

## Errata for TCG Trusted Platform Module Library

Family "2.0"  
Level 00 Revision 01.59  
November 8, 2019

---

Version 1.4  
January 9, 2023

Contact: [admin@trustedcomputinggroup.org](mailto:admin@trustedcomputinggroup.org)

PUBLISHED

## DISCLAIMERS, NOTICES, AND LICENSE TERMS

THIS ERRATA IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, 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 specification or any information herein.

This document is copyrighted by Trusted Computing Group (TCG), and no license, express or implied, is granted herein other than as follows: You may not copy or reproduce the document or distribute it to others without written permission from TCG, except that you may freely do so for the purposes of (a) examining or implementing TCG specifications or (b) developing, testing, or promoting information technology standards and best practices, so long as you distribute the document with these disclaimers, notices, and license terms.

Contact the Trusted Computing Group at [admin@trustedcomputinggroup.org](mailto:admin@trustedcomputinggroup.org) for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

## CHANGE HISTORY

VERSION	DATE	DESCRIPTION
1.0	December 18, 2019	Added <ul style="list-style-type: none"><li>• 2.1 TPM_SPEC Date Constants [specification text, code]</li><li>• 2.2 Non-orderly Shutdown – failedTries [code]</li></ul>
1.1	June 18, 2020	Added <ul style="list-style-type: none"><li>• 2.3 ACT preserveSignaled [specification text, code]</li></ul>
1.2	November 10, 2020	Added <ul style="list-style-type: none"><li>• 2.4 RSAES_Decode - padding [code]</li></ul>
1.3	March 11, 2022	Added <ul style="list-style-type: none"><li>• 2.5 TPM_EO – two's complement [code]</li><li>• 3.1.1 TPM2_NV_WriteLock, TPM2_NV_ReadLock [specification text]</li></ul>
1.4	January 9, 2023	Added <ul style="list-style-type: none"><li>• 2.6.1 CryptParameterEncryption/ Decryption [code]</li><li>• 2.6.2 TPM2_PolicyAuthorize [code]</li><li>• 2.6.3 CryptGenerateKeyDes [code]</li></ul>

## CONTENTS

DISCLAIMERS, NOTICES, AND LICENSE TERMS .....	1
CHANGE HISTORY .....	2
1 Introduction .....	4
2 Errata .....	5
2.1 TPM_SPEC Date Constants [specification text, code].....	5
2.2 Non-orderly Shutdown - <i>failedTries</i> [code].....	5
2.3 ACT <i>preserveSignaled</i> [specification text, code].....	5
2.4 RSAES_Decode - padding [code] .....	6
2.5 TPM_EO – two’s complement [code] .....	6
2.6 Size Checks .....	7
2.6.1 CryptParameterEncryption/Decryption [code].....	7
2.6.2 TPM2_PolicyAuthorize [code] .....	7
2.6.3 CryptGenerateKeyDes [code] .....	7
3 Clarifications .....	8
3.1 Error Codes.....	8
3.1.1 TPM2_NV_WriteLock, TPM2_NV_ReadLock [specification text].....	8

## 1 Introduction

This document describes errata and clarifications for the TCG Trusted Platform Module Library Family “2.0” Level 00 Revision 01.59 as published. The information in this document is likely – but not certain – to be incorporated into a future version of the specification. Suggested fixes proposed in this document may be modified before being published in a later TCG Specification. Therefore, the contents of this document are not normative and only become normative when included in an updated version of the published specification. Note that since the errata in this document are non-normative, the patent licensing rights granted by Section 16.4 of the Bylaws do not apply.

The heading of each errata in section 2 indicates whether the errata affects the specification text or the reference code implementation. This is indicated by the word “specification text” or “code” in square brackets (“[]”).

