-IO-compliant devices may expose multiple TDIs, assigned to different TVMs. It is the DSM responsibility to guarantee isolation between all assigned TDIs, on a per-TVM basis.

Each TDI must be exclusively assigned to no more than one TVM. However, a single TVM can simultaneously be assigned several TDIs, irrespective of whether they originate from the same physical device or not.

As a TVM TCB does not include PCIe switches and bridges, a selective PCIe IDE stream must be setup to guarantee end-to-end confidentiality and integrity protection between a TVM and a TDI. The host VMM is responsible for reserving a single selective IDE stream per physical device from which one or more TDIs are assigned to TVMs. That single PCIe IDE stream is then shared by all all TDIs originating from the same corresponding physical device. Since the host VMM is not in the TCB, its role in the stream setup is limited to selecting an available stream ID and configuring the device IDE capability. The selective IDE stream keys, for both link endpoints, are managed and configured by TCB elements (the TSM, DSM and platform RoT).

The main security objective of the CoVE-IO security model is to protect a TVM's confidential data integrity and confidentiality while TDIs are assigned to it. At the same time, availability of those assigned TDIs is out of this model scope as e.g. the host VMM could remove them from the TVM at any time it sees fit. Either the DSM or the TDI itself must clear and wipe all TVM confidential data in the TDI before the host software stack can fully reclaim an assigned TDI.

Chapter 6. Threat Model

6.1. Assets

The CoVE-IO security model aims at protecting the following assets:

1. The TVM confidential data, which includes its confidential main memory, code and execution state. A TVM execution state includes both its CPUs micro-architectural states and its assigned TDIs states.
2. An assigned TDI trusted I/O space that is mapped into the TVM address space, also known as a TDI trusted MMIO.

6.1.1. Security Objectives

The CoVE-IO security model objectives is to protect the above-described TVM assets **confidentiality** and **integrity** from components outside of the TCB.

Availability of these assets is out of the CoVE-IO security model scope.

6.2. Adversary Model

The CoVE-IO security model aims at protecting the above-described TVM assets from the following adversaries:

- *Privileged host software adversary*: This includes host software components executing in S and M mode like the host firmware, kernel, hypervisor, VMM, etc. As the system resource owner but also the TVMs lifecycle manager, those components can access and control all devices on the system.
- *Unprivileged host software adversary*: This includes host software components executing in U mode like e.g. the userspace parts of the host VMM.
- *Privileged guest software adversary*: This includes guest software components executing in VS mode like e.g. the guest kernel.
- *Unprivileged guest software adversary*: This includes guest software components executing in VU mode like e.g. the guest application workload.
- *Device firmware adversary*: This includes any firmware driving a device within the system's PCIe topology. As PCIe transaction generators those device firmware components can gain direct access to a TVM confidential memory.
- *Simple hardware adversary*: This includes adversaries that are able to probe visible buses on the motherboard, use JTAG based debuggers, power cycle the system, subject the system to thermal radiation.
- *Advanced hardware adversary*: This includes adversaries that, in addition to the simple hardware adversary capabilities, can also probe high speed buses, place interposers on visible buses, glitch clocks and voltage rails.

6.3. Threats

6.3.1. CoVE-IO-T001 - Trusted MMIO Malicious Access

Table 1. CoVE-IO-T001

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper and Disclosure	Privileged host software	In scope	Host component reads TVM confidential data

Description

A privileged host software component programs a TDI that is assigned to a TVM. By accessing the device MMIO space, the host component can program direct memory access destination addresses to either its own address space or unintended parts of the TVM address space.
Device generated data intended to be copied to the TVM confidential memory is respectively accessed by the host component instead or redirected to unintended parts of the TVM address space.

Mitigations

The `CoVE-IO-T001` threat can be addressed by preventing untrusted domain software components from accessing an assigned TDI, as follows:

- With TEE-I/O, a PCIe root port generates TLPs with the T-bit set only if the MMIO access originates from the trusted domain. Untrusted domain MMIO accesses must have the T-bit cleared.
- A TDI is assigned to a TVM when the TVM accepts it into its TCB, by notifying the TSM about it.
- The TEE-I/O DSM enforces that:
  - Before it is assigned to a TVM, a TDI must not directly access the TVM confidential memory.
  - Once assigned to a TVM, a TDI is in either the `LOCKED` or `RUN` TDISP state.
  - In both the `LOCKED` and `RUN` TDISP state, a TDI trusted MMIO space can only be accessed by a trusted domain generated TLP (T-bit set), through the TDI bound PCIe selective IDE stream.

