Trusted Platform Module Library



Vendor-Specific Values in Reference Code

Family “2.0”

Level 00 Revision 01.46

August 19, 2019

Contact: admin@trustedcomputinggroup.org

TCG Confidential

Copyright © TCG 2006-2019

**Licenses and Notices**

1. Copyright Licenses:

The TCG grants to the user of the other parts of this document the rights to reproduce, distribute, display, and perform the document for any purpose provided that this page is included in any copy of the document.

1. Disclaimers:

THE COPYRIGHT LICENSES SET FORTH ABOVE DO NOT REPRESENT ANY FORM OF LICENSE OR WAIVER, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, WITH RESPECT TO PATENT RIGHTS HELD BY TCG MEMBERS (OR OTHER THIRD PARTIES).

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ACCURACY, COMPLETENESS, OR NONINFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL OR SAMPLE.

Without limitation, TCG and its members and licensors disclaim all liability, including liability for infringement of any proprietary rights, relating to use of information in this document, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this document or any information herein.

CONTENTS

[1 Introduction 4](#_Toc17117600)

[2 Notation 5](#_Toc17117601)

[3 Processor Values 5](#_Toc17117602)

[4 Implemented Algorithms 6](#_Toc17117603)

[5 Key Size Selections 8](#_Toc17117604)

[6 Implemented Curves 9](#_Toc17117605)

[7 Implemented Commands 10](#_Toc17117606)

[8 Authenticated Timers 14](#_Toc17117607)

[9 PLATFORM 15](#_Toc17117608)

[10 Implementation Specific Values 16](#_Toc17117609)

**Tables**

[Table 1 — Defines for Processor Values 5](#_Toc17117615)

[Table 2 — Defines for Implemented Algorithms 6](#_Toc17117616)

[Table 3 — Defines for Key Size Constants 8](#_Toc17117617)

[Table 4 — Defines for Implemented Curves 9](#_Toc17117618)

[Table 5 — Defines for Implemented Commands 10](#_Toc17117619)

[Table 6 — Defines for ACT Values 14](#_Toc17117620)

[Table 6 — Defines for PLATFORM Values 15](#_Toc17117621)

[Table 7 — Defines for Implementation Values 16](#_Toc17117622)

# Introduction

This document contains tables that are used to define the properties of a TPM 2.0 implementation. A vendor would modify these tables according to their implementation choices. The chosen values should then be reflected in the header files used in compilation of the TPM code.

A set of software scripts are available that will automatically convert the values in the provided tables into appropriately formatted header files. For availability of these scripts, please contact TCG Administration ([admin@trustedcomputinggroup.org](mailto:admin@trustedcomputinggroup.org)).

This document is not a specification.

# Processor Values

These values are used to control generation of octet-swapping routines and the handling of bit fields structures that are sent on the TPM interface. The canonical octet ordering for the TPM input/output buffer is “big endian” (MSB0) with the most significant octet of any datum at the lowest address. The canonical bit numbering assign the bit number of 0 to the least significant bit of each

NOTE The setting for the exemplar is for the x86 family of processor.

Table 1 — Defines for Processor Values

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| BIG\_ENDIAN\_TPM | NO | set to YES or NO according to the processor |
| LITTLE\_ENDIAN\_TPM | !BIG\_ENDIAN\_TPM |  |
| MOST\_SIGNIFICANT\_BIT\_0 | NO | set to YES or NO according to the processor |
| LEAST\_SIGNIFICANT\_BIT\_0 | !MOST\_SIGNIFICANT\_BIT\_0 |  |
| AUTO\_ALIGN | NO | set to YES if the processor allows unaligned accesses |

# Implemented Algorithms

This table is used to indicate the algorithms that are implemented in a TPM. The selections in the Value column may be changed to reflect the implementation. The values shown are illustrative.

The "Implemented" column contains a "Y", "YES", or blank to indicate that the command is present in the implementation, an "N" or "NO" to indicate that the command is not implemented.

NOTE The leading and trailing “\_” characters are to avoid name space collisions with some crypto libraries.

