TRUSTED® COMPUTING GROUP

SPECIFICATION

TCG ACPI Specification

Version 1.4
Revision 15
April 3, 2024

Contact: admin@trustedcomputinggroup.org

PUBLISHED

DISCLAIMERS, NOTICES, AND LICENSE TERMS

CHANGE HISTORY

REVISION	DATE	DESCRIPTION
1.40	February 13, 2024	• Miscellaneous editorial updates • Adapt to new spec template • Adjust spec references for conventional BIOS and other specifications • Update Arm SMC start method to include CRB region size • Add AMD Mailbox start method • Add Arm FF-A start method

1 Contents

1.1 List of Tables

1 SCOPE

This specification defines the framework of necessary ACPI tables and basic methods to be used on a TCG compliant platform. The table and ACPI namespace objects provide enough information to the operating system to enable access to the TCG compliant hardware in a platform.

1.1 Key Words

The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this document normative statements are to be interpreted as described in RFC-2119, Key words for use in RFCs to Indicate Requirement Levels.

1.2 Statement Type

Please note a very important distinction between different sections of text throughout this document. There are two distinctive kinds of text: informative comment and normative statements. Because most of the text in this specification will be of the kind normative statements, the authors have informally defined it as the default and, as such, have specifically called out text of the kind informative comment. They have done this by flagging the beginning and end of each informative comment and highlighting its text in gray. This means that unless text is specifically marked as of the kind informative comment, it can be considered a kind of normative statement.

EXAMPLE: Start of informative comment

This is the first paragraph of 1–n paragraphs containing text of the kind *informative comment* ...

This is the second paragraph of text of the kind *informative comment* ...

This is the nth paragraph of text of the kind *informative comment* ...

To understand the TCG specification the user must read the specification. (This use of MUST does not require any action).

End of informative comment

1.3 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

The normative references for TPMs are:

1. The TPM Main Specification Level 2 Version 1.2
   Or
2. The TPM library Specification, Family “2.0”

The normative references for PC Client platforms are:

3. TCG PC Client Platform TPM Profile (PTP) Specification for TPM 2.0 Version 1.05 – in this document referred to as “PC Client PTP”
4. TCG PC Client Platform Firmware Profile (PFP) Specification, version 1.04 – in this document referred to as “PC Client PFP”
5. Advanced Configuration and Power Interface Specification, Revision 5.0 – in this document referred to as “ACPI Specification” (https://uefi.org/acpi/specs)

The normative references specifically for ARM platforms are:

6. SMC Calling Convention for System Software on Arm® Platforms, Document number: Arm DEN 0028B – in this document referred to as “SMC Calling Convention”
7. TPM Service Command Response Buffer Interface Over FF-A, ARM DEN 0138

1.4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

Term	Definition
buffer	A data structure used for transport to and from the TPM.
CLEAR	A bit with a value of zero (0), or the action of causing a bit to have a value of zero (0).
command	The values sent to the TPM to indicate the operation to be performed.
octet	Eight bits of data. **Note: Start of informative comment** On most modern computers, this is the smallest addressable unit of data. **End of informative comment**
response	The values returned by the TPM when it completes processing of a command.
SET	A bit with a value of one (1), or the action of causing a bit to have a value of one (1).\

1.5 Abbreviations

For the purposes of this document, the following abbreviations apply.

Abbreviation	Description
_CID	Compatibility ID
_HID	Hardware ID
TPM	Prefix for an indication passed from the system interface of the TPM to a Protected Capability defined in a TPM specification
ACPI	Advanced Configuration and Power Interface
ASL	ACPI Source Language
APIC	Advanced Programmable Interrupt Controller
CPU	Central Processing Unit
DSDT	Differentiated System Description Table
DWORD	Double word, a 32-bit number
FADT	Fixed ACPI Description Table (See [5] for the definition)
GAS	Generic Address Structure (See [5] for the definition)
GPE	ACPI General Purpose Event register block (See [5] for definition)
GPEx_STS	ACPI GPE Status register (part of the GPE register block specific for the event denoted by ‘x’ (0 or 1))
GSIV	Global System Interrupt Vector
ID	Identifier
I/O	Input/Output
LAML	Log Area Minimum Length
LASA	Log Area Start Address
MMIO	Memory Mapped IO
MSO	Most Significant Octet
NO-OP	No operation. A CPU operation that does nothing.
OEM	Original Equipment Manufacturer
OS	Operating System
OSPM	Operating System Power Management
PCI	Peripheral Component Interconnect
PNP	Plug and Play
SAPIC	Streamlined APIC
SCI	System Control Interrupt
TCG	Trusted Computing Group
TPM	Trusted Platform Module
TPM2_	Prefix for a command defined in the TPM 2.0 Library specification
QWORD	Quad Word, a 64-bit number
WORD	a 16-bit number

