Intel® TDX Module Extension for Quoting

874303-001US (DRAFT)

0.1

February 2026

Ref. # 874303-001US (DRAFT)

Notices and Disclaimers

Intel Corporation ("Intel") provides these materials as-is, with no express or implied warranties.

All products, dates, and figures specified are preliminary, based on current expectations, and are subject to change without notice. Intel does not guarantee the availability of these interfaces in any future product. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described might contain design defects or errors known as errata, which might cause the product to deviate from published specifications. Current, characterized errata are available on request.

Intel technologies might require enabled hardware, software, or service activation. Some results have been estimated or simulated. Your costs and results might vary.

No product or component can be absolutely secure.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted that includes the subject matter disclosed herein.

No license (express, implied, by estoppel, or otherwise) to any intellectual-property rights is granted by this document.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

Copies of documents that have an order number and are referenced in this document or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting http://www.intel.com/design/literature.htm.

Table of Contents

1. About this Document

1.1. Scope of this Document

The purpose of this document is to demonstrate an alternative design for obtaining Intel® Trust Domain Extensions (Intel® TDX)Attestation Quotes by using a new Intel® TDX Module Quoting Feature.

In the previous design of the Intel TDX Module, these steps were made by a specific type of **trust domain** (**TD)** called the **Quoting TD**. The alternative design presented in this document eliminates the need for Quoting TD and instead moves the abovementioned functionality into an Intel TDX Module Quoting Extension called **Quoting Service**.

The intention of this document is not to demonstrate the internal details of the Quoting Service architecture, but to provide enough information on how the functionality provided by this Intel TDX Module Extension can be used by an Operating System (OS) or Virtual Machine Manager (VMM).

This document is part of the **Intel TDX Module Architecture Specification Set**, which includes the following documents:

Table 1.1: Intel TDX Module Architecture Specification Set

Document Name	Reference	Description
Intel TDX Module Base Architecture Specification	[Intel TDX Module Base Spec]	Base Intel TDX Module architecture overview and specification, covering key management, TD lifecycle management, memory management, virtualization, measurement and attestation, service TDs, debug aspects etc.
Intel TDX Module TD Migration Architecture Specification	[TD Migration Spec]	Architecture overview and specification for TD migration
Intel TDX Module TD Partitioning Architecture Specification	[TD Partitioning Spec]	Architecture overview and specification for TD Partitioning
→ **Intel TDX Module Extension for Quoting**	[Intel TDX Module Quoting Extension]	Architecture overview and specification for TDX Module Quoting Extension
Intel TDX Module TDX Connect Specification	[TDX Connect Spec]	Architecture overview and specification for Intel TDX Connect
Intel TDX Module ABI Reference Specification	[Intel TDX Module ABI Spec]	Detailed Intel TDX Module Application Binary Interface (ABI) reference specification, covering the entire Intel TDX Module architecture
Intel TDX Module TDX Connect ABI Reference Specification	[Intel TDX Connect ABI Spec]	Detailed Intel TDX Module Application Binary Interface (ABI) reference specification, covering the TDX connect architecture
Intel TDX Module ABI Reference Tables	[Intel TDX Module ABI Tables]	A set of files detailing Intel TDX Module Application Binary Interface (ABI)
Intel TDX Module ABI Incompatibilities	[Intel TDX Module ABI Incompatibilities]	Description of the incompatibilities between Intel TDX 1.0 and Intel TDX 1.4/1.5 that may impact the host VMM and/or guest TDs

This document is a work in progress and is subject to change based on customer feedback and internal analysis. This document does not imply any product commitment from Intel to anything in terms of features and/or behaviors.

Note: The contents of this document are accurate to the best of Intel's knowledge as of the date of publication, though Intel does not represent that such information will remain as described indefinitely in light of future research and design implementations. Intel does not commit to updating this document in real time when such changes occur.

This section describes the notation used in this document.

1.2. Notation

This section describes the notation used in this document.

1.2.1. Requirement and Definition Commitment Levels

When specifying requirements or definitions, the level of commitment is specified following the convention of RFC 2119: Key words for use in RFCs to indicate Requirement Levels, as described in the following table:

Table 1.2: Requirement and Definition Commitment Levels

Keyword	Description
Must	"**Must**", "**Required**" or "**Shall**" means that the definition is an absolute requirement of the specification.
Must Not	"**Must Not**" or "**Shall Not**" means that the definition is an absolute prohibition of the specification.
Should	"**Should**", or the adjective "**Recommended**", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
Should Not	"**Should Not**", or the phrase "**Not Recommended**" means that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood, and the case must be carefully weighed before implementing any behavior described with this label.
May	"**May**", or the adjective "**Optional**", means that an item is discretionary. An implementation may choose to include the item, while another may omit the same item, because of various reasons.

1.3. Glossary

If required, define glossary items in the table below

Table 1.3: Intel TDX Glossary

Acronym	Full Name	New for TDX	Description

1.4. References

See the [TDX Module Base Spec].

2. Intel TDX Module Extension for Quoting

2.1. Overview

To create a remotely verifiable attestation, the TDREPORT_STRUCT should be converted into a Quote signed by a certified Quote signing key. The following models are supported for creating a Quote:

- Platforms that support Intel® Software Guard Extensions (Intel® SGX)can support Quoting Enclaves producing either Intel TDX or Intel SGX Quotes. A TD Quoting Enclave, when available, will produce legacy quotes for Intel TDX.

On platforms that support an enabled Security Engine (S3M):

- The security engine can be used to create an attestation x509 certificate.
  A Quoting TD can create legacy-style Quotes or x509 certificates. The Quoting TD itself is certified by a Security Engine-based Attestation.

This specification introduces a new Intel TDX Module Feature for producing Intel TDX Attestation Quotes, implemented using an NRX (Quoting Service), which is an alternative to the Quoting TD design outlined in the [Intel TDX Module Base Spec]. The components involved in Intel TDX Attestation using Quoting Service are shown in Figure 2.1.