Table 2 — Defines for Implemented Algorithms

| Algorithm Name | Implemented | Comments |
| --- | --- | --- |
| RSA | YES |  |
| SHA1 | YES |  |
| HMAC | YES | REQUIRED, do not change this value |
| TDES | NO |  |
| AES | YES |  |
| MGF1 | YES |  |
| XOR | YES |  |
| KEYEDHASH | YES | REQUIRED, do not change this value |
| SHA256 | YES |  |
| SHA384 | YES |  |
| SHA512 | NO |  |
| SM3\_256 | YES |  |
| SM4 | YES |  |
| RSASSA | YES | requires RSA |
| RSAES | YES | requires RSA |
| RSAPSS | YES | requires RSA |
| OAEP | YES | requires RSA |
| ECC | YES |  |
| ECDH | YES | requires ECC |
| ECDSA | YES | requires ECC |
| ECDAA | YES | requires ECC |
| SM2 | NO | requires ECC |
| CAMELLIA | YES |  |
| ECSCHNORR | YES | requires ECC |
| ECMQV | NO | requires ECC |
| SYMCIPHER | YES | REQUIRED, at least one symmetric algorithm shall be implemented |
| KDF1\_SP800\_56A | YES | requires ECC |
| KDF2 | NO |  |
| KDF1\_SP800\_108 | YES |  |
| CMAC | YES |  |
| CTR | YES |  |
| OFB | YES |  |
| CBC | YES |  |
| CFB | YES | REQUIRED, do not change this value |
| ECB | YES |  |

# Key Size Selections

The values in this table indicate the allowed key sizes for the implementation.

The Value column needs to have a list format (one or more values within “{}”).

Make sure that the Type column is properly labeled or strange values will be generated.

To prevent generation of the values for an algorithm, add any non-upper-case character to the start of the name (such as “-RSA\_KEY\_SIZES\_BITS” or “xRSA\_KEY\_SIZES\_BITS” or “//RSA\_KEY\_SIZES\_BITS”). Deleting the row will, of course, also work. Do not just leave an empty list of key sizes

ECC key sizes are determined by the curve selection in Table 4.

Table 3 — Defines for Key Size Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Value | Type | Comments |
| RSA\_KEY\_SIZES\_BITS | {1024, 2048} | Asymmetric | values for the reference implementation |
| TDES\_KEY\_SIZES\_BITS | {128, 192} | Block Cipher | values for the reference implementation |
| AES\_KEY\_SIZES\_BITS | {128, 256} | Block Cipher | values for the reference implementation |
| SM4\_KEY\_SIZES\_BITS | {128} | Block Cipher | not relevant in this implementation |
| CAMELLIA\_KEY\_SIZES\_BITS | {128, 256} | Block Cipher | not relevant for reference implementations |

# Implemented Curves

This table is used to indicate the curves that are implemented in a TPM. The selections in the Value column may be changed to reflect the implementation. The values shown are illustrative.

The “Implemented” column contains a “Y”, “YES”, or blank to indicate that the curve is present in the implementation, an “N” or “NO” to indicate that the curve is not implemented.

If ECC is not implemented, the contents of this table will not have an effect on the code generated.

NOTE The curve names are required to have a number of at least three digits in the last part of the name. ECC\_SRC\_P256 and ECC\_SRC\_P256A are both acceptable but ECC\_P256\_SRC is not.

Table 4 — Defines for Implemented Curves

| Curve Identifier | Implemented | Comments |
| --- | --- | --- |
| ECC\_NIST\_P192 | NO |  |
| ECC\_NIST\_P224 | NO |  |
| ECC\_NIST\_P256 | YES |  |
| ECC\_NIST\_P384 | YES |  |
| ECC\_NIST\_P521 | NO |  |
| ECC\_BN\_P256 | YES |  |
| ECC\_BN\_P638 | NO | This is needed to give 384-bit strength for ECDAA |
| ECC\_SM2\_P256 | YES |  |
| ECC\_TEST\_P192 | YES |  |

# Implemented Commands

This table is used to indicate which of the commands are implemented. In the reference implementation, this table determines which commands can be called and drives the generation of various command-dependent switch statements.