1.5.1 Bit and Octet Numbering and Order

An integer value is an array of one or more octets. The octet at offset zero within the array is the most significant octet (MSO) of the integer. Bit number 0 of that integer is its least significant bit in the last octet in the array.

Start of informative comment

Example: A 32-bit integer is an array of four octets; the MSO is at offset [0], and the most significant bit is bit number 31. Bit zero of this 32-bit integer is the least significant bit in the octet at offset [3] in the array.

Note: Array indexing is zero-based.

Note: This definition does not match the “network bit order” used in many IETF documents, such as RFC 4034. In those documents, the most significant bit of a datum has the lowest bit number. It is conventional practice to send that bit first when using a serial network protocol, and the bits are numbered in the order they are sent. This specification numbers bits according to their corresponding power of two within a datum. This numbering corresponds to the normal convention for bit numbering in hardware registers that hold integer values rather than fixed-point numbers.

End of informative comment

The first listed member of a structure is at the lowest offset within the structure and the last listed member is at the highest offset within the structure.

For a character string (letters delimited by “”), the first character of the string contains the MSO.

1.5.2 Numbers

Numbers are decimal unless a different radix is indicated.

Unless the number appears in a table intended to be machine readable, the radix is a subscript following the digits of the number. Only radix values of 2 and 16 are used in this specification.

Radix 16 (hexadecimal) numbers have a space separator between groups of two hexadecimal digits.

Start of informative comment

Example:

40 FF 12 34_{16}

End of informative comment

Radix 2 (binary) numbers use a space separator between groups of four binary digits.

Start of informative comment

Example:

$0100\ 1110\ 0001_2$

End of informative comment

For numbers using a binary radix, the number of digits indicates the number of bits in the representation.

Start of informative comment

Examples:

20_{16} is a hexadecimal number that contains exactly 8 bits and has a decimal value of 32.

$10\ 0000_2$ is a binary number that contains exactly 6 bits and has a decimal value of 32.

$0\ 20_{16}$ is a hexadecimal number that contains exactly 12 bits and has a decimal value of 32.

End of informative comment

A number in a machine-readable table may use the “0x” prefix to denote a base 16 number. In this format, the number of digits is not always indicative of the number of bits in the representation.

Start of informative comment

Example:

0x20 is a hexadecimal number with a value of 32, and the number of bits is determined by the context.

End of informative comment

2 Compliance

Systems providing an ACPI table for TPM 1.2 or TPM 2.0 MUST adhere to the layout of the tables described in this specification. A TCG compliant platform that implements ACPI SHALL provide the ACPI tables appropriate for the platform type.

3 ACPI Table

All TCG platforms supporting ACPI utilize the same header section layout, which is separated from the rest of the table with a double line. Table 2 describes the client ACPI table for TPM 1.2. Table 4 describes the server ACPI table for TPM 1.2. Table 7 describes the ACPI table for TPM 2.0, which can be used for client or server platforms.

Start of informative comment

The value in the signature field for the TPM 2.0 table ('TPM2') differs from the value for the TPM 1.2 tables ('TCPA').

ACPI tables have the little-endian byte format defined in the ACPI specification [5].

End of informative comment

3.1 Client ACPI Table for TPM 1.2

3.1.1 Client Common Header Values

These are the specific values to be used in the client platform version of the ACPI table for TPM 1.2.