## 2 Errata

### 2.1 TPM\_SPEC Date Constants [specification text, code]

Table 6 in Part 2, 6.1 TPM\_SPEC (Specification Version Values) should be replaced with:

Table 6 — Definition of (UINT32) TPM\_SPEC Constants <>

Name	Value	Comments
TPM_SPEC_FAMILY	0x322E3000	ASCII "2.0" with null terminator
TPM_SPEC_LEVEL	00	the level number for the specification
TPM_SPEC_VERSION	159	the version number of the spec (001.59 * 100)
TPM_SPEC_YEAR	2023	the year of the version
TPM_SPEC_DAY_OF_YEAR	9	the day of the year (January 9)

That is, the spec date fields TPM\_SPEC\_YEAR and TPM\_SPEC\_DAY\_OF\_YEAR should be set to the date of this Errata document.

### 2.2 Non-orderly Shutdown - *failedTries* [code]

The following text in Part 1, 19.8.6 Non-orderly Shutdown describes the reference code implementation of TPM2\_Startup() after a non-orderly Shutdown:

An alternative implementation sets an NV flag indicating that access to a DA protected object occurred during this boot cycle. After a non-orderly restart, if the flag is set, the TPM increments *failedTries* and clears the flag. If the flag is clear, there is no need to increment *failedTries*.

EXAMPLE This handles the case where a platform repeatedly does a non-orderly shutdown, possibly due to a low battery. Without the flag, *failedTries* would increment on each reboot and the TPM would go into lockout.

The reference code does not correctly implement the behavior described above if a DA protected object is accessed after a TPM2\_Shutdown(). In this case, the NV flag (indicating that access to a DA protected object occurred during this boot cycle) is not set correctly. When a power loss happens, *failedTries* is not incremented on the next TPM2\_Startup(). The reference code should be fixed.

The check and increment of *failedTries* on TPM2\_Startup() ensures that a failed authorization attempt is recorded by the TPM (e.g. because NV memory is unavailable).

### 2.3 ACT *preserveSignaled* [specification text, code]

The ACT *preserveSignaled* attribute is incorrectly described in the Library Spec Part 2 and 3, and is incorrectly implemented in the reference code in Part 4, 7.8.3.2 ActStartup(). The reference code always returns zero for the *preserveSignaled* attribute.

In Part 2, 8.12 TPMA\_ACT, the following text should be added to the description of the ACT attribute structure.

The *preserveSignaled* action over a power cycle is:

- Cold (with power loss between Shutdown and Startup) TPM Reset, TPM Restart, TPM Resume
  - *preserveSignaled* is set to CLEAR
- Warm (no power loss between Shutdown and Startup) TPM Reset, TPM Restart, TPM Resume
  - *preserveSignaled* holds the state of *signaled* before the power cycle

NOTE 1: *preserveSignaled* allows startup software to determine if the startup cycle was likely initiated by an ACT event. If power was lost, it doesn't care.

In Part 2, 8.12 TPMA\_ACT, Table 40 should be replaced with the following table:

**Table 40 — Definition of (UINT32) TPMA\_ACT Bits**

Bit	Name	Definition
0	signaled	<b>SET (1):</b> The ACT has signaled <b>CLEAR (0):</b> The ACT has not signaled
1	preserveSignaled	Preserves the state of <i>signaled</i> , depending on the power cycle
31:2	Reserved	shall be zero

In Part 3, 9.3 TPM2\_Startup, in the general description of the command actions on TPM Reset and on TPM Restart, the following bullet point should be changed.

From:

- For each ACT the timeout is reset to zero, the *signaled* attribute is set to CLEAR (if *preserveSignaled* is CLEAR), and the *authPolicy* is set to the Empty Buffer and its hashAlg is set to TPM\_ALG\_NULL.

To:

- For each ACT the timeout is reset to zero, the *signaled* attribute is set to CLEAR, its *authPolicy* is set to the Empty Buffer, and its hashAlg is set to TPM\_ALG\_NULL.

That is, the condition in brackets “(if *preserveSignaled* is CLEAR)” should be removed.

In Part 3, 32.2 TPM2\_ACT\_SetTimeout, in the general description, the following sentence should be added:

When this command is successful, *preserveSignaled* will be CLEAR.

The reference code should be fixed following the (above) corrections in Part 2 and 3.