The “Implemented or Dependent” column contains a “Y”, “YES”, or blank to indicate that the command is present in the implementation; an “N” or “NO” to indicate that the command is not implemented; and an algorithm value if implementation of the command is dependent on a setting in Table 2

Table 5 — Defines for Implemented Commands

| Name | Implemented | Comments |
| --- | --- | --- |
| ActivateCredential | YES |  |
| Certify | Y |  |
| CertifyCreation | Y |  |
| ChangeEPS | Y |  |
| ChangePPS | Y |  |
| Clear | Y |  |
| ClearControl | Y |  |
| ClockRateAdjust | Y |  |
| ClockSet | Y |  |
| Commit | Y | Requires ECC |
| ContextLoad | Y | Context |
| ContextSave | Y | Context |
| Create | Y |  |
| CreatePrimary | Y |  |
| DictionaryAttackLockReset | Y |  |
| DictionaryAttackParameters | Y |  |
| Duplicate | Y |  |
| ECC\_Parameters | Y | Requires ECC |
| ECDH\_KeyGen | Y | Requires ECC |
| ECDH\_ZGen | Y | Requires ECC |
| EncryptDecrypt | Y |  |
| EventSequenceComplete | Y |  |
| EvictControl | Y |  |
| FieldUpgradeData | N |  |
| FieldUpgradeStart | N |  |
| FirmwareRead | N |  |
| FlushContext | Y | Context |
| GetCapability | Y |  |
| GetCommandAuditDigest | Y |  |
| GetRandom | Y |  |
| GetSessionAuditDigest | Y |  |
| GetTestResult | Y |  |
| GetTime | Y |  |
| Hash | Y |  |
| HashSequenceStart | Y |  |
| HierarchyChangeAuth | Y |  |
| HierarchyControl | Y |  |
| HMAC | Y | Will be disabled if CMAC is implemented |
| MAC | Y | Will be enabled if CMAC is available |
| HMAC\_Start | Y | Will be disabled if CMAC is available |
| MAC\_Start | Y | Will be disabled if CMA is available |
| Import | Y |  |
| IncrementalSelfTest | Y |  |
| Load | Y |  |
| LoadExternal | Y |  |
| MakeCredential | Y |  |
| NV\_Certify | Y |  |
| NV\_ChangeAuth | Y |  |
| NV\_DefineSpace | Y |  |
| NV\_Extend | Y |  |
| NV\_GlobalWriteLock | Y |  |
| NV\_Increment | Y |  |
| NV\_Read | Y |  |
| NV\_ReadLock | Y |  |
| NV\_ReadPublic | Y | NV |
| NV\_SetBits | Y |  |
| NV\_UndefineSpace | Y |  |
| NV\_UndefineSpaceSpecial | Y |  |
| NV\_Write | Y |  |
| NV\_WriteLock | Y |  |
| ObjectChangeAuth | Y |  |
| PCR\_Allocate | Y |  |
| PCR\_Event | Y | PCR |
| PCR\_Extend | Y |  |
| PCR\_Read | Y | PCR |
| PCR\_Reset | Y | PCR |
| PCR\_SetAuthPolicy | Y |  |
| PCR\_SetAuthValue | Y |  |
| PolicyAuthorize | Y | Policy |
| PolicyAuthValue | Y | Policy |
| PolicyCommandCode | Y | Policy |
| PolicyCounterTimer | Y | Policy |
| PolicyCpHash | Y | Policy |
| PolicyDuplicationSelect | Y | Policy |
| PolicyGetDigest | Y | Policy |
| PolicyLocality | Y | Policy |
| PolicyNameHash | Y | Policy |
| PolicyNV | Y | Policy |
| PolicyOR | Y | Policy |
| PolicyPassword | Y | Policy |
| PolicyPCR | Y | Policy |
| PolicyPhysicalPresence | Y | Policy |
| PolicyRestart | Y |  |
| PolicySecret | Y | Policy |
| PolicySigned | Y | Policy |
| PolicyTicket | Y | Policy |
| PP\_Commands | Y |  |
| Quote | Y |  |
| ReadClock | Y |  |
| ReadPublic | Y |  |
| Rewrap | Y |  |
| RSA\_Decrypt | Y | Requires RSA |
| RSA\_Encrypt | Y | Requires RSA |
| SelfTest | Y |  |
| SequenceComplete | Y |  |
| SequenceUpdate | Y |  |
| SetAlgorithmSet | Y |  |
| SetCommandCodeAuditStatus | Y |  |
| SetPrimaryPolicy | Y |  |
| Shutdown | Y |  |
| Sign | Y |  |
| StartAuthSession | Y |  |
| Startup | Y |  |
| StirRandom | Y |  |
| TestParms | Y |  |
| Unseal | Y |  |
| VerifySignature | Y |  |
| ZGen\_2Phase | Y | Requires ECC |
| EC\_Ephemeral | Y | Requires ECC |
| PolicyNvWritten | Y |  |
| PolicyTemplate | Y |  |
| CreateLoaded | Y |  |
| PolicyAuthorizeNV | Y |  |
| EncryptDecrypt2 | Y |  |
| AC\_GetCapability | Y |  |
| AC\_Send | Y |  |
| Policy\_AC\_SendSelect | Y |  |
| Vendor\_TCG\_Test | Y |  |
| CertifyX509 | Y |  |
| ACT\_SetTimeout | Y |  |
| ECC\_Encrypt | Y |  |
| ECC\_Decrypt | Y |  |