Table 1: Defined values for Client ACPI Table for TPM 1.2

Field	Value	Description
Length	50	Size of the table
Revision	2	Revision for PC Client Platform Class
Platform Class	0	PC Client Platform Class

3.1.2 ACPI Table Layout

Table 2: TCG Hardware Interface Description Table Format for TPM 1.2 Clients

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'TCPA'. Signature for the TCG Hardware Interface Table.
Length	4	4	See Section 3.1.1. The length of this table starting from the Signature field up to and including the LASA field. It does not include the size of the area storing events or other data that is referenced or pointed to by any of these fields.
Revision	1	8	See Section 3.1.1. Revision of this table including the data and structures referenced by it. E.g., if the event structures with the area referenced by LASA change, this revision SHALL be incremented. **Note:** The purview of this revision is within the platform class as indicated by the Platform Class field. This means that each platform class increments this field autonomously. Software referencing this table should interpret the Platform Class field prior to interpreting this Revision field.
Checksum	1	9	Entire table MUST sum to zero.
OEMID	6	10	OEM ID. Per ACPI specification [5]. An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	For the TPM Interface Table, the table ID is the manufacturer model ID (assigned by the OEM identified by "OEM ID").
OEM Revision	4	24	OEM revision of TPM Interface Table for the given OEM Table ID. Per ACPI specification [5], this is "An OEM-supplied revision number. Larger numbers are assumed to be newer revisions."

Field	Byte Length	Byte Offset	Description
Creator ID	4	28	Vendor ID of the utility that created the table. For the tables containing Definition Blocks, this is the ID of the ASL Compiler.
Creator Revision	4	32	Revision of the utility that created the table. For the tables containing Definition Blocks, this is the revision of the ASL Compiler.
Platform Class	2	36	See Section 3.1.1.
Log Area Minimum Length (LAML)	4	38	Identifies the minimum length (in bytes) of the system's pre-boot TCG event log area. **Note**: The "PC Client PFP" specification [4] defines a minimum log size of 64KB.
Log Area Start Address (LASA)	8	42	Contains the 64-bit physical address of the start of the system's pre-boot TCG event log area, in QWORD format. **Note**: The log area ranges from address LASA to LASA+(LAML-1).

3.2 Server ACPI Table for TPM 1.2

3.2.1 Server Common Header Values

These are the specific values to be used in the server platform version of the ACPI table for TPM 1.2.

Table 3: Defined values for Server ACPI table for TPM 1.2

Field	Value	Description
Length	100	Size of the table
Revision	2	Revision for Server Platform Class
Platform Class	1	Server Platform Class

3.2.2 ACPI Table Layout

Table 4: TCG Hardware Interface Description Table Format for TPM 1.2 Servers

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'TCPA'. Signature for the TCG Hardware Interface Table.
Length	4	4	See Section 3.2.1. The length of this table starting from the Signature field up to and including the PCI Function Number field. It does not include the size of the area storing events or other data that is referenced or pointed to by any of these fields.
Revision	1	8	See Section 3.2.1. Revision of this table including the data and structures referenced by it. E.g., if the event structures with the area referenced by LASA change, this revision SHALL be incremented. **Note:** The purview of this revision is within the platform class as indicated by the Platform Class field. This means that each platform class increments this field autonomously. Software referencing this table should interpret the Platform Class field prior to interpreting this Revision field.
Checksum	1	9	Entire table MUST sum to zero.

