TCG Specification

TPM 2.0 Mobile Common Profile

Family "2.0"

Level 00 Revision 31

21 December 2015

Contact: admin@trustedcomputinggroup.org

TCG Published

TCG

Disclaimer

Disclaimer

TCG acknowledges the following contributors to this specification:

Alec Brusilovsky, Amy Nelson, Andreas Fuchs, Bo Bjerrum, Carey Huscroft, Carlin Covey, Cedric Colnot, David Challener, David Wooten, Dean Liberty, Emily Ratliff, Gilles Peskine, Graeme Proudler, Greg Kimberly, Hervé Sibert, Ira McDonald, Jan-Erik Ekberg, John Mersh, Joshua Schiffman, Kathleen McGill, Laszlo Hars, Michael Chan, Michael Peck, Mohamed Tabet, Monty Wiseman, Niall O'Donoghue, Nicolas Ponsini, Paul England, Paul Waller, Seigo Kotani, Steve Hanna, Sung Lee, Tom Moulton.

Table of Contents

List of Tables

1 Scope and Audience

The Trusted Computing Group TPM 2.0 Library Specification [1] defines a Trusted Platform Module (TPM). This TPM 2.0 Mobile Common Profile specification defines a profile of the TPM 2.0 Library Specification [1] that is applicable to all mobile devices (smartphones, feature phones, basic phones, etc.) that claim conformance to the TPM 2.0 Mobile Reference Architecture [4] and is optimized for ease-of-implementation in feature phones, basic phones, eBook readers, and other similar constrained mobile devices. The target audience for this specification includes designers, developers, and implementers of Trusted Computing technologies in mobile platforms.

The common industry terms for classes of mobile phones are:

- Basic phone – typically supports only basic cellular telephony and text messaging (SMS – Short Message Service).
- Feature phone – typically supports: (a) consumer-oriented features; (b) single-user usage model; (c) single rich OS; (d) no virtualization; (e) limited Internet access (web browser and email); and (f) user installation of selected applications (often under the control of mobile network operators).
- Smartphone – typically supports: (a) consumer and enterprise features; (b) multiple-user usage models; (c) sandboxing, virtualization, etc. for application isolation; (d) full Internet access (web, email, many other application protocols); (e) Wi-Fi, Bluetooth, NFC (Near Field Connect) and other wireless protocols; and (f) user installation of many applications (usually not under the control of mobile network operators).

1.1 Key words

The upper-case key words “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in this document indicate normative statements and are to be interpreted as described in [2].

1.2 Statement Type

There are two separate kinds of text throughout this specification: *informative comments* and *normative statements*.

All conforming implementations SHALL obey all normative statements of requirements. All conforming implementations SHOULD obey all normative statements of recommendations. Normative statements always include one or more of the uppercase key words “REQUIRED,” “SHALL,” “SHOULD,” or “RECOMMENDED” listed in section 1.1 above.

1.3 References

[1] Trusted Computing Group, Trusted Platform Module Library 2.0 Revision 1.16, Parts 1-4, October 2014

[2] IETF, Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, March 1997

[3] Trusted Computing Group, PC Client Platform TPM Profile (PTP) Specification, January 2015

[4] Trusted Computing Group, TPM 2.0 Mobile Reference Architecture 2.0 Revision 142, December 2014

[5] Trusted Computing Group, TCG Algorithms Registry Revision 1.22, February 2015

[6] Trusted Computing Group, TCG EFI Platform Specification (for TPM Family 1.1 or 1.2) 1.22 Revision 15, January 2014

2 Definition of Mobile Common Profile

A TPM Mobile compliant to this profile SHALL support all normative conformance requirements and SHOULD support all normative conformance recommendations specified in section 6 The Protected Environment, section 7 TPM Mobile Implementation, and section 8 TPM Mobile Identity and Ownership of TPM 2.0 Mobile Reference Architecture [4].

Informative examples of hardware architecture implementation alternatives for a TPM Mobile are described in Appendices A, B, and C of TPM 2.0 Mobile Reference Architecture [4]. Informative details of key management in a firmware TPM are described in Appendix D of TPM 2.0 Mobile Reference Architecture [4].

