TRUSTED® COMPUTING GROUP

SPECIFICATION

TPM 2.0 Authenticated Countdown Timer (ACT) Command

Version 1.0
Revision 3
September 6, 2019

Contact: admin@trustedcomputinggroup.org

Work in Progress

This document is an intermediate draft for comment only and is subject to change without notice. Readers should not design products based on this document.

PUBLIC REVIEW

DISCLAIMERS, NOTICES, AND LICENSE TERMS

CHANGE HISTORY

REVISION	DATE	DESCRIPTION
Version 1.00 Revision 1	September 2, 2019	• Moved to TCG specification template
Version 1.00 Revision 2	September 5, 2019	Addressed David and Ken's feedback: • 2.1.2 Clarified that each ACT has its own authPolicy • 3.4 Removed the modification from TPMI_RH_HIERARCHY_AUTH • 3.5.2 Added TPMI_RH_HIERARCHY_POLICY • 3.6.1 Fixed type of timeout parameter in TPMS_ACT_DATA • 4.1.1 Clarified ACT actions for the combinations ACT and startTimeout being zero, or non-zero • 4.3 Changed TPM2_SetPrimaryPolicy() to use TPMI_RH_HIERARCHY_POLICY instead of TPMI_RH_HIERARCHY_AUTH • 4.3.1 Clarified the enable flag associated with ACT authValue • 4.4 Removed the modification TPM2_HierarchyChangeAuth • Fixed trivial changes
Version 1.00 Revision 3	September 6, 2019	Edits during TPMWG Call 9/15: • 2.1.2 Clarified that ACT authPolicy is not enabled/disabled by phEnable • 4.1.1 Updated TPM2_ACT_SetTimeout command description • 5.1 Added info to be defined by platform spec

CONTENTS

1 Scope

This specification describes a new TPM 2.0 Library Spec command: TPM2_ACT_SetTimeout. The description is based on text only, no reference code is provided as part of this specification.

1.1 Purpose

The purpose of this specification is to give a preview of the new Authenticated Countdown Timer (ACT) command before the command (including reference code) will be integrated into a future TPM 2.0 Library Specification[1]. The sections 3 to 6 of this specification describe the additions and modifications to Part 1-3 of the Library Spec, which are necessary to support the ACT functionality. The benefit of documenting the ACT command in this form in a separate specification is to facilitate review of the new command.

1.2 Change Highlighting

The sections 3 and 4 of this specification use green font to highlight changes to tables in Part 2, and changes to command definitions in Part 3 of the Library Spec. This highlighting is specific to this specification and should help the reader to easier spot the change. When an entry is added to a table, the row before and after the inserted row (in green font) are shown. When text is added to an existing command definition, the added text is shown in green font. The changes are applied on Revision 1.55 of the Library Spec.

2 ACT Command Extension to Library Spec Part 1

2.1 Authenticated Countdown Timer (ACT)

2.1.1 Introduction

The functionality and commands described in this clause enable the TPM to manage multiple authenticated countdown timers (ACT).

2.1.2 Description

An ACT is a 32-bit counter that, when not already zero, will decrement by one each second that the TPM is powered.

The countdown timers are used to trigger events on a platform when they count down to zero, at which point they are said to timeout or expire. TPM2_ACT_SetTimeout() is used to set an ACT to a non-zero value and begin the timeout. On TPM Reset or TPM Restart, all ACT timeouts are set to zero with no side effects (no event triggered). ACT timeouts are preserved across TPM Resume.

The ACT timeouts are saved by TPM2_Shutdown(STATE). On TPM2_Startup(STATE), if the TPM shutdown was orderly, the saved ACT values are restored and the ACT resumes counting. If an ACT *startTimeout* has been written (TPM2_ACT_SetTimeout()) since the last TPM2_Startup(), then the current timeout of the ACT is saved by TPM2_Shutdown(STATE); otherwise, the saved value is one half of the current ACT timeout. If a TPM2_ACT_SetTimeout() occurs after the TPM2_Shutdown(), then the TPM state is no longer orderly, and a subsequent TPM2_Startup(STATE) will fail.

An ACT has an *authValue* and an *authPolicy*. The *authValue* is the same as the current *platformAuth* and can only be used if *phEnable* is SET. The *authPolicy* is ACT-specific and is neither enabled nor disabled by *phEnable*. Each ACT has its own *authPolicy*.

NOTE 1 A system might continue to operate after a TPM2_Shutdown(STATE). Therefore, saving half the timeout prevents an attacker from continually extending the timeout by doing TPM2_Shutdown(STATE) immediately after TPM2_Startup(STATE), and then restarting the system (TPM Resume) just before the timer expires.

TPM2_ACT_SetTimeout() must be properly authorized. Authorization may be provided either by *platformAuth* or by an ACT-specific *authPolicy*. The *startTimeout* parameter in TPM2_ACT_SetTimeout() is an integer number of seconds.

The *authPolicy* for an ACT can be changed by TPM2_SetPrimaryPolicy() using either *platformAuth* or the ACT-specific *authPolicy*.