Field	Byte Length	Byte Offset	Description
OEMID	6	10	OEM ID. Per ACPI specification [5]. An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	For the TPM Interface Table, the table ID is the manufacturer model ID (assigned by the OEM identified by “OEM ID”).
OEM Revision	4	24	OEM revision of TPM Interface Table for the given OEM Table ID. Per ACPI specification [5], this is “An OEM-supplied revision number. Larger numbers are assumed to be newer revisions.”
Creator ID	4	28	Vendor ID of the utility that created the table. For the tables containing Definition Blocks, this is the ID of the ASL Compiler.
Creator Revision	4	32	Revision of the utility that created the table. For the tables containing Definition Blocks, this is the revision of the ASL Compiler.
Platform Class	2	36	See Section 3.2.1.
Reserved	2	38	0 **Note:** This creates natural alignment for the fields that follow.
Log Area Minimum Length (LAML)	8	40	Identifies the minimum length (in bytes) of the system’s pre-boot TCG event log area.
Log Area Start Address (LASA)	8	48	Contains the 64-bit physical address of the start of the system's pre-boot TCG event log area, in QWORD format. **Note**: The log area ranges from address LASA to LASA+(LAML-1).
Specification Revision	2	56	Identifies the TCG specification revision, in BCD format, to which the interface was designed. The first byte holds the most significant digits, while second byte holds the least significant digits of the revision, e.g., a value of 01 02_{16} indicates the interface is compatible with TCG specification v1.2.
Device Flags	1	58	See Table 5
Interrupt Flags	1	59	See Table 6
GPE	1	60	The bit assignment of the SCI within the GPEx_STS register of a GPE described in the FADT that the interface triggers. **Note*:*** This field is valid only if Bit[2] of the Interrupt Flags field is set.)
Reserved	3	61	0
Global System Interrupt	4	64	The I/O APIC or I/O SAPIC Global System Interrupt used by the interface. **Note*:*** This field is valid only if Bit[3] of the Interrupt Flags field is set.
Base Address	12	68	The base address of the hardware register set described using the GAS. The Address_Space_ID field in the GAS can only have the values of 0 (System Memory) and 1 (System IO). All other values are not permitted. This address MUST be the Host Side address in the case of MMIO, and it MUST be the Host Side IO port address in the case of IO Port.
Reserved	4	80	Set to 0. This is to naturally align the data fields that follow.

Field	Byte Length	Byte Offset	Description
Configuration Address	12	84	The configuration address of the TPM hardware device described using the GAS. The Address_Space_ID field in the GAS can only have the values of 0 (System Memory) and 1 (System IO). All other values are not permitted. This is only valid if Bit[2] of the Device Flags field is set. This address MUST be the Host Side address in the case of MMIO, and it MUST be the Host Side IO port address in the case of IO Port.
PCI Segment Group Number	1	96	PCI Segment Group Number, if the TPM device is a PCI device
PCI Bus Number	1	97	PCI Bus Number, if the TPM device is a PCI device
PCI Device Number	1	98	Bit 4:0 – PCI Device Number: The PCI device number if the TPM device is a PCI device. Bit 7:5 – Reserved
PCI Function Number	1	99	Bit 2:0 – PCI Function Number: The PCI function number if the TPM device is a PCI device. Bit 7:3 – Reserved

3.2.3 Device Flags

Table 5: Bit layout of Device Flags

Bit	Description
7..3	Reserved
2	TPM configuration address valid 0 = TPM configuration address is invalid 1 = TPM configuration address is valid
1	TPM Bus is PNP 0 = FALSE (the TPM address and interrupt cannot be changed) 1 = TRUE (the TPM address and interrupt can be changed by PNP OS code)
0	PCI Device Flag. For PCI TCG devices, this bit is set. 0 = non-PCI device (The PCI Segment Group, Bus, Device and Function Number fields combined correspond to the ACPI _UID value of the device whose _HID or _CID contains a TPM plug and play ID.) 1 = PCI Device

3.2.4 Interrupt Flags

Table 6: Bit layout of Interrupt Flags

Bit	Description
7..4	Reserved
3	I/O APIC/SAPIC interrupt (Global System Interrupt) 0 = not supported 1 = supported
2	SCI triggered through GPE 0 = not supported 1 = supported
1	Interrupt Polarity, 0 = Active-High: This interrupt is sampled when the signal is high, or true. 1 = Active-Low: This interrupt is sampled when the signal is low, or false.

Bit	Description
	Note: PCI devices are always active low, so this bit is set to 1 for PCI devices.
0	Interrupt Mode, 0 = Level-Triggered: This interrupt is triggered in response to the signal being in either a high or low state. 1 = Edge-Triggered: This interrupt is triggered in response to a change in signal state, either high to low or low to high. **Note*:*** PCI devices are always level triggered, so this bit is set to 0 for PCI devices.