# Authenticated Timers

This table indicates which of the ACT are implemented. It is only necessary to list the implemented values

Table 6 — Defines for Implemented ACT

| Name | Implemented | Comments |
| --- | --- | --- |
| RH\_ACT\_0 | YES | assumed to be the AWT |
| RH\_ACT\_1 | NO | iisted to make sure macros are correct |
| RH\_ACT\_A | YES | currently, this is just used for testing to make sure that the macros generate numbers correctly |

# PLATFORM

These values are readable with TPM2\_GetCapability(). They are the TPM\_PT\_PS\_xxx values.

The vendor should update these values for the applicable platform specification. In this example, the values refer to the TPM specification values.

These values are not listed as “Definition of” because the values in this document are macro replacement values and not constants.

Table 6 — Defines for PLATFORM Values

| Name | Value | Comments |
| --- | --- | --- |
| PLATFORM\_FAMILY | TPM\_SPEC\_FAMILY |  |
| PLATFORM\_LEVEL | TPM\_SPEC\_LEVEL |  |
| PLATFORM\_VERSION | TPM\_SPEC\_VERSION |  |
| PLATFORM\_YEAR | TPM\_SPEC\_YEAR |  |
| PLATFORM\_DAY\_OF\_YEAR | TPM\_SPEC\_DAY\_OF\_YEAR |  |

# Implementation Specific Values

This table contains a collection of values used in various parts of the reference code. The values shown are illustrative.

