Intel® Trust Domain Partitioning-based Virtual Trust Platform Module (vTPM)

Design Guide

July 2024

Contents

Figures

Tables

1 Introduction

1.1 Background

A Trust Platform Module (TPM) provides the Root-of-Trust for Reporting (RTR) and Root-of-Trust for Storage (RTS) for a computer platform. With a platform-specific Root-of-Trust for Measurement (RTM), usually in the first boot code, the platform can support TPM-based attestation or TPM-based sealing.

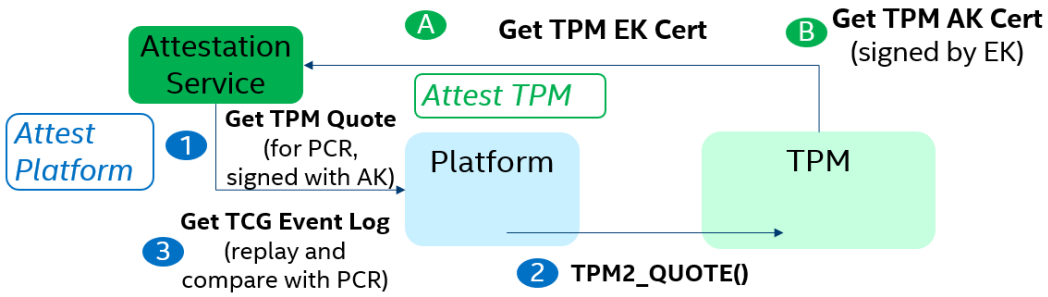

Figure 1: TPM-based Attestation

Figure 1: TPM-based Attestation shows a typical platform attestation with TPM. In a virtual platform, a virtual TPM (vTPM) may be used to support similar TPM-based attestation. It is widely adopted in the hypervisor environment and supported by the virtual machine monitor (VMM) vendor. A VMM may provide virtual TPM services to the guest environment.

However, with the Intel Trust Domain Extension, the VMM is out of Trust Computing Base (TCB) and no longer trusted. As such, a pure VMM-based virtual TPM is not feasible. We need to have another way to support a vTPM-based solution.

1.2 Overview

In this specification, we will describe a Trust Domain Partitioning (TD Partitioning) based vTPM solution, which can support vTPM functionality with VMM out of a TCB.

This document describes the design of the vTPM TD to support a TPM-based attestation use case. The TPM-based sealing use case is not covered in this document, because TDX architecture does not support sealing capability.

1.3 Terminology

Table 1: Terminology

Term	Description
CRB	Command-Response Buffer
DMA	Direct Memory Access
GHCI	Guest Hypervisor Communication Interface
L1	Layer 1 TD Partitioning software, also knowns as L1-VMM in TD

Term	Description
L2	Layer 2 TD Partitioning software, also knowns as L2-OS in TD
MMIO	Memory Mapped Input/Output
MR	Measurement Register
MRTD	Measurement Register for TD
NVS	Non-volatile Storage
RA-TLS	Remote Attestation TLS
PCR	Platform Configuration Register
PFP	Platform Firmware Profile
PTP	Platform TPM Profile
RTM	Root-of-Trust for Measurement
RTMR	Runtime Measurement Register
RTR	Root-of-Trust for Reporting
RTS	Root-of-Trust for Storage
SEAM	Secure Arbitration Module
SRTM	Static Root-of-Trust for Measurement
SVSM	Secure VM Service Module
TCB	Trust Computing Base
TD	Trust Domain
TD Partitioning	Trust Domain Partitioning
TDVF	Trust Domain Virtual Firmware
TDX	Trust Domain Extension
TPM	Trust Platform Module
VMM	Virtual Machine Monitor
vTPM	Virtual TPM

§

2 vTPM Architectural Overview

2.1 vTPM Architecture Solution

A vTPM can be supported in many ways, such as TD-based vTPM or TD Partitioning-based vTPM. See *Figure 2: vTPM Solutions.*

We do not consider an Intel TDX module as an option for vTPM, because of the following limitations:

- Complexity. It makes TCB larger.
- Long latency to generate a key in TPM.
- NV storage dependency.

This document will focus on the TD Partitioning-based solution (Option 2).

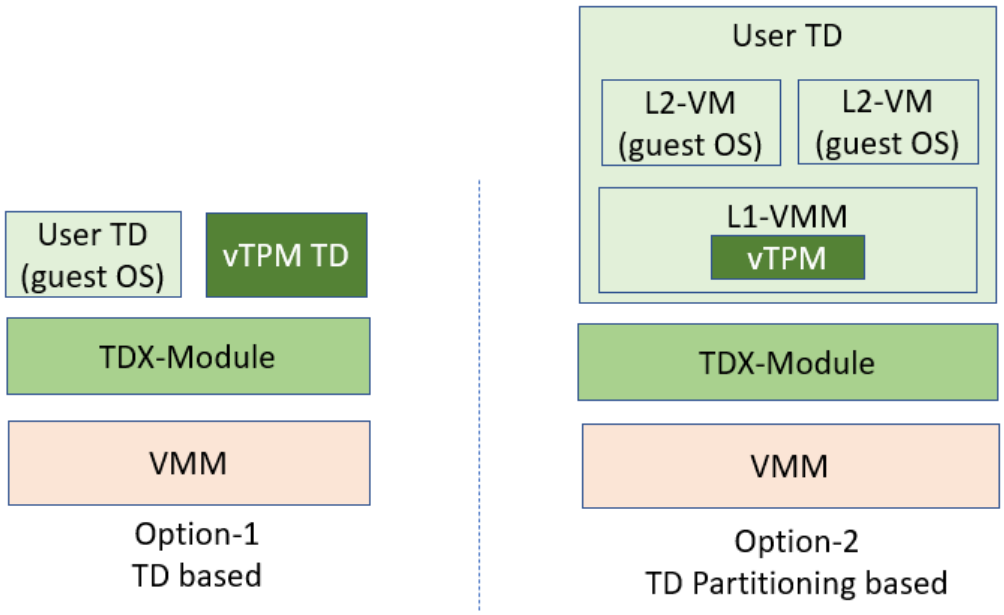

Figure 2: vTPM Solutions

2.1.1 RoT Roles in TD based and TD Partitioning based solutions

The following table shows the Root of Trust (RoT) roles in TD based and TD Partitioning based vTPM solutions.