3.3 ACPI Table for TPM 2.0

This is the definition for the ACPI table for TPM 2.0.

Table 7: TCG Hardware Interface Description Table Format for TPM 2.0

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'TPM2'. Signature for the TCG Hardware Interface Table.
Length	4	4	The length of this table starting from the Signature field up to and including the Start Method specific parameters field. (52 + size of Start Method specific parameters) If the table contains LAML and LASA field, Length is 80.
Revision	1	8	5. The current revision of this table.
Checksum	1	9	Entire table MUST sum to zero.
OEMID	6	10	OEM ID. Per ACPI specification [5]. An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	For the TPM Interface Table, the table ID is the manufacturer model ID (assigned by the OEM identified by "OEM ID").
OEM Revision	4	24	OEM revision of TPM Interface Table for the given OEM Table ID. Per ACPI specification [5], this is "An OEM-supplied revision number. Larger numbers are assumed to be newer revisions."
Creator ID	4	28	Vendor ID of the utility that created the table. For the tables containing Definition Blocks, this is the ID of the ASL Compiler.
Creator Revision	4	32	Revision of the utility that created the table. For the tables containing Definition Blocks, this is the revision of the ASL Compiler.
Platform Class	2	36	0 for client platforms. 1 for server platforms.
Reserved	2	38	0
Address of CRB Control Area or FIFO Base Address	8	40	For interfaces that use the Command Response Buffer, this field SHALL be the physical address of the Control Area. The Control Area contains status registers and the location of the memory buffers for communicating with the device. The area may be in either TPM 2.0 device memory or in memory reserved by the system during boot. Interfaces that do not require the Control Area SHALL set this value to zero. For a TPM implementation based on the "PC Client PTP" specification [3] the address of the Control Area SHALL be the address of the TPM_CRB_CTRL_REQ_0 register. For FIFO interfaces with a fixed physical base address as defined in "PC Client PTP" specification [3], this field may be set to zero. For interfaces that use a FIFO interface as defined in the PTP without a fixed base address, this field SHALL be the base address of the FIFO interface.
Start Method	4	48	The Start Method selector determines which mechanism the device driver uses to notify the TPM 2.0 device that a command is available for processing. This field SHALL contain one of the values specified in Table 8.
Start Method Specific Parameters	Variable (up to 16)	52	The content of the Start Method specific parameters is determined by the Start Method used by the system's TPM device interface. This field contains values that may be used to initiate command processing.

Field	Byte Length	Byte Offset	Description
			If the Start Method value is 2, then this field is at least four bytes in size and the first four bytes MUST be all zero. If the Start Method value is 11 then this field is 12 bytes in size and is described in Table 9. If the Start Method value is 13, then this field is 16 bytes in size and is described in Table 10. If the Start Method value is 15 then this field is 12 bytes in size and is described in Table 11. If LAML and LASA are present, then this field is 16 bytes in size.
Log Area Minimum Length (LAML)	4	68	Optional. Identifies the minimum length (in bytes) of the system's pre-boot TCG event log area. **Note**: The "PC Client PFP" specification [4] defines a minimum log size of 64KB.
Log Area Start Address (LASA)	8	72	Optional. Contains the 64-bit physical address of the start of the system's pre-boot TCG event log area, in QWORD format. **Note**: The log area ranges from address LASA to LASA+(LAML-1). **Note**: The format of the TCG event log area is defined in the "PC Client PFP" specification [4], Section 9. The crypto agile log format as defined by the "PC Client PFP" specification [4] should be used.

Table 8: Start Method values for ACPI table for TPM 2.0

Value	Description
0	Not allowed (indicates value has not been set).
1	Reserved for legacy use (vendor specific).
2	Uses the ACPI Start method.
3 – 5	Reserved for legacy use (vendor specific).
6	Reserved for the Memory mapped I/O Interface (TIS 1.2+Cancel).
7	Uses the Command Response Buffer Interface.
8	Uses the Command Response Buffer Interface with the ACPI Start Method.
9 – 10	Reserved for legacy use (vendor specific).
11	Uses the Command Response Buffer Interface with Arm Secure Monitor or Hypervisor Call (SMC/HVC)
12	Uses the FIFO Interface over I2C bus.
13	Uses the Command Response Buffer Interface with AMD Mailbox specific notification.
14	Reserved for future Memory mapped I/O Interface.
15	Uses the Command Response Buffer Interface with Arm Firmware Framework-A
16+	Reserved for future use

3.3.1 Start Method Specific Parameters for Arm SMC Start Method (11)

The following table describes the definition of the Start Method specific parameters when the ACPI Start Method value is 11.