This TPM 2.0 Mobile Common Profile is applicable to all mobile devices that (smartphones, feature phones, basic phones, etc.) that claim conformance to the TPM 2.0 Mobile Reference Architecture [4] and is optimized for ease-of-implementation in feature phones, basic phones, eBook readers, and other similar constrained mobile devices as described in Section 1.

2.1 Mandatory TPM 2.0 Library Specification Version

A TPM Mobile compliant to this profile SHALL implement all supported TPM commands (see section 2.4) as specified in the TPM 2.0 Library Specification [1] version 1.16 or later version.

2.2 Mandatory Platform Constants

A TPM Mobile compliant to this profile SHALL support all TPM 2.0 standard platform constants defined in section 6.13 TPM_PT (Property Tag) of TPM 2.0 Library Part 2 [1], unless their description says that they are optional. A TPM Mobile compliant to this profile SHALL support all platform-specific constants defined as mandatory in the following table. A TPM Mobile compliant to this profile MAY support additional platform constants.

A TPM Mobile compliant to this profile SHALL return the correct day and year of the implemented Mobile Common Profile version for the TPM_PT_PS_DAY_OF_YEAR and TPM_PT_PS_YEAR platform-specific constants.

Table 1 – TPM Mandatory Platform-Specific Constants

Capability Name	Returned Value	Description
TPM_PT_PS_FAMILY_INDICATOR	0x00000003	TPM_PS_CELL_PHONE TPM 2.0 Mobile Common Profile Specification
TPM_PT_PS_LEVEL	0x00000000	TPM 2.0 Mobile Common Profile Specification Level 00
TPM_PT_PS_REVISION	0x00000100	TPM 2.0 Mobile Common Profile Specification Revision 1.00
TPM_PT_PS_DAY_OF_YEAR	0x00000355	The day of the year of the implemented TPM 2.0 Mobile Common Profile Specification
TPM_PT_PS_YEAR	0x00002015	The year of the implemented TPM 2.0 Mobile Common Profile Specification

2.3 Mandatory and Recommended Algorithms

A TPM Mobile compliant to this profile SHALL implement all algorithms listed in the following table unless those algorithms are marked as RECOMMENDED. A TPM Mobile compliant to this profile SHOULD implement the algorithms marked as RECOMMENDED. A TPM Mobile compliant to this provide MAY support additional algorithms not listed in the table.

Notes:

1) Standard names for TPM algorithms are defined in the TCG Algorithms Registry [5].

2) The mandatory algorithms listed in the following table are a proper subset of the PC Client Platform TPM Profile [3], except for the addition of SHA384 as RECOMMENDED (for TLS/1.3 support) and a set of symmetric algorithms as RECOMMENDED (for TLS/1.3 support and support of the TPM2_EncryptDecrypt command) in this TPM Mobile profile (see section 2.4).

3) The symmetric cipher mode TPM_ALG_CFB is REQUIRED by TCG TPM 2.0 Library specification Part 1 [1] and is also necessary for implementation of TPM2_Create, TPM2_Load, TPM 2_ContextSave, TPM2_ContextLoad, and other TPM commands.

Table 2 – TPM Mandatory and Recommended Algorithms

Algorithm Name	Reference / Comments
TPM_ALG_RSA	IETF RFC 3447 (support for 2048-bit keys is REQUIRED; support for 1024-bits keys is OPTIONAL)
TPM_ALG_HMAC	ISO/IEC 9797-2 (Hash Message Authentication Code)
TPM_ALG_AES	ISO/IEC 18033-3 (support for 128-bit keys is REQUIRED; support for other key sizes is OPTIONAL)
TPM_ALG_KEYEDHASH	TCG TPM 2.0 Library specification [1]
TPM_ALG_XOR	TCG TPM 2.0 Library specification [1]
TPM_ALG_SHA256	ISO/IEC 10118-3 (SHA-2 support for 256-bit keys is REQUIRED; support for 512-bit key sizes is OPTIONAL)
TPM_ALG_SHA384	RECOMMENDED – ISO/IEC 10118-3 (SHA-2 support for 384-bit keys will be mandatory for TLS/1.3)
TPM_ALG_NULL	TCG TPM 2.0 Library specification [1]
TPM_ALG_RSASSA	IETF RFC 3447 (signature algorithm – RSASSA-PKCS1-v1_5)
TPM_ALG_RSAES	IETF RFC 3447 (padding algorithm – RSAES-PKCS1-v1_5)
TPM_ALG_RSAPSS	IETF RFC 3447 (signature algorithm – RSASSA-PSS)
TPM_ALG_OAEP	IETF RFC 3447 (padding algorithm – RSAES-OAEP)
TPM_ALG_ECDSA	ISO/IEC 14888-3 (ECC signature algorithm)
TPM_ALG_ECDH	US NIST SP800-56A (ECC Diffie-Hellman for secret sharing)
TPM_ALG_ECDAA	RECOMMENDED – TCG TPM 2.0 Library specification [1] (support for ECDAA is necessary for TPM 2.0 Direct Anonymous Attestation)
TPM_ALG_ECSCHNORR	RECOMMENDED – TCG TPM 2.0 Library specification [1] (support for ECSCHNORR is necessary for elliptic-curve based Schnorr signatures)