Table 2: Roles in TD based and TD Partitioning based solution

Role	vTPM TD (Opt-1)	TD Partitioning vTPM Service (Opt-2)
Virtual Root of Trust for Reporting (vRTR)	**vTPM TD**: TPM software stack.	**vTPM Service**: TPM software stack.
Virtual Root of Trust for Storage (vRTS)	**vTPM TD**: maintain the ephemeral NV storage inside of TD. The NV storage does not exist after the vTPM TD is shutdown.	**vTPM Service**: maintain the ephemeral NV storage inside of TD. The NV storage does not exist after the whole TD is shutdown.
Virtual Root of Trust for Measurement (vRTM)	**TDX-module**: create MRTD. create TDREPORT during mutual authentication. **vTPM TD**: extend TDREPORT to PCR[0] as evidence of initial boot code, after authentication.	**L1-VMM**: extend initial boot block to PCR[0].

NOTE: TPM based attestation information PCR[0] only records the measurement at the system boot time. A TCB update (such as TDX-module update) will not cause the PCR[0] change. This design is aligned with current [PFP]. The TCB update is similar to the TPM upgrade, which does not impact TPM EK. Only an explicit TPM2_ChangeEPS() will cause TPM EK change. At that time, the TPM EK will be regenerated.

2.2 vTPM Service requirement

2.2.1 General Requirement

1. The solution SHALL work in existing TDX 1.5 with TD Partitioning. One TD SHALL include one L1-VMM and MAY include multiple L2 User VM(s). The vTPM service SHALL be provided by L1-VMM.
2. The solution SHALL follow TPM 2.0 specification. (Not TPM 1.2)
    a. A vTPM service MAY provide partial of TPM services, based upon the real use case. (See 2.2.4 vTPM Feature).
3. The solution SHALL not change any TPM Software Stack (TSS).
    a. The solution MAY enlighten every TPM device driver (such as TDVF or TD-OS).

2.2.2 vTPM Environment

4. vTPM Service SHALL be in an isolated environment, independent from L2 user VM. It SHALL be provided by L1 VMM via L2 service VM or via L1 VMM Ring3 service.
5. vTPM Services SHALL provide one vTPM instance for each L2 VM.
6. vTPM service and its vTPM instances SHALL support migration if the whole TD is migratable.

2.2.3 vTPM Communication

7. L2 User VM SHALL follow TPM2 specification to send/receive TPM2 command in TPM software stack.
8. L2 User VM SHALL use a command-response-buffer (CRB) to send/receive TPM command. The CRB SHALL be private MMIO.

2.2.4 vTPM Feature

9. TPM Crypto primitives SHALL be supported.
10. TPM attestation use case (such as PCR_Extend/Quote) SHALL be supported.
11. Windows BitLocker use case (such as Seal/Unseal) SHALL be supported.
12. TPM defined Non-Volatile Storage (NVS) MAY be ephemeral (NVS disappear after the whole TD shutdown).

2.3 vTPM Launch Flow

A vTPM may include all TPM specification defined features. Besides a cryptographic algorithm, a TPM should include NVS to store the persistent key. However, a TD cannot provide the secure persistent NVS support. As such, the vTPM Service only maintains the ephemeral NVS inside of TD. See *Figure 3: vTPM Instance Launch Flow.*