Table 9: Start Method Specific Parameters for Arm SMC

Field	Byte Length	Byte Offset	Description
Interrupt	4	52[1]	Global System Interrupt Vector of the TPM interrupt. MUST be zero if interrupt is not supported
Flags	1	56	Bits 7:3 Reserved for future use Bit 2 – Attribute field valid 1: Attributes field contains valid data 0: Attributes field does not contain valid data Bit 1 – Hypervisor call 1: Hypervisor call (HVC) MUST be used for the Start Method 0: Secure monitor call (SMC) MUST be used for the Start Method Bit 0 – Interrupt support 1: Interrupt is supported. Interrupt is always edge triggered when using Arm SMC. 0: No Interrupt, (software MUST poll CRB status)
Operation Flags	1	57	Bits 7:1 – Reserved for future use Bit 0 – TPM Idle Support 1: CRB interface state transitions include “Idle”, “Ready”, “Reception”, “Execution”, and “Completion” 0: CRB interface state transitions only include “Ready” and “Execution” states
Attributes	1	58	Bits 7:4 – Reserved for future use Bits 3:2 – CRB region Size. This field specifies the per-locality size of the CRB region encompassing all registers and the command/response buffer. Values are defined as follows: 00 – 4 KiB 01 – 16 KiB 10 – 64 KiB Bits 1:0 – Memory Type. This field specifies the memory attributes of the CRB control area and command/response buffers. Values are defined as follows: 00 – Not cacheable. Device non-Gathering, non-Reordering, no Early Write Acknowledgement. 01 – Write combine. Normal memory, Outer non-cacheable, Inner non-cacheable. 10 – Write through. Normal Memory, Outer Write-through non-transient, Inner Write-through non-transient. 11 – Write back. Normal Memory, Outer Write-back non-transient, Inner Write-back non-transient. The memory attributes of the CRB defined here MUST be consistent with the attributes of the CRB region described in the EFI memory map.
Reserved	1	59	0

[1] Byte Offset of Start Method Specific Parameters starts from 52 per **Error! Reference source not found.**.

Field	Byte Length	Byte Offset	Description
SMC/HVC Function ID	4	60	This field provides the SMC/HVC call function ID that will invoke the TPM start method. Firmware SHALL implement the SMC call as an SMC32 or SMC64 Fast Call, compliant with the “SMC Calling Convention” specification [6].The call takes no client ID, no Secure OS ID, and no Session ID as parameters. The call SHALL return zero. The function ID SHALL be allocated from a Service Call Range over which the platform vendor has authority.

3.3.2 Start Method Specific Parameters for AMD Mailbox (13)

The following table describes the definition of the Start Method specific parameters when the ACPI Start Method value is 13.

Table 10: Start Method Specific Parameters for AMD Mailbox

Field	Byte Length	Byte Offset	Description
TPM Start Address	8	52[1]	Contains the 64-bit physical address of the 32-bit activation register to indicate a command or register update is ready in the ControlArea buffer for the TPM. Set this 32-bit value to 1 to indicate to the TPM the message is ready.
TPM Reply Address	8	60	Contains the 64-bit physical address of the 32-bit reply register to indicate the CRB Control Area or Response Area has been updated by the TPM. TPM sets this 32-bit value to 1 when complete.

3.3.3 Start Method Specific Parameters for Arm FF-A Start Method (15)

The following table describes the definition of the Start Method specific parameters when the ACPI Start Method value is 15. See the TPM Command Response Buffer Interface over FF-A [7] specification for start method ABI details.