Algorithm Name	Reference / Comments
TPM_ALG_KDF1_SP800_56A	US NIST SP800-56A (concatenation key derivation function)
TPM_ALG_KDF1_SP800_108	NIST SP800-108 (key derivation function)
TPM_ALG_ECC	ISO/IEC 15946-1 (Prime Field ECC)
TPM_ALG_SYMCIPHER	TCG TPM 2.0 Library specification [1] (object type for a symmetric block cipher)
TPM_ALG_CAMELLIA	RECOMMENDED – ISO/IEC 18033-3 (symmetric block cipher with various key sizes)
TPM_ALG_CTR	RECOMMENDED – ISO/IEC 10116 (Counter mode for all symmetric block ciphers)
TPM_ALG_OFB	RECOMMENDED – ISO/IEC 10116 (Output feedback mode for all symmetric block ciphers)
TPM_ALG_CBC	RECOMMENDED – ISO/IEC 10116 (Cipher Block Chaining mode for all symmetric block ciphers)
TPM_ALG_CFB	ISO/IEC 10116 (Cipher Feedback mode for all symmetric block ciphers)
TPM_ALG_ECB	RECOMMENDED – ISO/IEC 10116 (Electronic Codebook mode for all symmetric block ciphers)

2.3.1 Mandatory and Recommended ECC Curves

A TPM Mobile compliant to this profile SHALL implement all ECC curves listed in the following table unless those ECC curves are marked as RECOMMENDED. A TPM Mobile compliant to this profile SHOULD implement the ECC curves marked as RECOMMENDED. A TPM Mobile compliant to this provide MAY support additional ECC curves not listed in the table.

Notes:

1) Standard names for TPM ECC curves are defined in the TCG Algorithms Registry [5].

2) The mandatory ECC curves listed in the following table are a proper subset of the PC Client Platform TPM Profile [3].

Table 3 – TPM Mandatory and Recommended ECC Curves

Name	Comments
TPM_ECC_NIST_P256	US NIST Curve P-256 [5]
TPM_ECC_BN_P256	RECOMMENDED – Barreto-Naehrig Pairing 256 [5] (support for BN P256 is necessary for ECDAA support)

2.4 Mandatory and Recommended Commands

A TPM Mobile compliant to this profile SHALL implement all commands listed in the following table unless those commands are marked as RECOMMENDED. A TPM Mobile compliant to this profile SHOULD implement the commands marked as RECOMMENDED. A TPM Mobile compliant to this provide MAY support additional commands not listed in the table. A TPM Mobile compliant to this profile MAY support Physical Presence.

A TPM Mobile compliant to this profile SHALL support all normative conformance requirements and SHOULD support all normative conformance recommendations specified in section 8.5 TPM Mobile Startup and section 8.5.1 TPM2_Shutdown of TPM 2.0 Mobile Reference Architecture [4]

A TPM Mobile compliant to this profile SHALL support the command timeout requirements specified in Table 15 of the PC Client Platform TPM Profile [3] for all of the commands implemented in the TPM Mobile.

A TPM Mobile compliant to this profile SHOULD support the command duration recommendations specified in Table 15 of the PC Client Platform TPM Profile [3] for all of the commands implemented in the TPM Mobile.

Notes:

1) Standard names for TPM commands are defined in the TCG TPM 2.0 Library Specification [1].