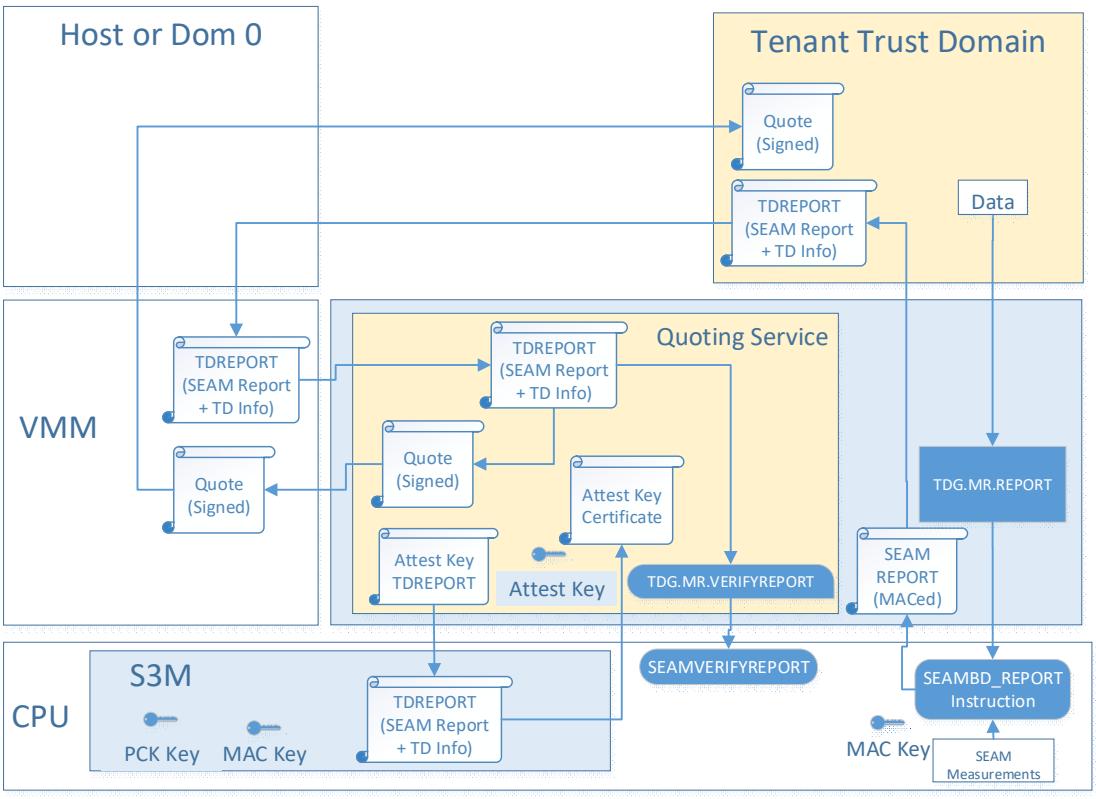

Figure 2.1: S3M Based Attestation using Intel TDX Module Feature for producing Intel TDX Attestation Quotes

2.2. Intel TDX Module Extensions overview

For overall information on Intel TDX Module Extensions please consult [Intel TDX Module Base Spec] section “TDX Module Extensions Overview”.

2.3. Initialization of Intel TDX Module Quoting Extension

The initialization of Intel TDX Module Quoting Extension happens in the same way as any other Intel TDX Module Extension and is described in detail in [Intel TDX Module Base Spec] section “TDX Module Extensions Initialization”.

Prior to calling the TDH.SYS.CONFIG/TDH.SYS.UPDATE host-side (SEAMCALL) interface function with FEATURES_ENABLE0/1.QUOTE (bit 50) set to 1 to enable Intel TDX Module Quoting Extension, a VMM can optionally discover and configure the number of threads in the virtual threads pool and the number of sessions that are going to be used by the Intel TDX Module Quoting Extension by using the following fields (see section 3.2.1 for detailed definitions):

- QUOTE_MAX_SESSIONS
- QUOTE_NUM_SESSIONS
- QUOTE_MAX_THREADS
- QUOTE_NUM_THREADS

For the explanation on how these values affect the concurrency, interruption, and overall session management in the Intel TDX Module, please consult [Intel TDX Module Base Spec] sections “Virtual Thread Pools” and “Sessions”.

Note: The value of QUOTE_NUM_THREADS is going to determine how the maximum number of TDH.QUOTE.* host-side (SEAMCALL) interface functions can be called in parallel by the VMM. If the VMM invokes QUOTE_NUM_THREADS + 1 TDH.QUOTE.* host-side (SEAMCALL) interface functions, the last one will get TDX_OPERAND_BUSY exit code.

Note: A quoting session consists of a single invocation of the TDH.QUOTE.GET host-side (SEAMCALL) interface function. As a result, once the TDH.QUOTE.GET host-side (SEAMCALL) interface function completes, no session state is stored by the Intel TDX Module.

2.4. Initial Certification of Attestation Keys

Before being able to sign Intel TDX Attestation Quotes, the Intel TDX Module must first certify the attestation keys it supports.

To enable each of the supported attestation keys to be used for producing a runtime Intel TDX Quote, the VMM must call a new **TDH.QUOTE.INIT** host-side (SEAMCALL) interface function (see section 3.3.1 for details).

A successful execution of TDH.QUOTE.INIT is indicated by the TDX_SUCCESS status code, and the bitmap of successfully enabled QUOTE_IDs is returned back to the VMM (see section 3.3.1 for details). Additionally, the VMM can use an existing TDH.SYS.RD host-side (SEAMCALL) interface function to read the QUOTE_ENABLED_QUOTE_IDS bitmask (see section 3.2.1 for definition of this field) to learn what attestation keys have been successfully enabled.

Note: If at least one of the supported attestation keys has been enabled as result of execution of TDH.QUOTE.INIT, the VMM always gets back the TDX_SUCCESS status code. Thus, if the VMM requires a specific attestation key to be enabled, it must always read either the TDH.QUOTE.INIT returned bitmap of the enabled QUOTE_IDs or explicitly read the QUOTE_ENABLED_QUOTE_IDS bitmask after TDH.QUOTE.INIT completes to confirm that a required key has been successfully enabled.

After the TDH.QUOTE.INIT has finished successfully, a VMM can use an existing TDH.SYS.RD host-side (SEAMCALL) interface function to query two additional parameters:

- QUOTE_MAX_SIZE (see section 3.2.1 for details) that specifies the maximum size of the buffer that must be allocated to contain the Intel TDX Quote received from Intel TDX Module. Prior to calling TDH.QUOTE.INIT, the QUOTE_MAX_SIZE returns 0.
- QUOTE_ID (see section 3.2.1 for details) that specifies the value which is going to be used for a TD-scope field ENABLED_QUOTE_IDS (see section 3.2.2 for details) at the time of the TD’s creation. In turn, a key selected from the TD-scope ENABLED_QUOTE_IDS field will be used as a default attestation key when calling TDH.QUOTE.GET for this TD without explicitly specifying an attestation key via the QUOTE_ID input parameter. The initial value for QUOTE_ID field is all-zeros, meaning no default exists.

Note: After a TD preserving update to a different version of a Intel TDX Module, a call to TDH.QUOTE.INIT may change the QUOTE_ENABLED_QUOTE_IDS, QUOTE_MAX_SIZE, or QUOTE_ID values, hence these values must be read again after a successful invocation of TDH.QUOTE.INIT if a VMM relies on them for correct operation.