Table 11: Start Method Specific Parameters for Arm FF-A

Field	Byte Length	Byte Offset	Description
Flags	1	52	Bits 7:1 Reserved for future use Bit 0 – FF-A Notification Support 1: Notifications are supported 0: Notifications are not supported, (software MUST poll CRB status)
Attributes	1	53	Bits 7:4 – Reserved for future use Bits 3:2 – CRB region Size. This field specifies the per-locality size of the CRB region encompassing all registers and the command/response buffer. Values are defined as follows: 00 – 4 KiB 01 – 16 KiB 10 – 64 KiB Bits 1:0 – Memory Type. This field specifies the memory attributes of the CRB control area and command/response buffers. Values are defined as follows: 00 – Not cacheable. Device non-Gathering, non-Reordering, no Early Write Acknowledgement. 01 – Write combine. Normal memory, Outer non-cacheable, Inner non-cacheable. 10 – Write through. Normal Memory, Outer Write-through non-transient, Inner Write-through non-transient. 11 – Write back. Normal Memory, Outer Write-back non-transient, Inner Write-back non-transient. The memory attributes of the CRB defined here MUST be consistent with the attributes of the CRB region described in the EFI memory map.
Partition ID	2	54	The partition ID of the FF-A secure partition that implements the TPM service. If the partition ID is 0, the client MUST discover the partition ID through FF-A using the UUID defined in the FF-A start method ABI [7].
Reserved	8	56	0

4 ACPI Device

Start of informative comment

Devices are not required to be exposed in ACPI namespace if the device exists on a bus that is enumerable by the Operating System, or on a bus which is plug-n-play capable.

A TCG platform class-specific ACPI Table may provide a mechanism that can be used before the ability to execute ACPI control methods in the OS is available. The Server ACPI Table in Section 3.2 of this specification is one example of such a table. This table is not, however, intrinsically supported in the OSPM as a way of discovering and reporting system resources. Therefore, it is recommended that non-PCI TCG Hardware on the platform be described in the ACPI name space. This makes it possible for the OSPM to enumerate the TCG Hardware as a device. In addition, the ACPI name space description is more flexible and friendly in hot-plug scenarios.

Note that to be ACPI compatible, the fixed resources for TCG Hardware must still be accounted for in accordance with the ACPI specification. If the device is not formally described in the ACPI Name Space, its resources must be described as fixed system resources or the resources appended to some other fixed resource system device in order to ensure that the OSPM does not attempt to allocate those resources to some other device.

To formally describe the TCG Hardware System Interface in ACPI Name Space, a TCG hardware device is created using the named device object. Table 12 is a non-exhaustive list of ACPI control methods that may be used in a TCG hardware device object, along with a recommended support level for each method.

End of informative comment

A TCG platform MAY provide an ACPI device object representing the TPM in the ACPI namespace if the bus where the TPM is located is not PNP capable or the bus is not exposed to the OS for PNP operations.

It is recommended that TPM 2.0 device object ACPI table appears under the DSDT table in the ACPI namespace. The TPM 2.0 device object should be located under the system bus.

Table 12: TCG Hardware Device Object Control Methods

Object	Description	Support Level
_ADR	Named object that evaluates to the interface's address on its parent bus. _ADR is a standard device configuration control method defined in the ACPI Specification [5].	Required only for devices on a bus that has standard enumeration mechanism.
_HID	Named object that provides the interface's Plug and Play identifier. This value may be TPM vendor specific. _HID is a standard device configuration control method defined in the ACPI Specification [5].	Required only for devices that do not have standard enumeration mechanism.
_CID	Named object (or list of named objects) that provide alternative Plug and Play identifiers. Depending on the platform, the _CID can be used to identify a compatible class driver for the TPM device. _CID is a standard device configuration control method defined in the ACPI Specification [5].	Optional
_STR	Named object that evaluates to a Unicode string that may be used by an OS to provide information to an end user describing the device. __STR is a standard device configuration control method defined in the ACPI Specification [5].	Optional
_UID	Named object that specifies a device's unique persistent ID, or a control method that generates it. _UID is a standard device configuration control method defined in the ACPI Specification [5].	Optional
_CRS	Named object that returns the TPM interface's current resource settings. Security hardware Interfaces are considered static resources; hence only return their defined resources. The address region definition is interface type/subtype dependent. _CRS is a standard device configuration control method defined in the ACPI Specification [5].	Required
_STA	Object that returns the status of the device: enabled, disabled or removed, as defined in the ACPI Specification [5]. If this method is not present, the device is assumed to be enabled.	Recommended