The *authPolicy* of an ACT is initialized to an Empty Policy by TPM Reset or TPM Restart but is preserved during TPM Resume.

NOTE 2 After TPM Reset or TPM Restart, *phEnable* is SET, allowing the platform to initialize any ACT *authPolicy*.

2.1.3 Typical Use

A typical example for the use of an ACT is as a watchdog timer that will cause a platform reset when the timer reaches zero (expires). In a system using an ACT, a periodic platform action outside the TPM indicates that the timeout should be set anew using TPM2_ACT_SetTimeout(). The most common reason why timeout is not set anew is that the local system is not behaving properly because of some type of corruption (either inadvertent or malicious). The intent of the timer is that, in the absence of a properly authorized timeout extension, the platform would be reset, putting it back into a known state with the expectation that the corruption can be removed. The reason for having an authenticated timeout is to allow an external entity to make a decision about the health of the system.

The example above is not the only one supported by an ACT. In fact, this specification does not mandate that any specific platform behavior occur as a result of a timer expiring. The action on timer expiration may be chosen by a platform-specific specification or be vendor specific.

Because *platformAuth* may be used to change the *authPolicy* or set a *startTimeout* for any ACT, the platform firmware has ultimate control of the ACT. On each TPM Reset or TPM Restart, the platform firmware is expected to set the

authPolicy for all ACT using *platformAuth*. This specification mandates no specific policy for any ACT, but it is expected that, in most cases, the platform firmware will either:

a) Initialize an ACT *authPolicy* with a policy that can only be satisfied by an entity trusted by the platform manufacturer; or

b) Initialize the ACT *authPolicy* so that it can be changed using *ownerAuth.*

In case a), the platform firmware may set an initial timeout to ensure that some corrective action will occur if malware prevents the trusted entity from setting the ACT.

NOTE 1 This is how the platform would typically initialize a watchdog timer

In case b), the system software is expected to take control of the ACT. The platform would not set an initial timeout as it is possible that the ACT will not be used by the system software.

NOTE 2 If the platform firmware does not initialize the ACT *authPolicy* before *phEnable* CLEAR, then the ACT cannot be used.

2.1.4 Failure Mode

If the TPM enters failure mode, the ACT should continue to count down and trigger the specified event should it expire.

NOTE 1 If the failure mode was caused by a timer failure or affects functionality which is required for the platform-specific event, the ACT might not trigger reliably.

TPM2_ACT_SetTimeout() shall not be usable while a TPM is in Failure Mode. This means that the timeout cannot be extended and that timed events will occur if the TPM is not powered down or Reset before the ACT expires. A platform-specific specification may specify that an event will have no effect if the TPM is in Failure Mode.

A TPM may allow reading of the remaining ACT time (TPM2_GetCapability(*capability* = TPM_CAP_ACT) when the TPM is in Failure Mode.

2.1.5 Field Upgrade

The behaviour of a TPM during Field Upgrade is undefined. However, it is preferred that ACT continue to operate normally during Field Upgrade except that the ACT may not be changed by TPM2_ACT_SetTimeout().

NOTE Since *platformAuth* is required to start a Field Upgrade, *platformAuth* can be used to set the *startTimeout* for any active ACT to a value that is sufficient to allow a Field Upgrade to complete.

2.1.6 Typical ACT authPolicy

This clause describes a typical ACT authorization policy that authorizes setting of *startTimeout* with an authentication credential (key). The signature created by the authentication key is used as a cryptographically protected deferral ticket for the ACT.

The ACT *authPolicy* is constructed using TPM2_PolicySigned() and may include other policy components. Authorization by multiple entities can be achieved by combining multiple TPM2_PolicySigned() commands using AND or OR terms.

The deferral ticket is provided to the TPM in TPM2_PolicySigned() as the *auth* parameter (the signed authorization). The signature verification key, the *authObject* in TPM2_PolicySigned(), may be a symmetric or asymmetric key.

NOTE The advantage of an asymmetric signing key is that only the public key needs to be provisioned into the TPM. In the case of a symmetric HMAC key, the HMAC key's *authPolicy* should restrict the key to be used only for TPM2_PolicySigned() and not for other commands like TPM2_HMAC() (i.e. the TPM should not be able to issue its own deferral tickets).

For the *nonceTPM* and *expiration* parameters of TPM2_PolicySigned(), the following is recommended:

a) *nonceTPM* present and not an Empty Buffer

b) *expiration* > 0

Both settings ensure that a deferral ticket is single-use. The presence of *nonceTPM* in TPM2_PolicySigned() prevents the same signature being used multiple times within a policy session to defer the ACT indefinitely.

NOTE The *nonceTPM* for the policy session changes at the end of TPM2_ACT_SetTimeout(). This invalidates the previous signature and prevents replay of TPM2_ACT_SetTimeout() without getting a new signature from the authorized entity.