A VMM can also specify its own preference for a default attestation key by writing a selected value into QUOTE_ID using TDH.SYS.WR after TDH.QUOTE.INIT has completed successfully. The written value must be a single bit which is present in the QUOTE_ENABLED_QUOTE_IDS bitmap. Every subsequent call to TDH.QUOTE.INIT will override the VMM set value and, if required, VMM must set it again.

The global values of QUOTE_ID and QUOTE_MAX_SIZE are used as default values for the TD-scope fields (see section 3.2.2 for details): ENABLED_QUOTE_IDS and QUOTE_MAX_SIZE. These TD scope fields are captured during TD creation and remain unchanged through TD’s lifetime. The checks that the Intel TDX Module is doing on these fields during Intel

TDX Migration are captured in section 2.4.2. VMM can optionally overwrite TD's scope ENABLED_QUOTE_IDS field prior to TD's finalization (see section 3.2.2 for details).

The overall flow from the VMM perspective is shown in Figure 2.2.

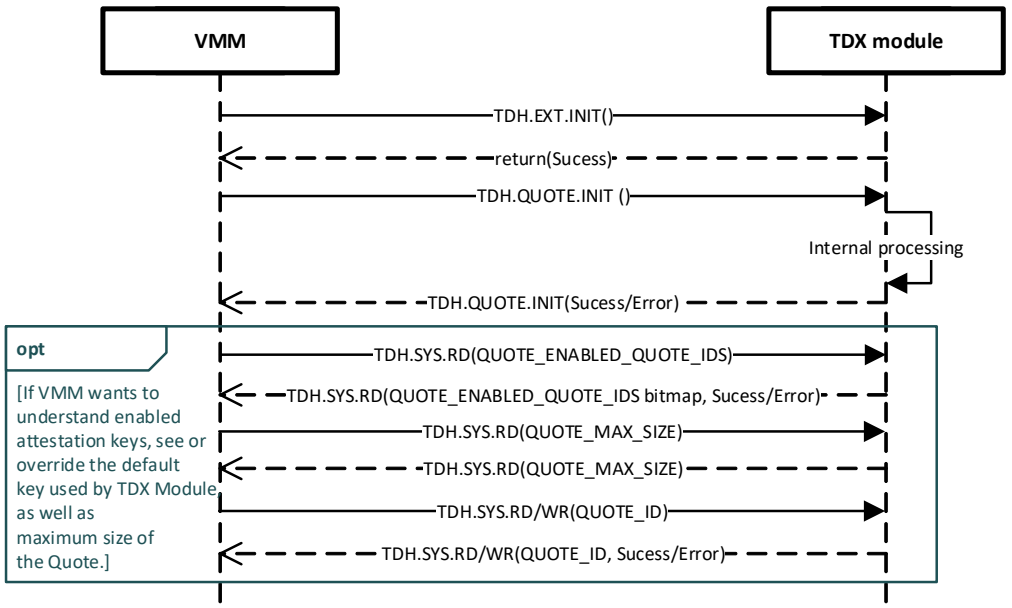

Figure 2.2 Quoting Service Initialization and Certification of Attestation Keys: VMM view

2.4.1. Certification of Attestation Keys: Internal details

For informational purpose, Figure 2.3 shows the same flow as in Figure 2.2, but also shows the internal interaction between the Intel TDX Module, Quoting Service and S3M Intel TDX Attestation mailbox.

2.4.2. Impacts on Intel TDX Migration

Section 3.2.2 defines two new TD-scope fields:

1. ENABLED_QUOTE_IDS
2. QUOTE_MAX_SIZE

These fields are captured during TD's creation and remain unchanged throughout TD's lifetime (including when TD is migrated). To ensure that a TD can correctly function on a destination platform, the Intel TDX Module does the following checks during Intel TDX Migration:

1. If the source TD has been configured with the Quoting Service feature enabled, Intel TDX Module checks that destination platform also has Quoting Service feature enabled. This check ensures that when running on a destination platform, TD can continue using Quoting Service.
2. All bits in the source TD's ENABLED_QUOTE_IDS must be present in the global in the set of QUOTE_ENABLED_QUOTE_IDS of the destination platform. This check ensures that when running on a destination platform, a TD is able to obtain Quotes using any of TD's enabled attestation keys ENABLED_QUOTE_IDS.
3. The size of source TD's QUOTE_MAX_SIZE is smaller or equal to the global QUOTE_MAX_SIZE field of the destination platform. This check ensures that when running on a destination platform, a buffer size that has been allocated by a TD for getting Intel TDX Quotes fits into the destination platform buffer.

Note: The above migration checks are going to be applied by Intel TDX Module even in cases when the VMM has enabled Quoting Service to be used by TDs in addition to the legacy Intel SGX-based Intel TDX Attestation, but a TD only supports and uses legacy Intel SGX-based Intel TDX Attestation. The reason for this is that Intel TDX Module is unaware of the software that runs inside a TD and this software might change during the lifetime of a TD.