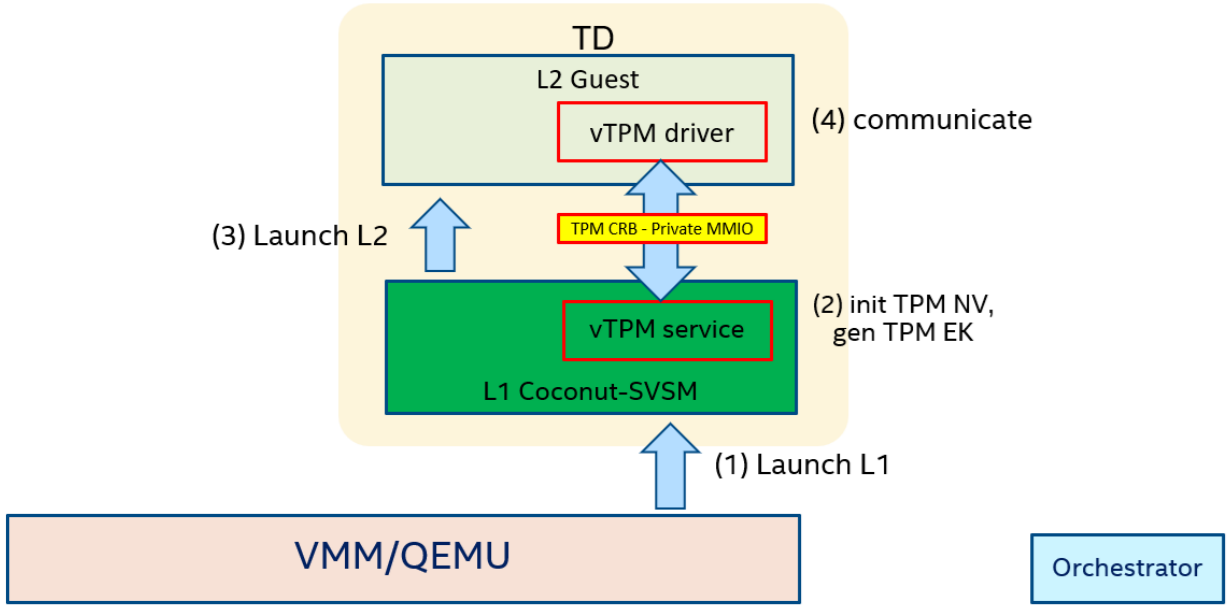

Figure 3: vTPM Instance Launch Flow

1) A VMM launches L1 VMM (here it is [Coconut-SVSM]).

2) L1 VMM launches vTPM service. It creates vTPM instance, initializes vTPM NV and generate vTPM EK.

3) L1 VMM launches L2 User VM.

4) The TPM driver in L2 User VM communicates with vTPM Service in L1 VMM via private MMIO based TPM CRB interface.

2.4 Coconut-SVSM Launch Flow

Figure 4: Coconut-SVSM Launch Flow shows the [Coconut-SVSM] launch flow.

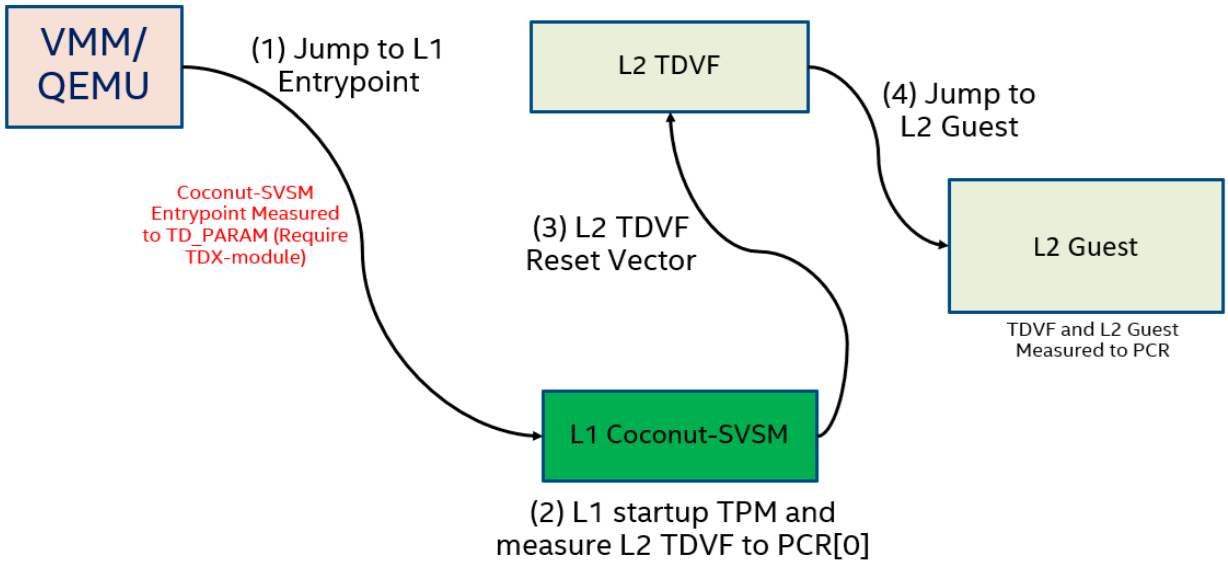

Figure 4: Coconut-SVSM Launch Flow

1) VMM/QEMU starts up L1 Coconut-SVSM.

1.1) The TD Launch process measures L1 Coconut-SVSM to MRTD.

1.2) Host-VMM asks TDX-module to launch L1 Coconut-SVSM with alternative Entrypoint (Alternative Entrypoint included in TD_PARAM).

2) L1 Coconut-SVSM starts up vTPM.

2.1) L1 Coconut-SVSM extends Launch Parameter to RTMR[0] (Legacy TDX 1.0 feature. The Launch Parameter is TD_HOB, please refer to [TDVF]).

2.2) L1 Coconut-SVSM initializes the ephemeral vTPM NVS.

2.3) L1 Coconut-SVSM generates ephemeral vTPM EK, including the TD Quote and Event Log. (For vTPM EK verification)

2.4) L1 Coconut-SVSM invokes TPM2_Startup().

2.5) L1 Coconut-SVSM extends SVSM-Version to PCR[0]. (Following PFP specification)

2.6) L1 Coconut-SVSM extends L2 TDVF to PCR[0]. (As the role of vRTM).

2.7) L1 Coconut-SVSM extends SEPARATOR to RTMR[0~3]. (Changing RTMR to avoid L2 vTPM emulation attack)

3) L1 Coconut-SVSM jumps to L2 TDVF reset vector - 0xFFFFFFF0.

3.1) L2 TDVF create TCG event log for SVSM-Version and L2 TDVF measurement.

3.2) L2 TDVF invokes TPM2_Extend() for rest of L2 component and creates the corresponding TCG event log.

4) L2 TDVF follows the normal VM boot flow to measurement the next level components and jumps to L2 Guest OS.

2.5 Trust Relationship

The whole L1 VMM and vTPM Service are the TCB for the L2 user VM, if the vTPM is used by L2 user VM.