6.3.2. CoVE-IO-T002 - Trusted MMIO Remapping

Table 2. CoVE-IO-T002

Asset	Threat	Adversary	Scope	Result
Device trusted MMIO	Tamper	Privileged host software	In scope	TVM programs a TDI that is unassigned to it

Description

A privileged host software component remaps a TVM assigned TDI MMIO guest physical address to an unassigned TDI MMIO host physical address.
The TVM programs a different TDI than the one that is assigned to it.

Mitigations

<table>
<tr><th>Asset</th><th>Threat</th><th>Adversary</th><th>Scope</th><th>Result</th></tr>
<tr><td colspan="5">The CoVE-IO-T002 threat can be addressed as follows:
• The TSM maintains second stage page tables (from trusted domain physical addresses to untrusted host domain physical addresses) in confidential memory.
• The untrusted domain software component must not set the second stage mappings for the TDI trusted MMIO. It can requests the TSM to do so on its behalf, through the CoVE-IO host ABI.
• The TSM must not enable Trusted MMIO mappings for an assigned TDI until the TVM accepts it.
• The TVM receives the TDI device interface report through TDISP, via the the TSM CoVE-IO guest ABI. This report is trusted by the TVM and contains the trusted MMIO ranges and order in which they must be mapped to the TVM address space.
• The TVM must explicitly accept the reported MMIO ranges, and the TSM must not enable them until they are accepted by the TVM.</td></tr>
</table>

6.3.3. CoVE-IO-T003 - Trusted MMIO PCIe Redirection

Table 3. CoVE-IO-T003

<table>
<tr><th>Asset</th><th>Threat</th><th>Adversary</th><th>Scope</th><th>Result</th></tr>
<tr><td>Device Trusted MMIO</td><td>Tamper</td><td>Privileged host software</td><td>In scope</td><td>TVM accesses an unassigned TDI trusted MMIO space</td></tr>
<tr><th colspan="5">Description</th></tr>
<tr><td colspan="5">A privileged host software component configures PCIe switches to redirect (or drop) MMIO accesses from the TVM to one of its assigned TDIs.
The host software component can trick the TVM into tampering with an untrusted device or an unassigned TDI MMIO.</td></tr>
<tr><th colspan="5">Mitigations</th></tr>
<tr><td colspan="5">The CoVE-IO-T003 threat can be addressed as follows:
• PCIe switches must not be included in the TVM trust boundary. This is achieved by only allowing PCIe selective IDE streams to be established between a physical device and the untrusted host domain.
• Although the VMM can tamper with the device IDE extended capabilities, the PCIe root port IDE settings must only be available to a TVM TCB component, either the TSM or a hardware root-of-trust.</td></tr>
</table>

6.3.4. CoVE-IO-T004 - Trusted MMIO PCIe Pre-Configuration

Table 4. CoVE-IO-T004

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper and Disclosure	Privileged or unprivileged host software	In scope	Guest software reads and writes resp. from and to another TVM confidential memory
Description				
The VMM maliciously pre-configures a TDI trusted MMIO and assigns it to a TVM. If either the TVM accepts the TDI as-is into its TCB, or the TDI is made operational before the TVM accepts it, the TDI can now access or tamper with the TVM confidential data on behalf of the host software component.				
Mitigations				
TBD				

6.3.5. CoVE-IO-T005 - Trusted MMIO Unauthorized Access

Table 5. CoVE-IO-T005

Asset	Threat	Adversary	Scope	Result
Device trusted MMIO	Tamper	Privileged host software	In scope	TVM accesses an unassigned TDI trusted MMIO space
Description				
A privileged host software component maps a TDI trusted MMIO space into TVM1 as part of the TDI assignement. Then it unassigns the TDI from TVM1 and assigns it to TVM2, withouth unmapping the TDI trusted MMIO space from TVM1. TVM1 can tamper with a TDI trusted MMIO while it is not assigned to it.				
Mitigations				
TBD				