2) The mandatory commands listed in the following table are a proper subset of the PC Client Platform TPM Profile [3], except for the addition of TPM2_EncryptDecrypt as RECOMMENDED (for symmetric cryptography with chaining support for bulk symmetric encryption/decryption) and TPM2_HMAC as mandatory (for single-buffer HMAC).

3) TPM2_EncryptDecrypt and symmetric algorithms support are important features for future mobile devices. However making these mandatory could result in the TPM Mobile implementation being classified as a bulk encryption device (i.e., subject to additional export controls). Therefore TPM2_EncryptDecrypt and the symmetric algorithms have been specified as RECOMMENDED to allow TPM Mobile suppliers and mobile device manufacturers to make independent choices about support for symmetric cryptography in specific markets.

4) See section 0 for rationale of detailed requirements for interface indication support in TPM Mobile implementations.

Table 4 – TPM Mandatory and Recommended Commands

Command Name	Comments
Interface Indications	
_TPM_Init	Means of indicating TPM initialization
Startup	
TPM2_Startup(CLEAR)	Used by the mobile device boot code for a discrete TPM Mobile or the Protected Environment for a firmware TPM Mobile to initialize the TPM with a fresh state (i.e., TPM Reset)
TPM2_Shutdown(CLEAR)	Used by the mobile device boot code for a discrete TPM Mobile or the Protected Environment for a firmware TPM Mobile to shutdown the TPM without saving any state
Testing	
TPM2_SelfTest	Used to cause the TPM Mobile to perform a test of its capabilities.
TPM2_IncrementalSelfTest	Used to cause the TPM Mobile to perform a test of the selected algorithms.
TPM2_GetTestResult	Used to return manufacturer-specific information regarding the results of a self-test and an indication of the test status.
Session Commands	
TPM2_StartAuthSession	Used to start an authorization session
Object Commands	
TPM2_Create	Used to create an object that can be loaded with TPM2_Load
TPM2_Load	Used to load public and private portions of an object previously created by the TPM
TPM2_LoadExternal	Used to load public portion of an object
TPM2_ReadPublic	Used to read public area of a loaded object
TPM2_Unseal	Used to return a loaded Sealed Data Object
TPM2_ActivateCredential	Used to associate a credential with an object in a way that ensures that the TPM has validated the parameters of the credentialed object
Duplicate Commands	
TPM2_Duplicate	Used to duplicate a loaded object (for export)
TPM2_Import	Used to import and encrypt an external object that can be loaded with TPM2_Load
Asymmetric Primitives	
TPM2_RSA_Encrypt	Used to encrypt data with RSA
TPM2_RSA_Decrypt	Used to decrypt data with RSA
TPM2_ECDH_KeyGen	Used to generate an ephemeral key pair
TPM2_ECDH_ZGen	Used to recover the Z value from a public point and a private key
TPM2_ECC_Parameters	Used to read the parameters of a selected ECC curve supported by the TPM

Command Name	Comments
Symmetric Primitives	
TPM2_EncryptDecrypt	RECOMMENDED – Used to perform symmetric encryption or decryption of a single data buffer, with chaining support for bulk symmetric encryption/decryption (note 1)
TPM2_Hash	Used to perform a selected hash on a single data buffer
TPM2_HMAC	Used to perform an HMAC computation on a single data buffer (note 1)
Random Number Generator	
TPM2_GetRandom	Used to read a string of random octets from an RNG implemented in the TPM
TPM2_StirRandom	Used to add additional information to the RNG as specified in US NIST SP800-90A
Hash/HMAC/Event Sequences	
TPM2_HMAC_Start	RECOMMENDED – Used to start an HMAC sequence.
TPM2_HashSequenceStart	RECOMMENDED – Used to start a hash or an Event Sequence.
TPM2_SequenceUpdate	RECOMMENDED – Used to add data to a hash or HMAC sequence.
TPM2_SequenceComplete	RECOMMENDED – Used to add the last part of data, if any, to a hash or HMAC sequence and returns the result.
TPM2_EventSequenceComplete	RECOMMENDED – Used to add the last part of data, if any, to an Event Sequence and return the result in a digest list.
Attestation Commands	
TPM2_Certify	Used to certify that a named object is loaded in the TPM
TPM2_CertifyCreation	Used to prove the association between an object and its creation data by validating a ticket that was previously produced by the TPM
TPM2_Quote	Used to quote PCR values in the TPM
Ephemeral EC Keys	
TPM2_Commit	RECOMMENDED – Used to perform the first part of an ECC anonymous signing operation (support is necessary for ECDAA)
TPM2_EC_Ephemeral	RECOMMENDED – Used to create an ephemeral key for use in a two-phase key exchange protocol – see TPM2_ZGen_2Phase
Signatures	
TPM2_Verify_Signature	Used to verify signatures created outside of the TPM
TPM2_Sign	Used to sign an externally provided hash
Command Audit	
<none>	<Command audit commands are OPTIONAL>
Integrity Collection (PCR)	
TPM2_PCR_Extend	Used to cause an update to the specified PCR with the specified list of hash values
TPM2_PCR_Event	Used to cause an update to the specified PCR with the specified event data
TPM2_PCR_Read	Used to read the values of all specified PCR