Table 7 — Defines for Implementation Values

| Name | Value | Description |
| --- | --- | --- |
| FIELD\_UPGRADE\_IMPLEMENTED | NO | temporary define |
|  |  |  |
|  |  |  |
|  |  |  |
| HASH\_LIB | Ossl | Selection of the library that provides the basic hashing functions. |
| SYM\_LIB | Ossl | Selection of the library that provides the low-level symmetric cryptography. Choices are determined by the vendor (See LibSupport.h for implications). |
| MATH\_LIB | Ossl | Selection of the library that provides the big number math including ECC. Choices are determined by the vendor (See LibSupport.h for implications). |
|  |  |  |
| IMPLEMENTATION\_PCR | 24 | the number of PCR in the TPM |
| PCR\_SELECT\_MAX | ((IMPLEMENTATION\_PCR+7)/8) |  |
| PLATFORM\_PCR | 24 | the number of PCR required by the relevant platform specification |
| PCR\_SELECT\_MIN | ((PLATFORM\_PCR + 7) / 8) |  |
| DRTM\_PCR | 17 | the D-RTM PCR  NOTE This value is not defined when the TPM does not implement D-RTM |
| HCRTM\_PCR | 0 | the PCR that will receive the H-CRTM value at TPM2\_Startup. This value should not be changed. |
| NUM\_LOCALITIES | 5 | the number of localities supported by the TPM  This is expected to be either 5 for a PC, or 1 for just about everything else. |
| MAX\_HANDLE\_NUM | 3 | the maximum number of handles in the handle area  This should be produced by the Part 3 parser but is here for now. |
| MAX\_ACTIVE\_SESSIONS | 64 | the number of simultaneously active sessions that are supported by the TPM implementation |
| CONTEXT\_SLOT | UINT16 | the type of an entry in the array of saved contexts |
|  |  |  |
| MAX\_LOADED\_SESSIONS | 3 | the number of sessions that the TPM may have in memory |
| MAX\_SESSION\_NUM | 3 | this is the current maximum value |
| MAX\_LOADED\_OBJECTS | 3 | the number of simultaneously loaded objects that are supported by the TPM; this number does not include the objects that may be placed in NV memory by TPM2\_EvictControl(). |
| MIN\_EVICT\_OBJECTS | 2 | the minimum number of evict objects supported by the TPM |
| NUM\_POLICY\_PCR\_GROUP | 1 | number of PCR groups that have individual policies |
| NUM\_AUTHVALUE\_PCR\_GROUP | 1 | number of PCR groups that have individual authorization values |
| MAX\_CONTEXT\_SIZE | 1264 |  |
| MAX\_DIGEST\_BUFFER | 1024 |  |
| MAX\_NV\_INDEX\_SIZE | 2048 | maximum data size allowed in an NV Index |
| MAX\_NV\_BUFFER\_SIZE | 1024 | maximum data size in one NV read or write command |
| MAX\_CAP\_BUFFER | 1024 | maximum size of a capability buffer |
| NV\_MEMORY\_SIZE | 16384 | size of NV memory in octets |
| MIN\_COUNTER\_INDICES | 8 | the TPM will not allocate a non-counter index if it would prevent allocation of this number of indices. |
| NUM\_STATIC\_PCR | 16 |  |
| MAX\_ALG\_LIST\_SIZE | 64 | number of algorithms that can be in a list |
| PRIMARY\_SEED\_SIZE | 32 | size of the Primary Seed in octets |
| CONTEXT\_ENCRYPT\_ALGORITHM | AES | context encryption algorithm  Just use the root so that the macros in GpMacros.h will work correctly. |
| NV\_CLOCK\_UPDATE\_INTERVAL | 12 | the update interval expressed as a power of 2 seconds  A value of 12 is 4,096 seconds (~68 minutes). |
| NUM\_POLICY\_PCR | 1 | number of PCR groups that allow policy/auth |
| MAX\_COMMAND\_SIZE | 4096 | maximum size of a command |
| MAX\_RESPONSE\_SIZE | 4096 | maximum size of a response |
| ORDERLY\_BITS | 8 | number between 1 and 32 inclusive |
| MAX\_SYM\_DATA | 128 | the maximum number of octets that may be in a sealed blob; 128 is the minimum allowed value |
| MAX\_RNG\_ENTROPY\_SIZE | 64 |  |
| RAM\_INDEX\_SPACE | 512 | Number of bytes used for the RAM index space. If this is not large enough, it might not be possible to allocate orderly indices. |
| RSA\_DEFAULT\_PUBLIC\_EXPONENT | 0x00010001 | 216 + 1 |
| ENABLE\_PCR\_NO\_INCREMENT | YES | indicates if the TPM\_PT\_PCR\_NO\_INCREMENT group is implemented |
| CRT\_FORMAT\_RSA | YES |  |
| VENDOR\_COMMAND\_COUNT | 0 |  |
| MAX\_VENDOR\_BUFFER\_SIZE | 1024 | Maximum size of the vendor-specific buffer |
| MAX\_DERIVATION\_BITS | 8192 | “L’ value for a derivation. This is the  maximum number of bits allowed from an instantiation of a KDF-DRBG. This is size is OK because RSA keys are never derived keys |
| RSA\_MAX\_PRIME | (MAX\_RSA\_KEY\_BYTES/2) |  |
| RSA\_PRIVATE\_SIZE | (RSA\_MAX\_PRIME \* 5) |  |
| SIZE\_OF\_X509\_SERIAL\_NUMBER | 20 |  |
| PRIVATE\_VENDOR\_SPECIFIC\_BYTES | RSA\_PRIVATE\_SIZE | This is a vendor-specific value so it is in this vendor-speific table. When this is used, RSA\_PRIVATE\_SIZE will have been defined |