The trust relationship between L2 user VM and vTPM Service is described in the following sections.

2.5.1 L2 User VM

A L2 user VM trusts a vTPM Service provided by the L1 VMM as vRTS and vRTR.

A L2 user VM trusts the L1 VMM as vRTM.

If the L1 VMM or vTPM Service is malicious, then the user TD does not know. But the verifier can detect that when it verifies the TD Quote (including MRTD and RTMR) in the vTPM EK certificate.

2.5.2 vTPM Service and L1 VMM

A vTPM Service and L1 VMM does not trust a user TD. L1 VMM shall prevent TPM CRB interface attack from L2. vTPM Service shall prevent TPM command attack from L2.

vTPM Service and L1 VMM only trust the TDX-module and feature specific TCB, such as Migration TD if the TD is migratable.

To prevent L2 from forging the measurement, L1 VMM shall extend the initial code of L2 TDVF to PCR[0].

If L2 TDVF is malicious, it can be detected by the verifier via vTPM PCR check.

To prevent L2 from emulating the vTPM Service to verifier, L1 VMM shall extend the separator to RTMR before launch to L2. This can ensure that the TD Quote generated by L2 will be different from the TD Quote in vTPM EK certificate.

If L2 guest emulate vTPM Service, it can be detected when the verifier checks TD Quote in the vTPM EK certificate.

2.5.3 vTPM Service <-> L2 User VM binding

To consider vTPM instance NVS* data binding for a user VM: for a physical TPM, there is no binding between HDD and TPM. This is similar for a virtual TPM. There is no binding between virtual HDD (storage) and vTPM.

There is a common request to provide a persistent binding between VM/TD and vTPM. That typically means to bind HDD and vTPM, which is not offered by the vTPM solution.

NOTE: in virtual TPM, we do not provide more security binding properties than physical TPM. This is similar to the following cases:

- Moving an OS disk to another machine.
- Booting another OS on the same machine.

For example, the VMM can launch user VM-1 with vTPM-1, and user VM-2 with vTPM-2. Then VMM shutdowns all and launces user VM-2 with vTPM-1.

- The attestation is not impacted, because all PCRs reset.

- The sealed data is not impacted, because we assume user VM shall use the correct sealing policy to seal the data, such as TPM2_PolicyPCR() or TPM2_PolicyPassword().

2.6 Combined Attestation

Figure 5: Combined Attestation Flow shows the flow of combined attestation.

Figure 6: Combined Attestation shows the components in the combined attestation.

- L1 VMM / Coconut SVSM / vTPM Service: the TDX attestation.
- L2 Guest / User VM: the standard TPM attestation.

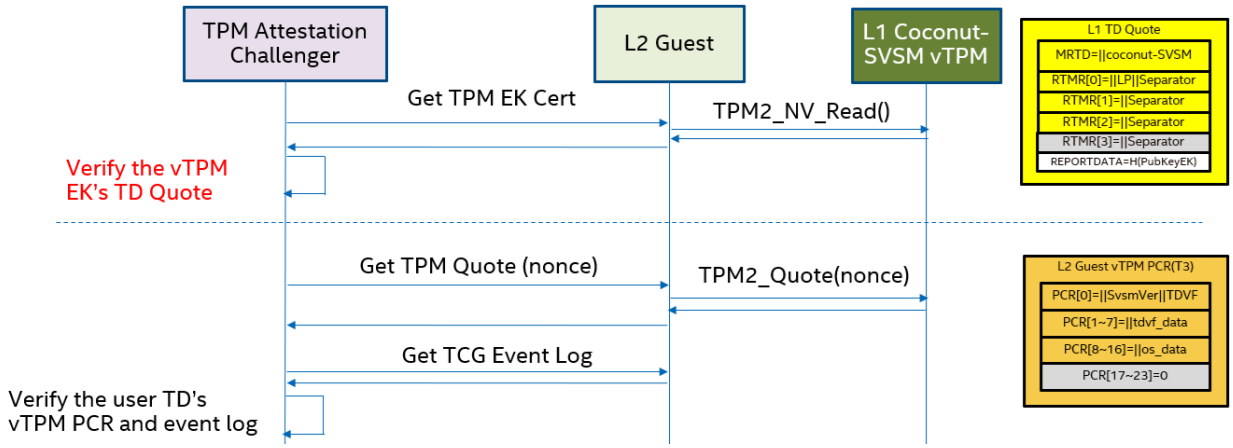

Figure 5: Combined Attestation Flow

NOTE: TPM Quote can provide the freshness of the PCR. [TPM] specification does not provide freshness of the TPM EK by design. A TPM Field Upgrade does not change TPM EK. Only after an explicit TPM2_ChangeEPS() will change TPM EK.

The connection between them is the vTPM Endorsement Key (EK). Every vTPM instance generates a new EK when the instance is created. This TPM EK X.509 certificate will include an Object ID (OID) to indicate the vTPM TD Quote. The REPORTDATA in the TDREPORT is the hash of the vTPM instance EK.

