Intel® TDX Virtual Firmware Design Guide

December 2023

Document Number: 344991-004US

Document Number: 344991-004US

Contents

Figures

Tables

1 Introduction

1.1 Background

Intel Total Memory Encryption (TME) engine encrypts the platform's entire memory with a single key and provides the ability to specify use of a specific key for a page of memory. The Multi-Key mode of TME (MK-TME) extends TME to support multiple encryption keys. In a virtualization scenario, a Virtual Machine Manager (VMM) or hypervisor will manage keys to transparently support legacy operating systems without any changes (thus, MKTME can also be viewed as TME virtualization in these scenarios). An operating system (OS) may take advantage of MKTME capabilities in a native or virtualized environment. When properly enabled, MKTME is available to each guest OS in a virtualized environment, so both native and guest OS can take advantage of MKTME. In these usages, the VMM is in the Trust Computing Base (TCB).

Intel Trust Domain Extensions (Intel TDX) refers to an Intel technology that extends Virtual Machines Extensions (VMX) and Multi-Key Total Memory Encryption (MKTME) with a new kind of virtual machine guest called a Trust Domain (TD). A TD runs in a CPU mode that protects the confidentiality of its memory contents and its CPU state from any other software, including the hosting Virtual Machine Monitor (VMM), unless explicitly shared by the TD itself.

The TDX solution is built using a combination of Intel® Virtual Machine Extensions (VMX) and Multi-Key Total Memory Encryption (MK-TME), as extended by the Intel® Trust Domain Instruction Set Architecture Extensions (TDX ISA). An attested software module called The Intel TDX module implements the TDX architecture.

The platform is managed by a TDX-aware host VMM. A host VMM can launch and manage both guest TDs and legacy guest VMs. The host VMM maintains all legacy functionality from the legacy VMs perspective; it's restricted only with regard to the TDs it manages.

In Summary, Intel TDX:

1) Removes the cloud host software and devices from the Trust Computing Base (TCB) of cloud (TD) tenants.

2) Provides memory encryption and integrity multi-tenancy for hardware attack protection.

3) Supports TD measurement for attestation to a relying party.

1.2 Overview

Intel® Trust Domain Extensions (Intel® TDX) provides the capabilities required to enable an TDX-aware VMM to manage the lifecycle of a TD.

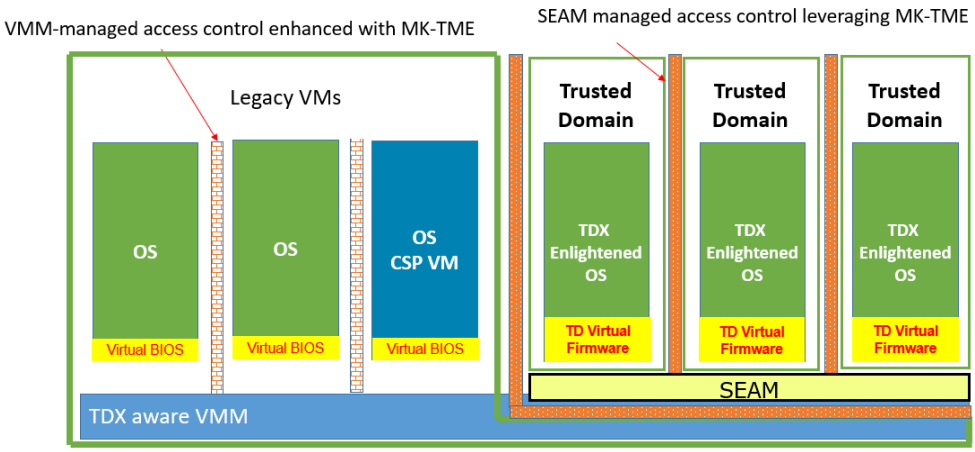

Figure 1-1: Intel TDX Architecture

Fundamental differences between Virtual Machine Extension (VMX) and Intel TDX are outlined in the following table.

Table 1-1: Differences between VMX and TDX

	Access to Guest State	Transitions for State Save & Restore	Controls
VMX	VMM has full access	VMM SW augments VMX transitions (e.g. general-purpose registers)	VMM has a rich set of controls to interact with the VM in order to de-privilege the VM
TDX	VMM has no direct access; TD may volunteer info	State transitions completed are managed by Intel TDX Module in Secure-Arbitration Mode (SEAM)	VMM has a limited set of controls for resource management

Trust Domain Virtual Firmware (TDVF) is required to provide TD services to the TD Guest OS. This document describes the TDVF architecture.

TDVF reference code is integrated to TianoCore open source repository: https://github.com/tianocore/edk2/tree/master/OvmfPkg. A minimal TDVF (td-shim) reference code is at confidential container repository: https://github.com/confidential-containers/td-shim.