Object	Description	Support Level
_DSM	Device Specific Method Function 0 – standard query function Function 1 – TCG Hardware Information Arguments: Arg0 (Buffer): UUID = {CF8E16A5-C1E8-4e25-B712-4F54A96702C8} Arg1 (Integer): Revision ID = 1 Arg2 (Integer): Function Index = 1 Arg3 (Package): Arguments = Empty Package Returns: Type: Package Package item 1: Type: Integer Purpose: status of operation Description: 0: Failure 1: Success Package item 2: Type: Package Purpose: TCG Revision implemented in security hardware Package item 1: Type: Integer Description: (BCD format) – most significant digits of TCG version Package Item 2: Type: Integer Description: (BCD format) – least significant digits of TCG version For example: a value of 0x0110 indicates the interface is compatible with TCG specification v1.1.	Optional
_GPE	Named object that evaluates to either an integer or a package. If _GPE evaluates to an integer, the value is the bit assignment of the SCI within the GPEx_STS register of a GPE block described in the FADT that the Security hardware device will trigger. If _GPE evaluates to a package, then that package contains two elements. The first is an object reference to the GPE Block device that contains the GPE register that will be triggered by the interface. The second element is numeric (integer) that specifies the bit assignment of the SCI within the GPEx_STS register of the GPE Block device referenced by the first element in the package. **Note:** This object is only provided if the interface supports a GPE.	Required if interrupt through GPE is supported

4.1 Optional Start Method for TPM 2.0 devices

Some platforms may implement an optional ACPI Start Method to permit the OS to request the firmware to execute or cancel a TPM 2.0 command. The use of the ACPI Start method is determined by the StartMethod field of the static TPM2 ACPI table (see Section 3.3). If the StartMethod field of the static ACPI table has the value of 2 or 8, which indicates the use of this method, the ACPI Start Method SHALL be implemented. The ACPI functions defined herein SHALL reside in the _DSM control method object.

The _DSM method defined herein SHALL be implemented as follows:

Function 0 – standard query function

Function 1 – ACPI Start Method

Arguments:

Arg0 (Buffer): UUID = 6bbf6cab-5463-4714-b7cd-f0203c0368d4

Arg1 (Integer): Revision ID = 0

Arg2 (Integer): Function Index = 1

Arg3 (Package): Arguments = Empty Package

Return Value:

Type: Integer

Description of the Return Value:

0: Success

1: General Failure

Functional Behavior:

This function tells the system to review the fields in the TPM 2.0 device control area and take the appropriate action, such as, but not limited to, execute or cancel a TPM 2.0 command.

Start of informative comment

The function is non-blocking. The call will return immediately. When Return Value 0 is returned, the TPM inspected the control area and will take appropriate action. For instance, if a command has been submitted and Start is SET, the command was accepted and will be executed by the TPM. Whenever possible the system should return a TPM response instead of a General Failure from this call. For example, if a command cannot be processed when the method is called, a Response buffer of TPM2_RC_Retry could be written to the TPM's response area and the Start field in the Control Area could be CLEARed. If the command was cancelled because the Cancel field was set, this method may write a TPM2_RC_Cancelled return code in the Response buffer, clear the Start field in the Control Area and return a value of 0. Alternatively, if the Cancel field is set the method may return a value of 0 and the TPM 2.0 device may later complete or cancel the command per the requirements for cancelling a command.

When 1 is returned, the firmware is unable to read or act upon the request. Other than the return value, the request is equivalent to a NO-OP. Examples are (a) a bad OS driver requests execution of additional commands before a prior command completed, (b) the control area is not in physical memory, or (c) the command or response physical addresses do not exist. A return value of 1 may cause software to stop using the TPM 2.0 device until the next full system boot (this excludes hibernation/resume cycles).

End of informative comment