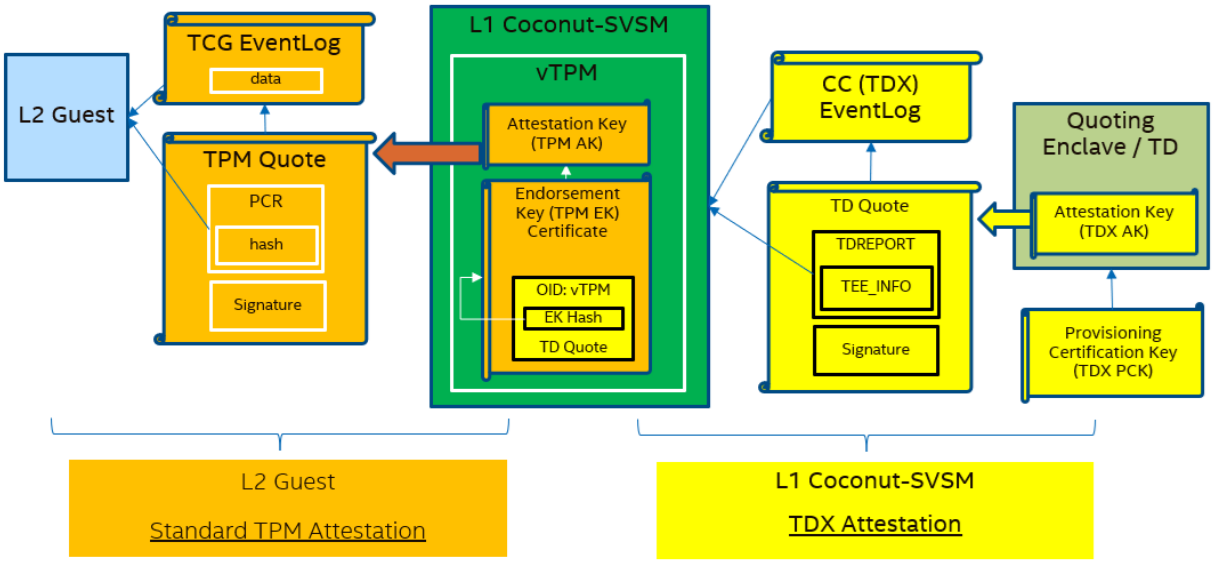

Figure 6: Combined Attestation

2.6.1 L2 User VM measurement register

Figure 7: L2 User VM MR/PCR Change Flow shows how user TDs MR and vTPM PCR are changed in various phases.

- T0 means just after L1 Coconut-SVSM startup, before L1 vTPM startup.
- T1 means just after L2 Guest startup, before L2 vTPM driver startup.
- T2 means L2 Guest TDVF phase after L2 vTPM driver startup.
- T3 means L2 Guest OS phase after L2 Guest TDVF phase.

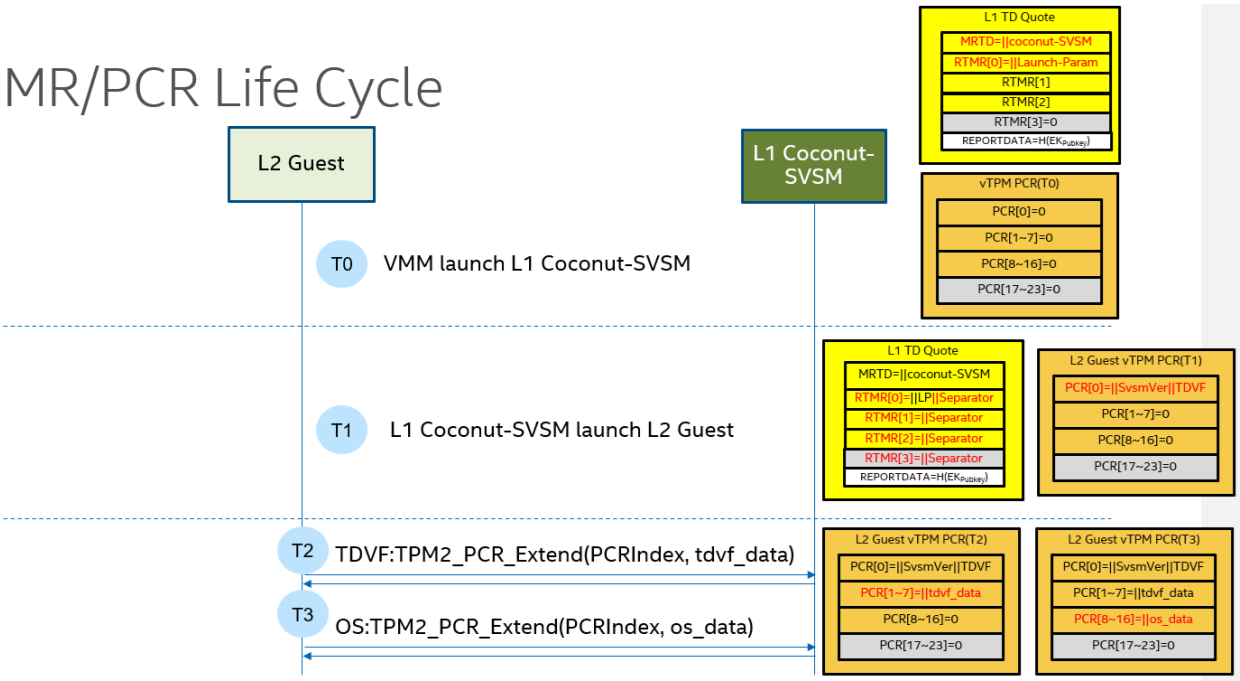

Figure 7: L2 User VM MR/PCR Change Flow

Since the L2 user VM will use vTPM PCR, it should not use RTMR. As such we will change the measurement register value as

Table 3: TD Measurement Register.

The L1 VMM needs to poison all RTMRs to prevent measurement forging before launch L2 user VM.

Table 3: TD Measurement Register

TD Measurement Register	Value
MRTD	No change. **Intel TDX Module extends L1 to MRTD**.
RTMR[0]	L1 extends Launch Parameter. Same as legacy TDX feature. L1 extends SEPARATOR to prevent L2 vTPM emulation attack.
RTMR[1]	L1 extends SEPARATOR to prevent L2 vTPM emulation attack.
RTMR[2]	L1 extends SEPARATOR to prevent L2 vTPM emulation attack.
RTMR[3]	L1 extends SEPARATOR to prevent L2 vTPM emulation attack.