1.3 Terminology

Table 1-2: Terminology

Term	Description
ACPI	Advanced Configuration and Power Interface

AP	Application Processor
APIC	Advanced Programmable Interrupt Controller
BFV	Boot Firmware Volume
BSP	Bootstrap Processor
CFV	Configuration Firmware Volume
CSM	Compatibility Support Module
DXE	Driver Execution Environment
EPT	Extended Page Table
GPA	Guest Physical Address
GPAW	Guest Physical Address Width
HOB	Hand Off Block
HPA	Host Physical Address
MADT	Multiple APIC Description Table
MKTME	Multi-Key Total Memory Encryption
MPWK	Multiple Processor Wakeup
MRCONFIGID	Measurement Register of configuration data
MRTD	TD Measurement Register
OVMF	Open Virtual Machine Firmware
PEI	Pre-EFI Initialization
RA-TLS	Remote Attestation-TLS
RTMR	Runtime Extendable Measurement Register
SEAM	Secure-Arbitration Mode
SEC	Security Phase
SMM	System Management Mode
SMX	Safer Mode Extensions
TCB	Trust Computing Base
TCG	Trusted Computing Group
TD	Trust Domain
TDVF	Trust Domain Virtual Firmware
TDX	(Intel) Trust Domain Extensions
TLS	Transport Layer Security
TME	Total Memory Encryption
UEFI	Unified Extensible Firmware Interface
VCPU	Virtual CPU
VE	Virtualization Exception
VM	Virtual Machine

VMCS	Virtual Machine Control State
VMM	Virtual Machine Monitor
VMX	Virtual Machine Extension

§

2 Architectural Overview

2.1 TDVF Features

TDVF has the following features:

1) Use Unified Extensible Firmware Interface (UEFI) Secure Boot as base with extensions for TD launch.

2) Use Trusted Computing Group (TCG) Trusted Boot to perform a measured and verified launch of a guest OS loader or kernel.

3) Simplify firmware by removing features found in traditional UEFI implementations:

    a) SEC, PEI, SMM (DXE Only)

    b) CSM (UEFI Class 3 OS only)

    c) Setup UI

    d) Recovery

    e) Capsule-based Firmware Update

    f) ACPI S3 (not supported by TDX guests)

2.2 TD Hardware

A TD is based on the following hardware:

1) CPU: x2APIC

2) CPU: xFLUSH, STI, CLI, LIDT, LGDT instruction are allowed.

3) VMM-specific Virtual Device (block device, console, network). This is highly dependent on the hypervisor configuration (e.g. VirtIo device for KVM/XEN, Vmbus device for Microsoft Hyper-V).

4) Hot Plug – CPU and memory hotplug are not supported now. Device hotplug is out of scope of this document.

5) TD may or may not support below feature. If it is supported, the device must be access via TDCALL [TDG.VP.VMCALL] interface instead of direct hardware access. For example, the IO access needs to be replaced by TDCALL [TDG.VP.VMCALL] <INSTRUCTION.IO>.

    a) Persistent NV storage.

    b) The emulated physical device. (Graphic, Keyboard, Storage, etc.)

c) I/O Subsystems (PCI, USB, ISA, DMA, IOMMU, PIC, PIT, etc.). For example, the PCI might only be used to emulate the VirtIo device.

d) MMIO (APIC, HPET)

e) vTPM

2.3 Boot Flow

A TD launch takes below steps:

1) VMM sets up TDVF, calls Intel TDX module to create the initial measurement, then calls Intel TDX module to launch TDVF.

2) TDVF boots and enables UEFI Secure Boot.

3) TDVF prepares CC event log and launches the OS loader.

2.3.1 VMM Setup Phase

The TDVF image includes firmware code, which is measured into TD Measurement Register (MRTD). The TDVF image may also include static configuration data, including UEFI Secure Boot certificates (PK/KEK/db/dbx). The VMM provides dynamic configuration data, including the hand-off block (HOB) list as a parameter for the entrypoint.

The VMM calls Intel TDX module to initialize TD memory. This includes firmware code and UEFI Secure Boot configuration read-only data captured by the tenant:

- Intel TDX module associates memory via Extended Page Table (EPT) with the TD guest.
- Intel TDX module associates logical processors with TD guest via TD-Virtual Machine Control State (VMCS).
- Intel TDX module performs TDENTER instruction on all processors, including Bootstrap Processor (BSP) and Application Processors (APs).

2.3.2 TDVF Launch Phase

TDVF is launched on all processors:

- All processors start in 32-bit protected mode with flat descriptors (paging disabled).
- The CPU with VCPU_INDEX 0 is elected as BSP, the other CPUs are APs.
- Startup code switches to 4-level paging enabled (0-4GB). Option for startup code to switch to 5-level paging enabled (0-4GB).
- BSP performs Virtual Firmware initialization and determines how many APs to wake via TDCALL [TDG.VP.INFO].