## 2.4 RSAES\_Decode - padding [code]

The RSAES\_Decode() function in Part 4, 10.2.17.4.12, which performs the decoding for RSAES-PKCS1-V1\_5-DECRYPT as defined in PKCS#1V2.1, behaves incorrectly if the padding string (PS) is 7 octets.

According to PKCS#1V2.1, the length of the padding string (PS) must be at least 8 octets. However, the RSAES\_Decode() reference code function permits a padding string of 7 bytes. The RSAES\_Decode() function should be corrected to reject a padding string that is less than 8 bytes.

## 2.5 TPM\_EO – two’s complement [code]

According to Part 3, 23.9 TPM2\_PolicyNV and 23.10 TPM2\_PolicyCounterTimer,

“The signed arithmetic operations are performed using twos-complement.”

For two negative values, the reference code (in Part 4, 9.11.2.2 SignedCompareB()) implements the signed arithmetic operations using sign-magnitude. The reference code should be fixed to match the description in Part 3.

## 2.6 Size Checks

### 2.6.1 CryptParameterEncryption/Decryption [code]

The functions CryptParameterEncryption() and CryptParameterDecryption() in the reference code in Part 4, 10.2.6.6.5 and 10.2.6.6.6 do not correctly check the size of the parameter buffer to be encrypted or decrypted. To fix the issue, the functions should be corrected to check that the parameter buffer (a TPM2B type field) is at least 2 bytes in length and should use the function UINT16\_Unmarshal() to read the size of the buffer instead of BYTE\_ARRAY\_TO\_UINT16().

The fixed CryptParameterDecryption() function will return TPM\_RC\_INSUFFICIENT if the input buffer does not contain enough data to read the UINT16 size field.

The fixed CryptParameterEncryption() function will enter failure mode and return TPM\_RC\_FAILURE if the internal response buffer does not contain enough data for the UINT16 size field.

### 2.6.2 TPM2\_PolicyAuthorize [code]

TPM2\_PolicyAuthorize() in the reference code in Part 3, 23.16 does not correctly check the size of the *keySign* parameter. To fix the issue, the TPM will check that *keySign* (a TPM2B type field) is at least 2 bytes in length or otherwise return TPM\_RC\_INSUFFICIENT.

### 2.6.3 CryptGenerateKeyDes [code]

The function CryptGenerateKeyDes() in the reference code in Part 4, 10.2.9.2.3 does not correctly check the symmetric key size provided in the *sensitive* parameter. To fix the issue, the function will check that the size of the requested TDES key is a multiple of 8 bytes or otherwise the TPM will return TPM\_RC\_SYMMETRIC.

## 3 Clarifications

### 3.1 Error Codes

#### 3.1.1 TPM2\_NV\_WriteLock, TPM2\_NV\_ReadLock [specification text]

There is an ambiguity in the command descriptions of TPM2\_NV\_WriteLock and TPM2\_NV\_ReadLock whether the TPM shall return success or an (authorization) error code if TPM2\_NV\_WriteLock or TPM2\_NV\_ReadLock is executed with improper authorization for an NV index that is already write or read-locked.

The description in Part 3, 31.11 TPM2\_NV\_WriteLock should be interpreted as if

TPM2\_NV\_WriteLock may either return TPM\_RC\_SUCCESS or TPM\_RC\_NV\_AUTHORIZATION if TPMA\_NV\_WRITELOCKED for the NV Index is already SET, and proper write authorization (as determined by TPMA\_NV\_PPWRITE, TPMA\_NV\_OWNERWRITE, TPMA\_NV\_AUTHWRITE, and the *authPolicy* of the NV Index) is not provided.

The description in Part 3, 31.14 TPM2\_NV\_ReadLock should be interpreted as if

TPM2\_NV\_ReadLock may either return TPM\_RC\_SUCCESS or TPM\_RC\_NV\_AUTHORIZATION if TPMA\_NV\_READLOCKED for the NV Index is already SET, and proper read authorization (as determined by TPMA\_NV\_PPREAD, TPMA\_NV\_OWNERREAD, TPMA\_NV\_AUTHREAD, and the *authPolicy* of the NV Index) is not provided.