2.6.2 L2 User VM vTPM Platform Configuration Register (PCR)

In general, TDVF shall follow the TCG PFP specification to extend corresponding entries. See *Table 4: vTPM Platform Configuration Register.*

Table 4: vTPM Platform Configuration Register (PCR)

TPM PCR	Value
0	Firmware Code. (Follow TCG PFP spec) TDX specific actions: 1. TDVF gets measurement hash of SVSM-Version and TDVF from L1 and creates the corresponding event log during boot. (Similar to Intel Boot Guard Technology)
1	Firmware Configuration Data. (Follow TCG PFP spec)
2	Option ROM Code. (Follow TCG PFP spec)
3	Option ROM Data. (Follow TCG PFP spec)
4	OS Loader Code. (Follow TCG PFP spec)
5	Boot Configuration. (Follow TCG PFP spec)
6	OEM Specific Data. (Follow TCG PFP spec)
7	Secure Boot Configuration. (Follow TCG PFP spec)
8~15	OS application

2.7 vTPM EK Certificate

There could be two possible vTPM EK certificate modes.

- **vTPM CA self-signed EK**. The vTPM Service CA self-signs the vTPM EK Certificate. The vTPM Service CA certificate includes a vTPM Service Quote for vTPM Service verification. See *Figure 6: Combined Attestation.*
- **Service CA-signed EK**. If there is a vTPM service CA, the vTPM Service can send the vTPM TD Quote and vTPM EK PubKey to the vTPM service CA and get a vTPM EK Certificate signed by the vTPM service CA.

Table 5: vTPM EK Mode

	vTPM CA self-signed mode	Service CA-signed mode
vTPM EK Cert Verification	No change required	No change required
vTPM CA Cert Verification	Required to be enlightened to understand TD Quote	Required to be enlightened to trust the vTPM Service CA

2.8 vTPM Challenge Summary

[TCG VTPM] describes a set of challenges for the virtual TPM implementation. We summarize the solution in the following table.

Table 6: vTPM Challenges in vTPM Architecture Specification

Challenges	Solution
Protecting Virtual TPM Storage	Use L1 VMM in the TD to protect vTPM
Protecting vTPM Secrets across Reboots	No persistent storage. The NVS is inside of vTPM Service. See 7 vTPM NV Storage Management.
Attestation	Use Combined Attestation
Supporting Different vTPM Version	N/A. Only support TPM2.0
Field Upgrade of vTPM	vTPM Service teardown/launch. See 10 vTPM Field Upgrade.
vTPM Backup and Restore	Not supported.
Migration	vTPM Service Migration. See 9 vTPM Migration.

Intel® Trust Domain Partitioning-based Virtual TPM

Since the vTPM is used for confidential computing environment, we have new challenges summarized in the following table.

Table 7: vTPM Challenges in Confidential Computing

Challenges	Solution
Protecting communication between TEE and vTPM	Private MMIO between L1 VMM / vTPM Service and L2 Guest / User VM.

§

3 vTPM Design Overview

3.1 Design Overview

vTPM Service is a component in L1 VMM. vTPM Service SHALL be provided by L1 VMM via L2 service VM or via L1 VMM Ring3 service. Coconut-SVSM prefers to use Ring3 service.

3.1.1 L1 vTPM Service design

vTPM includes below components.

- **TPM Command Lib** provides TPM command process capability. It should reuse the existing known good TPM implementation, such as Microsoft TPM2.0 Reference, https://github.com/microsoft/ms-tpm-20-ref.
- **TPM CRB Lib** provides TPM CRB MMIO process capability.
- **Crypto Lib** is backend for TPM Command lib.
- **vTPM NVS Management** is to manage the ephemeral NVS instance.
- **vTPM EK Management** is to generate the ephemeral EK, including TD Quote and Event Log.

3.1.2 L1 vRTM design

vRTM includes below functions.

- Startup TPM
- Extend SVSM-Version
- Extend TDVF
- Provide API to allow L2 Guest get the measurement hash of SVSM-Version and TDVF.

3.1.3 L1 VMM RTMR measurement

L1 VMM needs to perform below functions.

- Extend TD Launch Parameter to RTMR[0], after L1 VMM is launched.
- Extend SEPARATOR to RTMR[0~3], just before L2 Guest is launched.

§

4 vTPM IO Interface

A L2 user VM SHALL use TPM CRB private MMIO interface to communicates with L1 vTPM Service.

§

5 vTPM L1/L2 Interface

5.1 TDG.VP.VMCALL<Service.L1VTPM>

This function is used to allow L2 TD to invoke L1 vTPM service.

```
#define VMCALL_SERVICE_L1VTPM_GUID \
{0x766cf580, 0x8dc3, 0x4cea, 0xa9, 0x4e, 0xe5, 0x42, 0x4d, 0xa1, 0xda, 0x56}
```

5.1.1 TDG.VP.VMCALL <Service.L1VTPM >

Table 8: User TD TPM <Service.L1VTPM > Command

Field	Offset (Bytes)	Length (bytes)	Description
Version	0	1	0: for this data structure
Command	1	1	1: **Detect**
Reserved	2	2	Reserved

Table 9: User TD TPM <Service.L1VTPM > Response

Field	Offset (Bytes)	Length (bytes)	Description
Version	0	1	0: for this data structure
Command	1	1	1: **Detect**
Status	2	1	Status of the response: 0x0: Reserved. 0x1: The L1VTPM service is present. It sends TPM2_Startup() and extends the initial boot block. The corresponding TCG2 Events are recorded in additional Data field. NOTE: The initial boot block must include the TCG component. 0x2~0xFE: Reserved. 0xFF: The L1VTPM service is not present, or there is something wrong with L1VTPM service.
Reserved	3	1	Reserved
Additional Data	4	Variable	A list of GUIDed HOB - EFI_TCG_EVENT2_HOB_GUID, defined at https://github.com/tianocore/edk2/blob/master/SecurityPkg/Include/Guid/TcgEventHob.h It is present only if Status is 0x1.