Command Name	Comments
TPM2_PCR_Allocate	RECOMMENDED – Used to allocated a new PCR bank
Enhanced Authorization (EA) Commands	
TPM2_PolicySigned	Used to tie policy to a signing key by including the Name of the signing key in the *policyDigest*
TPM2_PolicyOR	Used to allow options in authorizations without requiring that the TPM evaluate all of the options
TPM2_PolicyPCR	Used to cause conditional gating of a policy based on PCR
TPM2_PolicyNV	Used to cause conditional gating of a policy based on the contents of an NV Index
TPM2_PolicyCommandCode	Used to limit the authorization to a specific command code
TPM2_PolicyAuthorize	Used to allow policies to change, by letting a policy authority sign a new policy so that it may be used in an existing policy
TPM2_PolicyAuthValue	Used to allow a policy to be bound to the authorization value of the authorized entity
TPM2_PolicyPassword	Used to allow a policy to be bound to the authorization value of the authorized object
Hierarchy Commands	
TPM2_CreatePrimary	Used to create and load a Primary Object under one of the Primary Seeds or a Temporary Object under TPM_RH_NUL with a specified template for the object to be created
TPM2_HiearchyControl	Used to enable and disable use of a hierarchy and its associated NV storage
TPM2_Clear	Used to remove all TPM context associated with a specific Owner
TPM2_HierarchyChangeAuth	Used to allow the authorization secret for a hierarchy or lockout to be changed using the current authorization value as the command authorization
Dictionary Attack Functions	
TPM2_DictionaryAttackLockReset	Used to cancel the effect of a TPM lockout due to a number of successive authorization failures
TPM2_DictionaryAttackParameters	Used to change the lockout parameters
Field Upgrade	
TPM2_FieldUpgradeStart	RECOMMENDED – Used to pass *platformPolicy* and a TPM Vendor Authorization Key to authorize a Field Upgrade Manifest (support is appropriate for a discrete TPM Mobile)
TPM2_FieldUpgradeData	RECOMMENDED – Used to pass the actual field upgrade image to be installed on the TPM Mobile (support is appropriate for a discrete TPM Mobile) – the exact format of *fuData* is vendor-specific – this command is only possible following a successful TPM2_FieldUpgradeStart()
TPM2_FirmwareRead	RECOMMENDED – Used to read a copy of the current firmware installed in the TPM Mobile (support is appropriate for a discrete TPM Mobile)