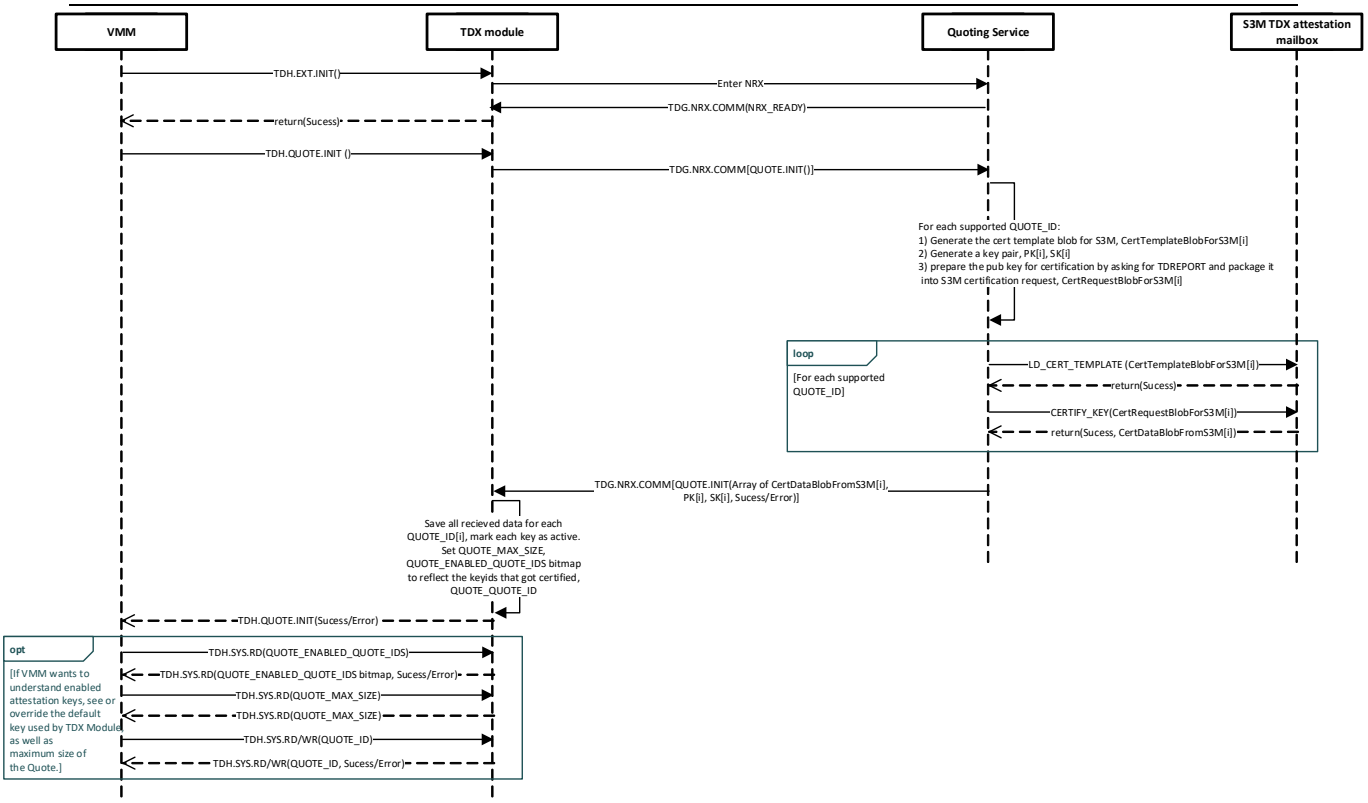

Figure 2.3 Quoting Service Initialization and Certification of Attestation Keys: internal details

2.5. Runtime processing

In order to support Guest TD's TDVMCALL queries for Intel TDX Quotes, a new Intel TDX Quote, **TDH.QUOTE.GET** host-side (SEAMCALL) interface function (see section 3.3.2 for details) is introduced. TDH.QUOTE.GET takes two primary inputs:

- A buffer containing the TDREPORT_STRUCT received from the TD guest that is being asked to be converted into an Intel TDX Quote.
- A QUOTE_ID identifier to explicitly select the attestation key which should be used for signing the returned Intel TDX Quote. If QUOTE_ID is all-ones, then Intel TDX Module will use the value of a TD-scope ENABLED_QUOTE_IDS (see section 3.2.2 for definition) to see what attestation keys have been enabled for this TD and use its internal ranking to determine the correct attestation key from this set (if VMM has set more than one bit in the bitmask).

The overall flow is outlined in Figure 2.4.

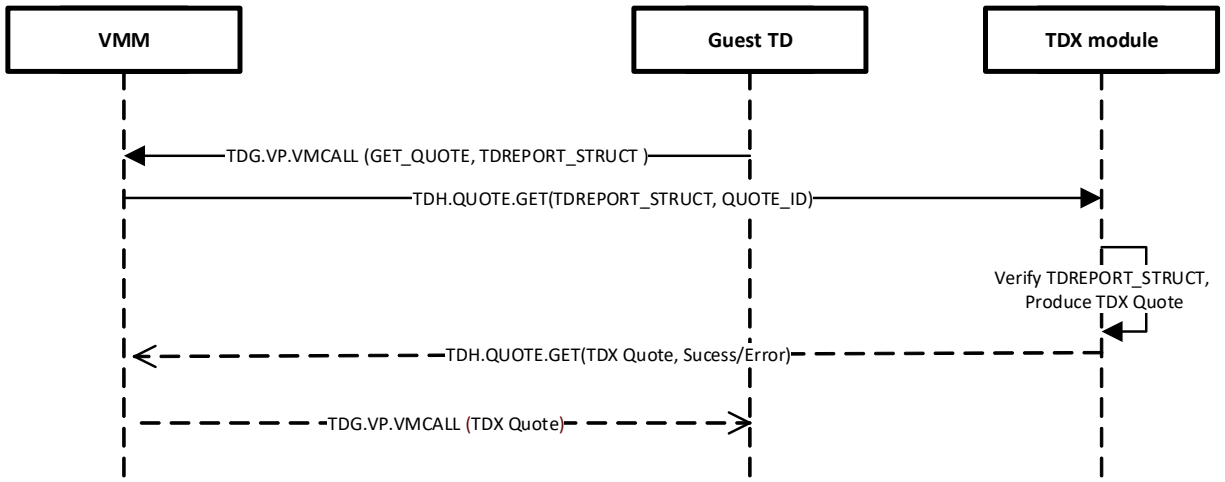

Figure 2.4 Runtime production of Intel TDX Quotes: VMM view

2.6. Update of Intel TDX Module Quoting Extension

In order to update any Intel TDX Module Extension (including Quoting Service), an Intel TDX Module TD preserving update is required. The overall flow is depicted in Figure 2.5 and more details can be found in [Intel TDX Module Base Spec] section "TDX Module Extensions Aspects of TD-Preserving Update".

2.6.1. Attestation keys management on SVN changes during a TD preserving update

Intel TDX Module will keep the SVN versioning of the attestation keys and certification data and perform checks against the currently running SVN on each TDH.QUOTE.GET host-side (SEAMCALL) interface function. If an increase of SVN is detected by the Intel TDX Module, and the Intel TDX Quote has been successfully generated, then the Intel TDX Module will return a special status code, TDX_QUOTE_SUCCESS_ATT_KEY_OUTDATED, together with the Intel TDX Quote, indicating that the attestation key material is not up-to-date. This flow is also shown in Figure 2.5.

Upon receiving this status code, VMM can decide to perform the steps to ask Intel TDX Module to reinitialize its attestation keys via the TDH.QUOTE.INIT host-side (SEAMCALL) interface function. After this function completes successfully, all the following invocations of TDH.QUOTE.GET host-side (SEAMCALL) interface function will start to use the updated attestation keys.

Note: If VMM has configured QUOTE_NUM_THREADS = 1, all parallel invocations of TDH.QUOTE.GET until TDH.QUOTE.INIT is completed will receive TDX_OPERAND_BUSY exit code.

Alternatively, the VMM can proactively update attestation keys right after the TD Preserving update (that increases the SVN) has completed successfully and the Quoting Feature has been initialized but before the first TDH.QUOTE.GET host-side (SEAMCALL) interface function is invoked.