§

6 vTPM Profile

vTPM Service follows [TCG PTP] specification, "TPM Attributes" chapter and "TPM Capabilities and Commands" chapter.

6.1 vTPM Attributes

6.1.1 vTPM Algorithms

vTPM Service follows [TCG PTP] specification, "PC Client Algorithms" section and "PC Client Curves" section.

6.1.2 vTPM NVS

vTPM Service follows [TCG PTP] specification, "NV Storage Requirement" section.

The vTPM Service will maintain the ephemeral vTPM NVS inside of vTPM TD.

6.1.3 vTPM EK Certificate

vTPM Service follows [TCG PTP] specification, "Endorsement Key Certificate" section.

The vTPM instance shall be provisioned with EK certificates, following [TPM2 EK] specification. For example, NV Index 0x01c00002 is for RSA 2048 EK Certificate. NV Index 0x01c0000a is for ECC NIST P256 EK Certificate.

The vTPM instance shall be provisioned with the EK certificates chains, following [TPM2 EK] specification. For example, started from NV Index 0x01c00100 till 0x01c001ff.

The vTPM instance EK certificate shall be issued by a vTPM CA, which could be self-signed or a vTPM Service CA. The flow is:

1) vTPM Service creates a private/public key-pair as the vTPM instance EK with TPM2_CreatePrimary().
2) vTPM Service exports the EK public key from the vTPM instance.
3) vTPM TD asks the CA to generate a X.509 certificate and sign the X.509 certificate for the public key as the final vTPM instance EK certificate.
4) vTPM Service writes the vTPM instance EK certificate and the vTPM CA certificate to the TPM NV Index.

In vTPM self-signed mode, the vTPM CA certificate shall be issued by the vTPM Service itself. The flow is:

1) vTPM Service generates a private/public key-pair for this vTPM as the vTPM CA key.
2) vTPM Service uses the hash of CA public key as REPORTDATA and generates a vTPM TD_Quote.
3) vTPM Service generates the X.509 certificate for the CA public key, including OID for the vTPM TD_Quote and vTPM CC event log.

See *Table 10: vTPM CA Certificate Field* for more details.

Table 10: vTPM CA Certificate Field

Field	Description	Required
Version	Version of the encoded certificate shall be present and shall be version 3 (**value 0x2**)	Mandatory
Serial Number	Serial number shall be present with a positive integer value. For example: **Serial Number: 1**	Mandatory
Signature Algorithm	Signature algorithm shall be present. For example: **sha384WithRSAEncryption**	Mandatory
Issuer	Issuer distinguished name shall be specified.	Mandatory
Subject Name	Subject name shall be present and shall represent the distinguished name associated with the certificate. It shall be same as Issuer.	Mandatory
Validity	Certificate may include this attribute. If the validity attribute is present, the value for **notBefore** field should be assigned the generalized **19700101000000Z** time value and **notAfter** field should be assigned the generalized **99991231235959Z** time value.	Mandatory
Subject Public Key Info	Device public key and the algorithm shall be present. For example: **Public Key Algorithm: rsaEncryption** **Modulus: …** **Exponent: 65537 (0x10001)**	Mandatory
X509v3 Extension: Basic Constraints	**CA: TRUE**.	Optional
X509v3 Extension: Subject Key Identifier	Subject Key Identifier	Optional
X509v3 Extension: Authority Key Identifier	It should be same as Subject Key Identifier.	Optional
X509v3 Extension: Extended Key Usage	vTPM TD issued CA Certificate indicator "2.16.840.1.113741.1.5.5.4.5"	Mandatory
OID:TD_Quote	vTPM TD_Quote "2.16.840.1.113741.1.5.5.4.2"	Mandatory
OID:Event_Log	vTPM CC Event Log	Mandatory

Field	Description	Required
	"2.16.840.1.113741.1.5.5.4.3"	

6.1.4 vTPM PCR

vTPM Service follows [TCG PTP] specification, "PCR Requirements" section.

vTPM only supports SRTM PCR[0~16]

6.1.5 vTPM AK Certificate

vTPM Service follows [TPM2 KEY] specification to generate Attestation Key (AK). The detail setup is in "Identity Provisioning" section of [TPM2 KEY].

NOTE: OEM creates Initial AK (IAK) while owner creates Local AK (LAK).

6.2 vTPM Capabilities and Commands

6.2.1 vTPM Command

vTPM Service follows the [TCG PTP] specification, "Command Table" section.

6.2.2 vTPM Locality

vTPM only supports locality 0 (zero). Other localities (1~4) are unsupported.

6.2.3 vTPM Timeout

vTPM uses a different interface. As such, it does not follow [TCG PTP] specification, "**Interface Timeouts**" section, including TIMEOUT_A, TIMEOUT_B, TIMEOUT_C, TIMEOUT_D. The vTPM deriver in TD may wait longer.

6.3 vTPM Software Interface

6.3.1 vTPM Interface Type

L2 User VM and L1 vTPM Service use TPM CRB Private MMIO to transmit messages.

