

# ERRATA

## ERRATA

Errata Version 1.2  
February 16, 2015

## FOR

## TCG Trusted Platform Module Library

Specification Version 2.0  
Revision 1.16  
October 30, 2014

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

**TCG Published**

Copyright © TCG 2015

## 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 and its members and licensors disclaim 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 [www.trustedcomputinggroup.org](http://www.trustedcomputinggroup.org) for information on specification licensing through membership agreements.

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

## CONTENTS

1. Introduction .....	1
2. Errata .....	1
2.1 Policy with trial session .....	1
2.2 Encryption with trial session.....	1
2.3 CFB Mode Parameter Encryption .....	1
2.4 Authorization sessions .....	1
2.5 TPM2_Quote.....	1
2.6 lockoutAuth behavior .....	1
2.7 Sign/decrypt attribute encoding .....	2
2.8 TPM2_EncryptDecrypt.....	2
2.9 XOR Obfuscation .....	2
2.10 TPM2_NV_UndefineSpaceSpecial.....	3
2.11 Error Codes.....	3
2.11.1 Introduction .....	3
2.11.2 TPM2_HMAC, TPM2_HMAC_Start.....	3
2.11.3 TPM2_PolicySigned .....	3
2.11.4 TPM2_NV_DefineSpace.....	3
2.11.5 TPM2_NV_Read, TPM2_NV_Write, TPM2_NV_Certify .....	3



## 1. Introduction

This document describes errata and clarifications for the TCG Trusted Platform Module Library Version 2.0 Revision 1.16 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.

## 2. Errata

### 2.1 Policy with trial session

In Part 3, section 23.4 and 23.9, the general description of the commands TPM2\_PolicySecret and TPM2\_PolicyNV specifically say that the authorization value is not checked if a trial session is being used. The exemplar code actually does do this check. As a result, if the authorization value given is not correct, the result may be either a success or an authorization failure (with accompanied increase in the dictionary attack counter). In a future version of the specification, this may be reconciled. It should be noted that if policies are calculated outside the TPM, no authorization check is needed.

### 2.2 Encryption with trial session

In the current specification, Part 1 Table 7, it indicates that you can use encryption and decryption attributes with a trial session. In fact you cannot because a trial session handle can never be used in the authorization area.

### 2.3 CFB Mode Parameter Encryption

Equation (34) in Part 1, section 21.3, CFB Mode Parameter Encryption needs to have *sessionValue* instead of *sessionKey* in the equation and the following description. The exemplar code uses *sessionValue* which is correct.

### 2.4 Authorization sessions

In Part 1, section 19.6.17 Authorization Session Termination, the current specification says

- when the TPM executes TPM2\_Startup(TPM\_SU\_CLEAR), all authorization sessions are terminated; and
- when the TPM executes TPM2\_Startup(TPM\_SU\_STATE), authorization sessions in TPM memory will be terminated but sessions stored off the TPM will remain active.

This is incorrect. It should state

- on TPM Reset, all authorization sessions are terminated; and
- on TPM Resume or TPM Restart, authorization sessions in TPM memory will be terminated but sessions context saved off the TPM will remain active.

### 2.5 TPM2\_Quote

If no key is provided in the *signHandle* parameter then this command can either return an error or (if the signing scheme is present) a TPMS\_ATTEST structure using the hash algorithm provided in the signing scheme.

Currently the reference code returns an error while the text in Part 3, section 18.4 TPM2\_Quote indicates that the TPMS\_ATTEST structure is returned.

### 2.6 lockoutAuth behavior

In Part 1, section 19.11.5 Authorization Failures Involving *lockoutAuth*, it indicates that if *lockoutRecovery* is set to zero, then the TPM will not allow the usage of *lockoutAuth* after an authorization failure until the

next TPM Reset. This disagrees with Part 3, section 25.3 TPM2\_DictionaryAttackParameters, which indicates that a TPM Reset or TPM Restart is required.

In the reference implementation the use of *lockoutAuth* is enabled on any TPM2\_Startup() if *lockoutRecovery* is zero. The behavior in the reference code is correct.

## 2.7 Sign/decrypt attribute encoding

According to the current definition in Part 1, Table 24, a SymCipher object can be used to encrypt and decrypt data if the *decrypt* attribute is SET. This is incorrect. The current Table 24 is only valid for KeyedHash and asymmetric objects.

For a SymCipher object the *sign* attribute encodes encryption and the *decrypt* attribute encodes decryption. Thus, a SymCipher object can only be used for encryption if the *sign* attribute is SET. To allow encryption and decryption, both the *sign* and *decrypt* attribute need to be SET. This allows SymCipher objects to differentiate between encryption and decryption.

This issue affects the command TPM2\_EncryptDecrypt() and the table in Part 2 that defines the allowed scheme values for SymCipher object .