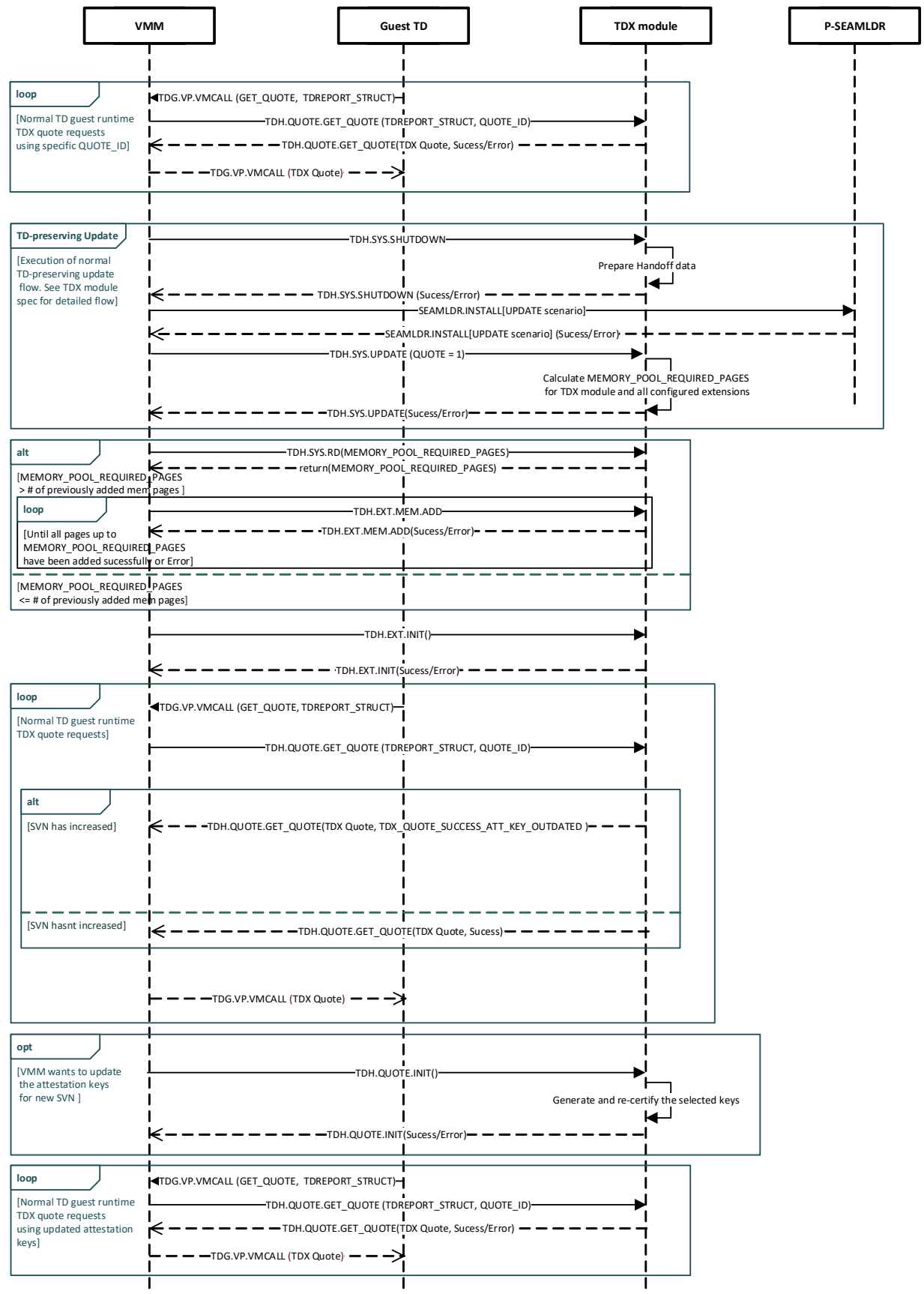

Figure 2.5 Update of Quoting Service or Intel TDX Module via TD Preserving update: VMM view

2.7. Fatal Error Handling in Intel TDX Module Quoting Extension by VMM

If Intel TDX Module Quoting Extension (Quoting Service) experiences a fatal error, the Intel TDX Module returns an TDX_EXT_FATAL_ERROR back to the VMM as a result of the ongoing host-side (SEAMCALL) interface function. The way to restart any Intel TDX Module Extension is to perform a TD preserving update or reload the Intel TDX Module. The TD preserving update flow to recover from Quoting Service fatal error is shown in Figure 2.6.

More details on how the Intel TDX Module handles fatal errors for its extensions can be found in [Intel TDX Module Base Spec] section "TDX Module Extensions Fatal Error Handling".

Note: The flow in Figure 2.6 assumes the same version of Intel TDX module is loaded and hence there is no SVN change in the Intel TDX module. If a newer version of Intel TDX Module is loaded with a higher SVN, then VMM will get a special return code TDX_QUOTE_SUCCESS_ATT_KEY_OUTDATED similarly as in Figure 2.5.