§

7 vTPM NV Storage Management

L1 vTPM Service only provides ephemeral NV Storage support inside of the TD. That means the TPM Instance NVS can be reused if vTPM Instance is not deleted and vTPM Service is not shutdown. After the VMM terminates the vTPM Service, the NV Storage does not exist. When VMM launches the L1 VMM and vTPM Service again, the vTPM Service need reprovision the vTPM instance. There is no persistent NV storage, because the TDX architecture does not have sealing capability.

The L1 vTPM Service will management the NV Storage in the TD for the vTPM instance. When a vTPM instance is created, the vTPM NVS instance is created. When a vTPM instance is destroyed, the vTPM NVS instance is destroyed.

An OSV may add extension to support persistent NV storage for the L1 vTPM Service. For example, using a NVS server to provide secure storage. That is out of scope of this document.

§

8 L1 vTPM Service Measurement

L1 VMM contains the L1 vTPM Service. See the following table for details.

Table 11: TD Measurement Registers for L1 vTPM Service

Typical Usage	Register	CC Event Log	Extended by	Content
L1 VMM	MRTD	NO	SEAMCALL [TDH.MR.EXTEND]	L1 Coconut-SVSM
L1 VMM Config	RTMR [0]	YES	TDCALL [TDG.MR.RTMR.EXTEND]	Launch Parameter (TD_HOB)
SEPARATOR	RTMR [0~3]	YES	TDCALL [TDG.MR.RTMR.EXTEND]	Four bytes 0 (zero)

§

9 vTPM Migration

If the TD is migratable, a vTPM Service and L2 User VM shall be bound to one TD with the same Migration Service TD to support TDX live migration in TDX 1.5.

§

10 vTPM Field Upgrade

A vTPM Service may support TPM field upgrade. In virtual TPM case, it means the vTPM Service is updated. The VMM needs to tear down the old TD and starts a new TD.

10.1 vTPM instance EK Cert update

vTPM instance EK includes the TD Quote.

If the vTPM Service restarts, the vTPM instance in the new vTPM Service will recreate a new EK Cert based upon the new TD Quote.

§

Appendix A Reference

A.1 Standards

[TCG VTPM] TCG Virtualized Trusted Platform Architecture Specification, https://trustedcomputinggroup.org/resource/virtualized-trusted-platform-architecture-specification/

[TPM2] TPM2 Library Specification, https://trustedcomputinggroup.org/resource/tpm-library-specification/

[TPM2 PP] TCG Protection Profile for PC Client Specification TPM2.0, https://trustedcomputinggroup.org/resource/pc-client-protection-profile-for-tpm-2-0/

[TPM2 EK] TCG EK Credential Profile for TPM2.0, https://trustedcomputinggroup.org/resource/tcg-ek-credential-profile-for-tpm-family-2-0/

[TPM2 PROVISION] TCG TPM2.0 Provisioning Guidance, https://trustedcomputinggroup.org/resource/tcg-tpm-v2-0-provisioning-guidance/

[TPM2 KEY] TPM2.0 Keys for Device Identity and Attestation, https://trustedcomputinggroup.org/resource/tpm-2-0-keys-for-device-identity-and-attestation/

[TCG PTP] TCG PC Client Platform TPM Profile Specification, https://trustedcomputinggroup.org/resource/pc-client-platform-tpm-profile-ptp-specification/

[TCG PFP] TCG PC Client Specific Platform Firmware Profile Specification, https://trustedcomputinggroup.org/resource/pc-client-specific-platform-firmware-profile-specification/

[TCG EFI] TCG EFI Protocol Specification, https://trustedcomputinggroup.org/resource/tcg-efi-protocol-specification/

[TCG ACPI] TCG ACPI Specification, https://trustedcomputinggroup.org/resource/tcg-acpi-specification/

[TCTI] TCG TSS 2.0 TPM Command Transmission Interface (TCTI) API Specification, https://trustedcomputinggroup.org/resource/tss-tcti-specification/

[TCG TAP] TCG Trusted Attestation Protocol (TAP) Information Model for TPM Families 1.2 and 2.0 and DICE Family 1.0, https://trustedcomputinggroup.org/resource/tcg-tap-information-model/

[TDVF] Intel TDX Virtual Firmware Design Guide, https://software.intel.com/content/www/us/en/develop/articles/intel-trust-domain-extensions.html

[GHCI] Guest Hypervisor Communication Interface Spec v1.5 (PDF), https://cdrdv2-public.intel.com/726792

A.2 Web Resources

[TPM2 ms-tpm-20-ref] Microsoft TPM2.0 Reference, https://github.com/microsoft/ms-tpm-20-ref

[TPM2 libtpms] Linux TPM2.0 Reference, https://github.com/stefanberger/libtpms

[TPM2 Software] TPM2.0 Software Community, https://tpm2-software.github.io/

[TPM2 Tutorials] TPM2.0 Software Tutorials, https://tpm2-software.github.io/tutorials/

[TPM2 Remote Attestation] Remote Attestation Best Known Methods, https://tpm2-software.github.io/tpm2-tss/getting-started/2019/12/18/Remote-Attestation.html

[TPM2 Remote Attestation tool] Remote Attestation with TPM2 Tools, https://tpm2-software.github.io/2020/06/12/Remote-Attestation-With-tpm2-tools.html

[Safeboot Remote Attestation] Safeboot Remote Attestation, https://safeboot.dev/attestation/

[Keylime] https://next.redhat.com/project/keylime/

[Coconut-SVSM] https://github.com/coconut-svsm

§