## 2.8 TPM2\_EncryptDecrypt

The scheme selection in TPM2\_EncryptDecrypt() is specified incorrectly in Part 3 section 15.2, since it doesn't work like the other scheme selections. The correct behavior is specified below:

- 1) *keyHandle* is not allowed to reference a restricted key (TPM\_RC\_ATTRIBUTES).
- 2) If the key has a mode that is not TPM\_ALG\_NULL, then the caller cannot override it. The caller must either pick the same mode or TPM\_ALG\_NULL (TPM\_RC\_MODE).
- 3) If both the caller and the key have a mode selection of TPM\_ALG\_NULL, then it is an error (TPM\_RC\_MODE).

To allow these selections, Table 126 in Part 2 should be replaced with:

**Table 1 — Definition of TPMU\_SYM\_MODE Union**

Parameter	Type	Selector	Description
!ALG.S	TPMI_ALG_SYM_MODE+	TPM_ALG_!ALG.S	
sym	TPMI_ALG_SYM_MODE+		when selector may be any of the symmetric block ciphers
xor		TPM_ALG_XOR	no mode selector
null		TPM_ALG_NULL	no mode selector

## 2.9 XOR Obfuscation

When a KeyedHash Object is used for encryption, the creator of the key has the option of limiting the use of the key to specific schemes or of deferring the choice until the object is used. The current scheme definitions do not currently allow this selection.

To allow the intended functionality, Table 139 in Part 2 should be replaced with:

**Table 2 — Definition of TPMS\_SCHEME\_XOR Structure**

Parameter	Type	Description
hashAlg	TPMI_ALG_HASH+	the hash algorithm used to digest the message
kdf	TPMI_ALG_KDF+	the key derivation function

Currently, no command supports direct XOR encryption/decryption.

## 2.10 TPM2\_NV\_UndefineSpaceSpecial

In the reference implementation the TPM enters Failure Mode if the command TPM2\_NV\_UndefineSpaceSpecial() is executed with a policy that contains TPM2\_PolicyAuthValue(). When the response HMAC is computed the code tries to access the authorization value for an NV index that was deleted during the command. This causes the TPM simulator to assert.

In a future version of the Library specification the TPM will use EmptyAuth for the computation of the response HMAC if the *authValue* for an entity was deleted during the command.

Only platform manufacturers are affected by this issue. It is recommended that platform manufacturer do not include TPM2\_PolicyAuthValue() in the authorization policy used to authorize TPM2\_NV\_UndefineSpaceSpecial().

## 2.11 Error Codes

### 2.11.1 Introduction

The following section resolves ambiguities with regards to errors codes where the specification text and the reference code specify something different.

### 2.11.2 TPM2\_HMAC, TPM2\_HMAC\_Start

In Part 3, section 15.4 TPM2\_HMAC and 17.2 TPM2\_HMAC\_Start, the general description states that the TPM shall return TPM\_RC\_ATTRIBUTES if the key referenced by handle is not a signing key. The correct error code is TPM\_RC\_KEY (same as for all signing commands including attestation commands where the key is not a signing key). The reference code implements the error codes correctly.

### 2.11.3 TPM2\_PolicySigned

The Error Return Code Table in Part 3, section 23.3 TPM2\_PolicySigned indicates that TPM\_RC\_KEY is returned if authObject is not a signing scheme. This is incorrect. The *sign* attribute is not required to be SET for authObject in TPM2\_PolicySigned. The reference code is implemented correctly.

### 2.11.4 TPM2\_NV\_DefineSpace

The current description in Part 3, section 31.3 TPM2\_NV\_DefineSpace indicates that if TPMA\_NV\_EXTEND is SET, then *publicInfo*→*dataSize* shall match the digest size of the *publicInfo.nameAlg* or the TPM shall return TPM\_RC\_SIZE. However, the reference code returns TPM\_RC\_ATTRIBUTES. The correct response for this error is TPM\_RC\_SIZE as described in the specification text.

### 2.11.5 TPM2\_NV\_Read, TPM2\_NV\_Write, TPM2\_NV\_Certify

According to the command description in Part 3, TPM2\_NV\_Read, TPM2\_NV\_Write and TPM2\_NV\_Certify should return the error code TPM\_RC\_NV\_RANGE when the range defined by the *size* and *offset* parameter is outside the range of the referenced NV Index.

However, TPM\_RC\_NV\_RANGE is not allowed to have a response code modifier that would provide additional information about the type of error. In the interest of finer differentiation, TPM\_RC\_VALUE should be returned if the failure is caused by the *size* parameter or the *offset* parameter and TPM\_RC\_NV\_RANGE should be used to indicate a failure caused by the combination of *size* and *offset*.