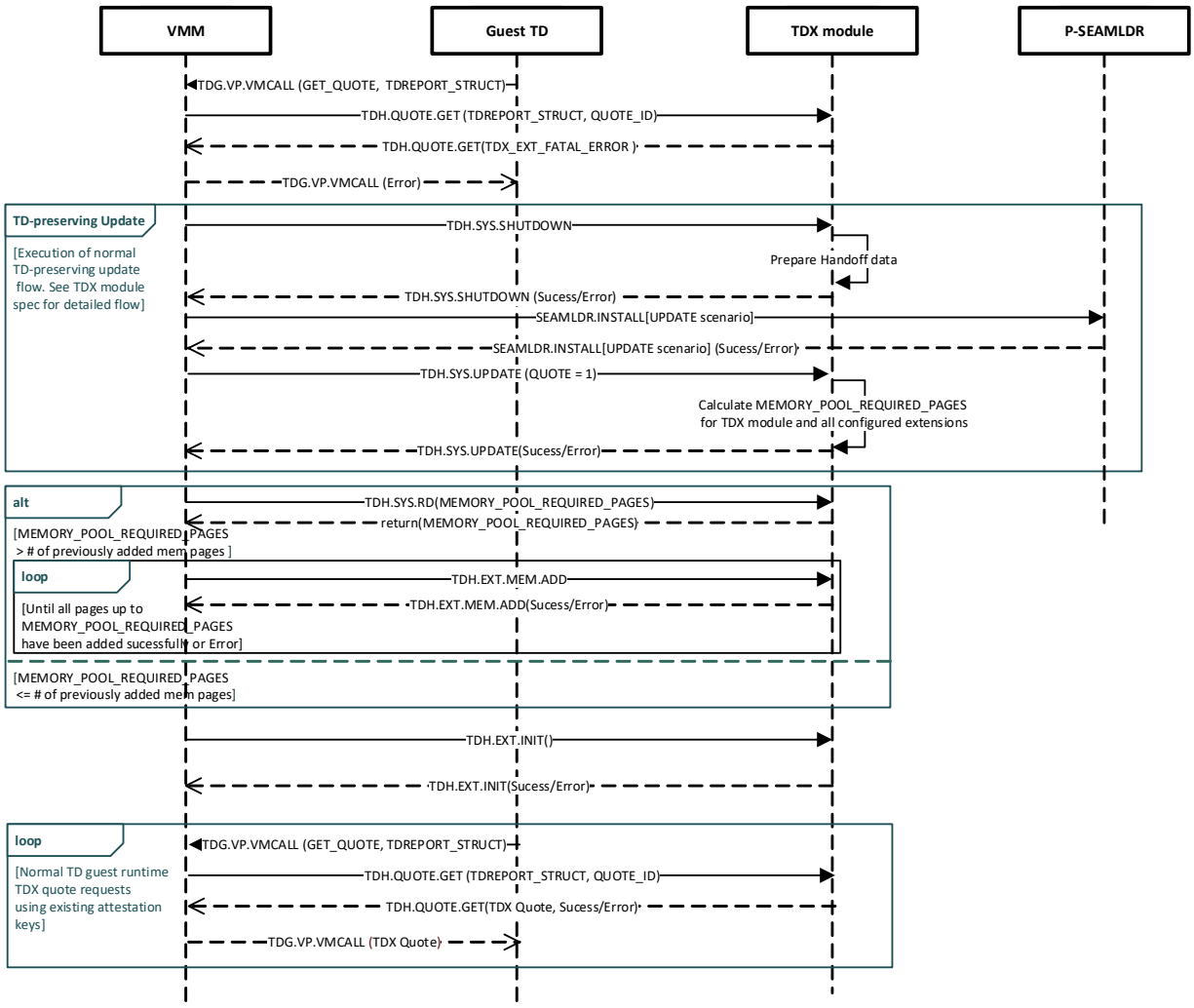

Figure 2.6 Recovering of Quoting Extension in case of the Fatal Error: VMM view

3. Quoting related Intel TDX Module Specification Changes

Note: This section will be merged in the future into [Intel TDX Module ABI Spec].

3.1. Intel TDX Module Enumeration & Configuration

A new bit will be added to TDX_FEATURES0.QUOTE (bit 50), to enumerate Intel TDX Module support for the Quoting Service Feature.

Table 3.1: TDX_FEATURES0 Definition

Bit(s)	Name	Description
…	…	…
50	QUOTE	Intel TDX Module supports producing Intel TDX Attestation Quotes
…	…	…

At runtime VMM can query (using TDH.SYS.RD*) whether the Quoting Service Feature has been enabled or not using a new TDX_FEATURES_ENABLED.QUOTE (bit 50). A TD can similarly query this status using TDG.SYS.RD*.

Note: The TDX_FEATURES_ENABLED.QUOTE (bit 50) gets set only after the Quoting Service has been initialized by successful execution of TDH.QUOTE.INIT.

3.2. Data Types

The following data types will be merged in the future to the “global fields” and “TDR/TDCS fields” spreadsheets respectively.

3.2.1. New: Global metadata

Name	Access	Size in Bytes	Type	Description
QUOTE_ENABLED_QUOTE_IDS	RO	8	64 bit bitmap	The VMM can read this field in order to get a bitmap of QUOTE_IDs that reflects attestation keys that have been enabled and available for obtaining a TDX Attestation Quote via TDH.QUOTE.GET. This value may change every time VMM calls TDH.QUOTE.INIT. The initial value for this field is all-zeros, meaning no attestation keys have been enabled. The values of currently defined QUOTE_IDs are specified in Table 3.8
QUOTE_ID	RW	8	64-bit bitmap with a single bit set	The VMM can read this field in order to obtain the default key which is going to be used for Intel TDX Attestation when calling TDH.QUOTE.GET without explicitly specifying an attestation key via QUOTE_ID input parameter. This value may change every time the VMM calls TDH.QUOTE.INIT. The initial value for this field is all-zeros, meaning no default exists. A VMM can specify its own preference for default attestation key by writing a selected value into QUOTE_ID using TDH.SYS.WR after TDH.QUOTE.INIT has completed successfully. The written value must be a single bit which is present in the QUOTE_ENABLED_QUOTE_IDS bitmap. Every subsequent call to TDH.QUOTE.INIT will override the VMM set value and, if required, VMM must set it again.
QUOTE_MAX_SIZE	RO	4	Unsigned integer	The VMM can read this field to get the maximum size of the buffer that must be allocated to contain the Intel TDX Quote received from Intel TDX Module.

Name	Access	Size in Bytes	Type	Description
				This value can change every time the VMM calls TDH.QUOTE.INIT. Prior to calling TDH.QUOTE.INIT, the QUOTE_MAX_SIZE returns 0.
QUOTE_MAX_SESSIONS	RO	2	Unsigned integer	The VMM can read this field in order to determine the maximum number of sessions that Quoting feature supports
QUOTE_MAX_THREADS	RO	2	Unsigned integer	The VMM can read this field in order to determine the maximum number of concurrent threads that Quoting feature supports
QUOTE_NUM_SESSIONS	RW	2	Unsigned integer	The VMM can read this field in order to determine the currently configured number of sessions that Quoting feature is using. The VMM can also write the selected value into this field using TDH.SYS.WR. The value must be less or equal to QUOTE_MAX_SESSIONS
QUOTE_NUM_THREADS	RW	2	Unsigned integer	The VMM can read this field in order to determine the currently configured number of concurrent threads that Quoting feature is using. The VMM can also write the selected value into this field using TDH.SYS.WR. The value must be less or equal to QUOTE_MAX_THREADS

3.2.2. New: TD-Scope Metadata

Name	Access	Size in BYTES	Type	Description
ENABLED_QUOTE_IDS	RO	8	64 bit bitmap	A TD can read this field in order to see the attestation keys that are enabled for Intel TDX Attestation. The default value of this field is captured during TD's creation time from the global field QUOTE_ID. VMM can override this default value prior to finalizing a TD. Any bits that VMM sets in ENABLED_QUOTE_IDS must be a subset of QUOTE_ENABLED_QUOTE_IDS. The value of ENABLED_QUOTE_IDS does not change during TD's lifetime, including on Intel TDX Migration. Intel TDX module will ensure that upon the TD's migration, the destination platform is capable of correctly supporting this field value. See checks in section 2.4.2 for details. A TD only should read this field if the Intel TDX Module Quoting Service Feature discovery reports as enabled (TDX_FEATURES_ENABLED.QUOTE (bit 50) equals 1).
QUOTE_MAX_SIZE	RO	4	Unsigned integer	A TD can read this field in order to see the maximum size of the Intel TDX Quote in bytes. The value of this field is captured during the TD's creation time from global field QUOTE_MAX_SIZE. This value does not change during the TD's lifetime. Intel TDX module will ensure that upon the TD's migration, the destination platform is capable of correctly supporting this field value. See checks in section 2.4.2 for details. A TD only should read this field if the Intel TDX Module Quoting Service Feature discovery reports as enabled (TDX_FEATURES_ENABLED.QUOTE (bit 50) equals 1).