Command Name	Comments
Context Management	
TPM2_ContextSave	Used to save a session context, object context, or sequence object context outside the TPM
TPM2_ContextLoad	Used to reload a context that has been saved by TPM2_ContextSave()
TPM2_FlushContext	Used to remove all context associated with a loaded object or session from TPM memory, but cannot be used to remove a persistent object from the TPM
TPM2_EvictControl	Used to allow a transient object to be made persistent or a persistent object to be evicted
Clocks and Timers	
TPM2_ReadClock	Used to read the current TPMS_TIME_INFO structure that contains the current setting of *Time*, *Clock*, *resetCount,* and *restartCount*
TPM2_ClockSet	Used to *advance* the value of the TPM Mobile's *Clock*
TPM2_ClockRateAdjust	Used to adjust the rate of advance of *Clock* and *Time* to provide a better approximation to real time
Capability Commands	
TPM2_GetCapability	Used to read various information regarding the TPM and its current state
TPM2_TestParms	Used to check if specific combinations of algorithm parameters are supported
Non-volatile Storage	
TPM2_NV_DefineSpace	Used to define the attributes of an NV Index and to cause the TPM to reserve space to hold the data associated with the NV Index
TPM2_NV_UndefineSpace	Used to remove an NV Index from the TPM
TPM2_NV_UndefineSpaceSpecial	Used to remove a platform-created NV Index that has TPMA_NV_POLICY_DELETE SET.
TPM2_NV_ReadPublic	Used to read the public area and Name of an NV Index
TPM2_NV_Write	Used to write a value to an area in NV memory previously defined by TPM2_NV_DefineSpace()
TPM2_NV_Increment	If an Index has TPMA_NV_COUNTER SET, this command may be used to increment the value in an NV Index by one
TPM2_NV_Extend	If an Index has TPMA_NV_EXTEND SET, this command may be used extend a value to an area in NV memory previously defined by TPM2_NV_DefineSpace
TPM2_NV_Read	Used to read a value from an area in NV memory previously defined by TPM2_NV_DefineSpace()

2.4.1 Mandatory Self-Test Support

A TPM Mobile compliant to this profile SHALL support self-test as defined in section 10 in TPM 2.0 Library Part 3 [1]. If a command is received that requires return of a value that depends on untested functions, then the TPM Mobile SHALL test the required functions before completing that command.

If a self-test fails at any time, the TPM Mobile will enter Failure mode. While in Failure mode, the TPM Mobile will return TPM_RC_FAILURE for any command other than TPM2_GetTestResult() and TPM2_GetCapability(). The TPM Mobile will remain in Failure mode until the next _TPM_Init.

Notes:

1) Support for incremental self-test is especially important for a firmware implementation of a TPM Mobile that is running in the single-core used by the Boot ROM.

2) Compliance testing and certification for standards for hardware security modules typically mandate that the TPM Mobile will test its functions before the results that depend on those functions may be returned.

2.5 Mandatory PCR Support

A TPM Mobile compliant to this profile SHALL support at least eight SHA256 PCR in a single bank defined in the following table. A TPM Mobile compliant to this profile MAY support additional higher-numbered PCR and/or additional banks of PCR.

A TPM Mobile compliant to this profile SHALL NOT support reset of any PCR defined in the following table, to avoid collisions with low-numbered PCR usage in the PC Client Platform TPM Profile [3].

Notes:

1) The mandatory PCR listed in the following table are a proper subset of the PC Client Platform TPM Profile [3].

2) The Mobile Common Profile is optimized for ease-of-implementation in constrained mobile devices that: (a) might not have any equivalent to the PC BIOS; (b) might not have any equivalents to TPM restart and TPM resume operations; and (c) might not support multiple localities (see section 2.6). These optimizations reduce the PCR range appropriate for such mobile devices.

Table 5 – TPM Mandatory PCR Support

PCR	Properties	Initial Value	Description
0	TPM_PT_PCR_EXTEND_L0	Defined by HW Platform	World-extendable PCR
1-7	TPM_PT_PCR_EXTEND_L0	0	World-extendable PCR

2.5.1 Recommended PCR Functions

A TPM Mobile compliant to this profile SHOULD support the recommended PCR functions defined in the following table.

Notes:

1) The recommended PCR functions listed in the following table are a proper subset of the PC Client Platform TPM Profile [3].

2) The recommended PCR functions listed in the following table are identical to the TCG EFI Platform Specification (for TPM Family 1.1 or 1.2) [6].

Table 6 – TPM Recommended PCR Functions

PCR	Function
PCR0	RECOMMENDED – Platform firmware code integrated into system board or SoC
PCR1	RECOMMENDED – Platform firmware data associated with code measured in PCR0
PCR2	RECOMMENDED – Platform firmware code (including drivers) added by VARs
PCR3	RECOMMENDED – Platform firmware data associated with code measured in PCR2
PCR4	RECOMMENDED – Pre-OS diagnostics and OS loader code
PCR5	RECOMMENDED – Pre-OS diagnostics and OS loader data associated with code measured into PCR4
PCR6	RECOMMENDED – Platform-specific OS and secure boot code
PCR7	RECOMMENDED – Platform-specific secure boot policy data associated with code measured in PCR6

2.6 Mandatory Locality Support

A TPM Mobile compliant to this profile SHALL support locality 0. A TPM Mobile compliant to this profile MAY support other localities.