6.3.6. CoVE-IO-T006 - PCIe Link Man-In-The-Middle

Table 6. CoVE-IO-T006

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper and Disclosure	Advanced hardware	In scope	A hardware adversary probes or places an interposer on the PCIe physical link between a TVM and its assigned TDI

Asset	Threat	Adversary	Scope	Result
Description				
A skilled hardware adversary with system physical access probes or places an interposer in the PCIe physical link. It can then eavesdrop, replay or event tamper with a TVM confidential data.				
Mitigations				
TBD				

6.3.7. CoVE-IO-T007 - PCIe ID Spoofing

Table 7. CoVE-IO-T007

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper and Disclosure	Device firmware	In scope	Host software reads and writes from and to a TVM confidential memory
Description				
A device firmware spoofs a PCIe Requester ID (RID) to generate PCIe packets with an existing, assigned TDI RID and get direct memory access to the corresponding TVM confidential memory.				
Mitigations				
TBD				

6.3.8. CoVE-IO-T008 - Confused Deputy DMA Remapping

Table 8. CoVE-IO-T008

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper and Disclosure	Privileged guest software	In scope	Guest software reads and writes resp. from and to another TVM confidential memory
Description				
TVM1 and TVM2 are assigned resp. TDI1 and TDI2. TDI1 and TDI2 belong to the same physical device. TVM1 programs TDI1 with TVM2's address space. TVM2 confidential memory is accessed by an unassigned TDI.				
Mitigations				
TBD				

6.3.9. CoVE-IO-T009 - DMA Remapping

Table 9. CoVE-IO-T009

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper and Disclosure	Privileged host software	In scope	Host software reads and writes from and to a TVM confidential memory
Description				
The privileged host software component manipulates an assigned TDI guest physical address (GPA) to host physical address (HPA) mappings. The TDI direct memory access to and from the TVM confidential data is then redirected to the host software component address space, allowing it to eavesdrop or tamper with the TVM confidential data.				
Mitigations				
TBD				

6.3.10. CoVE-IO-T010 - DMA Remapping

Table 10. CoVE-IO-T010

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Tamper	Privileged host software	In scope	TDI writes into unintended portions of a TVM confidential memory
Description				
The privileged host software component manipulates the guest physical address (GPA) to host physical address (HPA) mappings to create inconsistencies between the TVM and its assigned TDI mappings for the same GPA ranges. The TDI writes physical adresses that are different than the ones the TVM programmed it with, and tampers the TVM confidential memory. Moreover, the TVM memory reads from the intended GPA return results that are inconsistent with the actual device operation.				
Mitigations				
TBD				

6.3.11. CoVE-IO-T011 - TDI Denial of Service

Table 11. CoVE-IO-T011

Asset	Threat	Adversary	Scope	Result
TVM confidential data	Denial of service	Privileged host software	**Not** in scope	TVM can not access a TDI that is assigned to it
Description				
A privileged host software component resets or powers down an assigned TDI or its physical device, while the TDI is assigned to a TVM. The TVM is no longer able to directly access its assigned TDI.				
Mitigations				
TBD				

6.4. Requirements

List CoVE-IO security requirements to address the threat model.

Chapter 7. Theory of Operations

Chapter 8. Device Attestation

Chapter 9. Confidential VM Extension (CoVE) IO ABI

Index

References

- [CoVE] RISC-V Confidential VM Extension, github.com/riscv-non-isa/riscv-ap-tee
- [CC] Confidential Computing, confidentialcomputing.io/about/
- [PCI-SIG] PCI SIG, pcisig.com/
- [DMTF] Distributed Management Task Force, www.dmtf.org/
- [SPDM] Secure Protocol and Data Model Specification, www.dmtf.org/dsp/DSP0274
- [SecuredSPDM] Secured Messages using SPDM Specification, www.dmtf.org/dsp/DSP0277
- [PCIE] PCI Express Base Specification, members.pcisig.com/wg/PCI-SIG/document/18363
- [TDISP] TEE Device Interface Protocol ECN, members.pcisig.com/wg/PCI-SIG/document/18268?uploaded=1
- [IOMMU] RISC-V IOMMU, github.com/riscv-non-isa/riscv-iommu/blob/main/riscv-iommu.pdf