3.2.3. New: Structures

3.2.3.1. HPA_LINKED_LIST Type

The HPA_LINKED_LIST is a linked list of 4KB pages that contains a list of HPAs. It is used, for example, as an input to TDH.QUOTE.INIT and as input and output to/from TDH.QUOTE.GET. Each page in linked list is 4KB aligned. Each linked list page can contain up to 511 entries of type HPA_LIST Entry (see [Intel TDX Module ABI Spec] section 4.7.6, Table 4.51 for definition of HPA_LIST Entry). The last 64 bits of each page may contain another HPA pointing to the next HPA_LINKED_LIST page or NULL_PA (-1) if there is no next HPA_LINKED_LIST page.

3.3. Host-Side (SEAMCALL) Functions

3.3.1. New: TDH.QUOTE.INIT

Enable the Quoting Feature for all available attestation keys.

3.3.1.1. Input Operands

Table 3.2 TDH.QUOTE.INIT Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Number	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function
		23:16	Version Number	Selects the SEAMCALL interface function version
		63:24	Reserved	Must be 0

3.3.1.2. Output operands

Table 3.3 TDH.QUOTE.INIT Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
RCX	If RAX returns TDX_SUCCESS, the bitmap of successfully enabled attestation keys in the same format as QUOTE_ENABLED_QUOTE_IDS Unmodified otherwise.
Other	Unmodified

3.3.1.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.3.1.3.1. Overview

TDH.QUOTE.INIT allows VMM to enable the Intel TDX Module Quoting Feature for all supported attestation keys.

Only one call to TDH.QUOTE.INIT is allowed at any given time, any parallel invocation of TDH.QUOTE.INIT before the completion of the existing running call will return a busy indication (TDX_OPERAND_BUSY).

Note: VMMs must not launch TDs that are expected to use Intel TDX Module Quoting Feature before successful completion of TDH.QUOTE.INIT since such TDs won't see the Intel TDX Module Quoting Feature as enabled.

The VMM is not required to call TDH.QUOTE.INIT again after a TD preserving update (assuming TDH.QUOTE.INIT has completed successfully prior to TD preserving update) unless it wants the Quoting Service to re-generate and re-certify a new set of attestation keys.

3.3.1.3.2. Enumeration

Support for TDH.QUOTE.INIT is enumerated by FEATURES_ENABLE0.QUOTE (bit 50), readable by TDH.SYS.RD*.

3.3.1.3.3. Interruptibility

TDH.QUOTE.INIT is interruptible. If a pending interrupt is detected during operation, TDH.QUOTE.INIT returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.

For the general explanation on how interruption and resumption is handled for all Quoting Service functions please consult [Intel TDX Module Base Spec] section “Request Interruption and Resumption”.

Rest of details are TBD

3.3.1.4. *Operands Information*

To understand the table and text below, please refer to the [Intel TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.4: TDH.QUOTE.INIT Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	QUOTE_STATE	N/A	RW	Hidden	N/A	Exclusive	None	None
Implicit	N/A	N/A	S3M	N/A	RW	Hidden	N/A	Exclusive	None	None

3.3.1.5. *Completion Status Codes*

Table 3.5 TDH.QUOTE.INIT Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description	DETAILS_L2 Bits 31:0
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.	Operand ID. Detailed reason for getting BUSY indicator: • S3M • QUOTE_STATE
TDX_OPERAND_INVALID		
TDX_SUCCESS	Operation is successful	
TDX_LIMIT_CPUID_MAXVAL_SET	IA32_MISC_ENABLES MSR bit 22 (Limit CPUID Maxval) is set.	
TDX_INCONSISTENT_MSR	IA32_TSC_ADJUST MSR value is different than the value captured during the TDH.SYS.INIT interface function.	
TDX_TSC_ROLLBACK	Time Stamp Counter value is lower than on last TD exit.	

Completion Status Code	Description	DETAILS_L2 Bits 31:0
TDX_INTERRUPTED_RESUMABLE	Interrupt is pending, the operation can be resumed	
TDX_EXT_NOT_READY		
TDX_EXT_NOT_INITIALIZED		
TDX_EXT_FATAL_ERROR		
TDX_QUOTE_GEN_KEY_ERROR	Failed to generate any supported attestation key	
TDX_QUOTE_CERT_KEY_ERROR	Failed to certify any supported attestation key	
TDX_QUOTE_CERT_CHAIN_ERROR	The attestation keys could not be certified due to a problem with the platform certificate chain	

3.3.2. New: TDH.QUOTE.GET Leaf

Request for an Intel TDX Quote.

3.3.2.1. *Input Operands*

Table 3.6 TDH.QUOTE.GET Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Number	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function
		23:16	Version Number	Selects the SEAMCALL interface function version
		63:24	Reserved	Must be 0
RCX		The physical address of the target TD's TDR page (HKID bits must be 0) If this is a non-TD scope invocation of TDH.QUOTE.GET (see TDH.MIG.SETUP), then it should be set to NULL_PA (-1).		
RDX	Control parameters	Selects the unique request ID, attestation key ID and operation		
		Bits	**Field**	**Description**
		11:0	REQUEST_ID	A unique ID identifying this TDH.QUOTE.GET call in case it gets interrupted and needs to be restarted
		31:12	Reserved	Must be 0
		47:32	QUOTE_ID	The type of the QUOTE_ID as defined in Table 3.8 Setting this to all-ones means that a key currently specified in TD's QUOTE_ID will be used.

Operand	Name	Description		
				For non-TD scope invocations of TDH.QUOTE.GET, this field must be set to all-ones indicating that a key currently specified in a global field QUOTE_ID will be used.
		63:48	Reserved	Must be 0
R8	TDREPORT_STRUCT	Input buffer containing TDREPORT_STRUCT		
		Bits	**Field**	**Description**
		0:0	STRUCT_TYPE	Determines the content of HPA_ADDRESS: Simple HPA address of a single page: 0 HPA_LINKED_LIST format, see section 3.2.3.1
		11:1	Reserved	Must be 0
		51:12	HPA_ADDRESS	Content depends on the value of STRUCT_TYPE
		63:52	Reserved	Must be 0
R9	TDREPORT_STRUCT_SIZE	The size of the TDREPORT_STRUCT in bytes		
R10	TDX_QUOTE_BUF_HPA_LIST	Reference to a linked list of memory pages containing the the output Intel TDX Quote, following HPA_LINKED_LIST format, see section 3.2.3.1		
R11	TDX_QUOTE_BUF_SIZE	The size of the VMM allocated buffer for the output Intel TDX Quote in bytes. Must be bigger or equal to the value read from the QUOTE_MAX_SIZE field.		