2.7 Mandatory NV Storage Support

A TPM Mobile compliant to this profile SHALL support all TPM NV storage requirements defined as mandatory in the following table. A TPM Mobile compliant to this profile MAY support additional NV storage.

A TPM Mobile compliant to this profile SHALL include an implementation-defined amount of NV storage sufficient for pre-defined TPM internal data and TPM permanent resources, in addition to the minimum size for the NV area listed for applications below.

A TPM Mobile compliant to this profile SHALL also include NV storage for at least one permanent EK certificate (typically between 1,600 bytes and 4,096 bytes, depending on algorithm and key size), in addition to the the minimum size for the NV area for applications below and the storage of all other TPM internal data and TPM permanent resources in the paragraph above.

Notes:

1) The mandatory NV storage support requirements listed in the following table are a proper subset of the PC Client Platform TPM Profile [3].

2) The mandatory NV storage support requirements listed in the following table are lower than specified in the PC Client Platform TPM Profile [3] because of the typical memory and speed constraints for basic phones and feature phones.

Table 7 – TPM Mandatory NV Storage Support

NV Requirement	Minimum Value	Comments
Minimum size for NV area for use by applications	2,048 Bytes	This indicates the minimum amount of total NV space that can be used by applications in NV commands, in addition to the minimum requirements defined in the following rows of this table. This does not include NV storage for pre-defined TPM internal data, TPM permanent resources, or permanent EK certificate(s).
Minimum number of counter indices	4	Corresponds to the TPMA_NV_COUNTER bit.
Minimum number of reserved PCR-style indices	4	Corresponds to the TPMA_NV_EXTEND bit.
Minimum number of reserved bit fields	0	Corresponds to the TPMA_NV_BITS bit.
Minimum number of hybrid indices	0	Corresponds to the TPMA_NV_ORDERLY bit.
Minimum number of persistent objects	7	Corresponds to TPM_PT_HR_PERSISTENT_MIN.

2.8 Mandatory Resource Support

A TPM Mobile compliant with this profile SHALL support all resource requirements defined as mandatory in the following table. A TPM Mobile compliant to this profile MAY support additional resources.

Notes:

1) The mandatory Resource support requirements listed in the following table are a proper subset of the PC Client Platform TPM Profile [3].

2) The mandatory Resource support requirements listed in the following table are lower than those specified in the PC Client Platform TPM Profile [3] because of the typical memory and speed constraints for basic phones and feature phones.

Table 8 – TPM Mandatory Resource Support

Resource Type	Minimum
Active Sessions	16
Concurrent loaded sessions	3
Concurrent loaded objects	3

2.9 Mandatory Hierarchy Support

A TPM Mobile compliant with this profile SHALL support the Platform, Endorsement, Storage, and NULL hierarchies.

Note: The mandatory hierarchy support requirements for the TPM Mobile are identical to the PC Client Platform TPM Profile [3].

2.10 Mandatory Platform Interface Indication Support

A TPM Mobile compliant to this profile SHALL implement all mandatory platform interface indications listed in the following table. A TPM Mobile compliant with this profile MAY support other platform interface indications.

Notes:

1) The mandatory platform interface indications listed in the following table are a proper subset of the PC Client Platform TPM Profile [3].

2) The mandatory platform interface indications listed in the following table are fewer than specified in the PC Client Platform TPM Profile [3] because of the typical memory and speed constraints for basic phones and feature phones.

Table 9 – TPM Platform Interface Indication Support

Platform Signal	TPM Mobile	Description
_TPM_Init	REQUIRED	Means of indicating TPM Mobile initialization (but not whole system power-on)
_TPM_Hash_Start	OPTIONAL	Means of indicating hash sequence start and whether it is allowed between _TPM_Init and TPM_Startup.
_TPM_Hash_Data	OPTIONAL	Means of indicating hash sequence update and whether it is allowed between _TPM_Init and TPM_Startup.
_TPM_Hash_End	OPTIONAL	Means of indicating hash sequence end and whether it is allowed between _TPM_Init and TPM_Startup.