3.3.2.2. *Output operands*

Table 3.7 TDH.QUOTE.GET Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
RCX	If RAX returns TDX_SUCCESS or TDX_QUOTE_SUCCESS_ATT_KEY_OUTDATED, the size of the output Intel TDX Quote in bytes. Unmodified otherwise.
Other	Unmodified

3.3.2.3. *Leaf Function Description*

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.3.2.4. *Overview*

TDH.QUOTE.GET allows to convert a given TDREPORT to an Intel TDX Quote for either a TD identified by TD's TDR page provided in RCX or for a TDREPORT not associated with any TD (non-TD scope invocation of TDH.QUOTE.GET).

RDX[15:0] must contain a unique REQUEST_ID and RDX[31:16] (QUOTE_ID) must be either set to a desired supported QUOTE_ID (see Table 3.8), or alternatively set to all-ones, indicating that a default attestation key should be used.

In case QUOTE_ID is set to all-ones in a TD-scope invocation of TDH.QUOTE.GET, the default attestation key is taken from a TD-scope field QUOTE_ID (see section 3.2.2 for definition). If the VMM sets QUOTE_ID to a value not present in QUOTE_ENABLED_QUOTE_IDS (see section 3.2.1 for definition), TDH.QUOTE.GET returns with a TDX_QUOTE_

NON_ENABLED_QUOTE_ID status in RAX. Similarly, if the VMM sets QUOTE_ID to a value not supported by Quoting Service, TDH.QUOTE.GET returns with a TDX_QUOTE_UNSUPPORTED_QUOTE_ID status in RAX.

For non-TD scope invocations of TDH.QUOTE.GET, QUOTE_ID must be set to all-ones indicating that a key currently specified in a global field QUOTE_ID should be used.

The TDREPORT_STRUCT input buffer should contain the TDREPORT_STRUCT that is requested to be converted into an Intel TDX Quote. TDREPORT_STRUCT_SIZE should reflect the size of TDREPORT_STRUCT in bytes. If Quoting Service fails to successfully verify the provided TDREPORT_STRUCT, TDH.QUOTE.GET returns with a TDX_QUOTE_TDREPORT_ERROR status in RAX.

A successful invocation of TDH.QUOTE.GET returns with either TDX_SUCCESS or TDX_QUOTE_SUCCESS_ATT_KEY_OUTDATED status in RAX and the resulting Intel TDX Quote is provided in R10. For details when TDX_QUOTE_SUCCESS_ATT_KEY_OUTDATED is returned please see section 2.6.1).

Table 3.8 QUOTE_ID Values

Bit position	Name	Description
0	Reserved	Reserved for deprecated format.
1	TDX_QUOTE_ID_LEGACY_TDQE_ECDSA_256	The quote is a legacy Quote format. The Intel TDX Quoting Enclave signs the quote with ECDSA-256 using a signing key rooted in a hardware root of trust.
2	TDX_QUOTE_ID_EAT_CWT_TDXMQS_DICE_ECDSA_384	The quote is an IETF Entity Attestation Token (EAT) encoded as a CBOR Web Token (CWT). The Intel TDX Module Quoting Service signs the quote with ECDSA-384 using a signing key derived from a hardware root of trust following the Trusted Computing Group (TCG) Device Identifier Composition Engine (DICE) architecture.
3	TDX_QUOTE_ID_EAT_CWT_TDXMQS_DICE_MLDSA_87	The quote is an IETF Entity Attestation Token (EAT) encoded as a CBOR Web Token (CWT). The Intel TDX Module Quoting Service signs the quote with MLDSA 87 using a signing key derived from a hardware root of trust following the Trusted Computing Group (TCG) Device Identifier Composition Engine (DICE) architecture.
4	TDX_QUOTE_ID_EAT_CWT_SE_DICE_ECDSA_384	The quote is an IETF Entity Attestation Token (EAT) encoded as a CBOR Web Token (CWT). The security engine hardware signs the quote with ECDSA 384 using a signing key derived from a hardware root of trust following the Trusted Computing Group (TCG) Device Identifier Composition Engine (DICE) architecture.
5	TDX_QUOTE_ID_EAT_CWT_SE_DICE_MLDSA_87	The quote is an IETF Entity Attestation Token (EAT) encoded as a CBOR Web Token (CWT). The security engine hardware signs the quote with MLDSA 87 using a signing key derived from a hardware root of trust following the Trusted Computing Group (TCG) Device Identifier Composition Engine (DICE) architecture.
6:63	Reserved	Reserved for future Quote formats and crypto algorithms

3.3.2.4.1. Enumeration

Support of TDH.QUOTE.GET is enumerated by FEATURES_ENABLE0.QUOTE (bit 50), readable by TDH.SYS.RD*.

3.3.2.4.2. Interruptibility

TDH.QUOTE.GET is interruptible. If a pending interrupt is detected during operation, TDH.QUOTE.GET returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.

For the general explanation on how interruption and resumption is handled for all Quoting Service functions please consult [Intel TDX Module Base Spec] section "Request Interruption and Resumption".

Rest of details are TBD

3.3.2.5. *Operands Information*

To understand the table and text below, please refer to the [Intel TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.9: TDH.QUOTE.GET Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Explicit	RDX	N/A	Quote session context	N/A	RW	Hidden	N/A	Exclusive	N/A	N/A
Explicit	R8	HPA	HPA Linked List page	Blob	R	Shared	4KB	Shared	Shared	Shared
Explicit	R10	HPA	HPA Linked List page	Blob	RW	Shared	4KB	Shared	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	4KB	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(h)	N/A	N/A

3.3.2.6. *Completion Status Codes*

Table 3.10 TDH.QUOTE.GET Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_SUCCESS	Operation is successful
TDX_LIMIT_CPUID_MAXVAL_SET	IA32_MISC_ENABLES MSR bit 22 (Limit CPUID Maxval) is set.
TDX_INCONSISTENT_MSR	IA32_TSC_ADJUST MSR value is different than the value captured during the TDH.SYS.INIT interface function.
TDX_TSC_ROLLBACK	Time Stamp Counter value is lower than on last TD exit.
TDX_INTERRUPTED_RESUMABLE	Interrupt is pending, the operation can be resumed
TDX_EXT_NOT_READY	
TDX_EXT_NOT_INITIALIZED	
TDX_EXT_FATAL_ERROR	

Completion Status Code	Description
TDX_QUOTE_UNSUPPORTED_QUOTE_ID	QUOTE_ID is not supported
TDX_QUOTE_ NON_ENABLED_ QUOTE_ID	QUOTE_ID has not been enabled for runtime use
TDX_QUOTE_TDREPORT_ERROR	Failed to verify the TDREPORT
TDX_QUOTE_GEN_ERROR	Failed to generate Intel TDX Quote
TDX_QUOTE_SUCCESS_ATT_KEY_OUTDATED	Operation is successful, but the currently used key pair for this QUOTE_ID is not up-to-date with platform SVN