The non-negative *expiration* prevents TPM2_PolicySigned() creating a policy ticket which may be reused with TPM2_PolicyTicket() over a period of time to defer the ACT.

The ability to change *startTimeout* of TPM2_ACT_SetTimeout() should be limited by including *cpHash* in the ACT *authPolicy*. This can be achieved in two ways:

1) The ACT authorization policy includes TPM2_PolicyCpHash(). In this case, the entity setting the policy determines the *startTimeout* value.

2) The ACT authorization policy includes TPM2_PolicySigned() with *cpHashA* set. In this case, the signer (of the deferral tickets) determines the *startTimeout* value.

3 ACT Command Extension to Library Spec Part 2

3.1 Modification to 6.5.2 TPM_CC Listing

Add the command code for TPM2_ACT_SetTimeout() to Table 12 of Part 2 and update the last entry.

Table 1 — Definition of (UINT32) TPM_CC Constants (Numeric Order) <IN/OUT, S>

Name	Command Code	Dep	Comments
TPM_CC_CertifyX509	0x00000197		
TPM_CC_ACT_SetTimeout	0x00000198		
TPM_CC_LAST	0x00000198		Compile variable. May increase based on implementation.

3.2 Modification to 6.12 TPM_CAP

Add the ACT capability to read the ACT data with TPM2_GetCapability() to Table 22 in Part 2 and update the last entry.

Table 2 — Definition of (UINT32) TPM_CAP Constants

Capability Name	Value	Property Type	Return Type
TPM_CAP_AUTH_POLICIES	0x00000009	TPM_HANDLE(2)	TPML_TAGGED_POLICY
TPM_CAP_ACT	0x0000000A	TPM_HANDLE(2)	TPML_ACT_DATA
TPM_CAP_LAST	0x0000000A		

3.3 Modification to 7.4 TPM_RH (Permanent Handles)

Add the range of ACT authorization handles to Table 28 in Part 2 and update the last entry.

Table 3 — Definition of (TPM_HANDLE) TPM_RH Constants <S>

Name	Value	Type	Comments
TPM_RH_AUTH_FF	0x4000010F	A	End of the range of vendor-specific authorization values.
TPM_RH_ACT_0	0x40000110	A, P	Start of the range of authenticated timers
TPM_RH_ACT_F	0x4000011F	A, P	End of the range of authenticated timers
TPM_RH_LAST	0x4000011F	R	The top of the reserved handle area This is set to allow TPM2_GetCapability() to know where to stop. It may vary as implementations add to the permanent handle area.

3.4 Addition to 8 Attribute Structures

Add the following new attribute structure TPMA_ACT to clause 8 in Part 2.

3.4.1 TPMA_ACT

This attribute is used to report the ACT state. This attribute may be read using TPM2_GetCapability(*capability* = TPM_CAP_ACT, *property* = TPM_RH_ACT_"x" where "x" is the ACT number (0-F)). The signaled value must be preserved if the TPM has not lost power. The signaled value may be preserved over a power cycle of a TPM.

NOTE: The ACT signaled value is reset to zero when the ACT is next accessed by TPM2_ACT_SetTimeout().

Table 4 — Definition of (UINT32) TPMA_ACT Bits

Bit	Name	Definition
0	signaled	**SET (1):** The ACT has signaled **CLEAR (0):** The ACT has not signaled
1	preserveSignaled	**SET (1):** The ACT signaled bit is preserved over a power cycle **CLEAR (0):** The ACT signaled bit is not preserved over a power cycle
31:2	Reserved	shall be zero

3.5 Addition to 9 Interface Types

Add the following new interface type TPMI_RH_ACT to clause 9 in Part 2.

3.5.1 TPMI_RH_ACT

This interface type is used to identify the ACT instance used in TPM2_ACT_SetTimeout().

Table 5 — Definition of (TPM_HANDLE) TPMI_RH_ACT Type

Value	Comments
{TPM_RH_ACT_0:TPM_RH_ACT_F}	handles for the Authenticated Countdown Timers
#TPM_RC_VALUE	response code returned when the unmarshaling of this type fails

3.5.2 TPMI_RH_HIERARCHY_POLICY

This interface type is used as the type of a handle in a command when the handle is required to be one of the hierarchy selectors, the Lockout Authorization, or an ACT. This type is used in TPM2_SetPrimaryPolicy().

Table 6 — Definition of (TPM_HANDLE) TPMI_RH_HIERARCHY_POLICY Type <IN>

Values	Comments
TPM_RH_OWNER	Storage hierarchy
TPM_RH_PLATFORM	Platform hierarchy
TPM_RH_ENDORSEMENT	Endorsement hierarchy
TPM_RH_LOCKOUT	Lockout Authorization
{TPM_RH_ACT_0:TPM_RH_ACT_F}	Authenticated Countdown Timer
#TPM_RC_VALUE	response code returned when the unmarshaling of this type fails

3.6 Addition to 10.8 Property Structures

Add the following new TPMS_ACT_DATA structure to clause 10.8 in Part 2.

3.6.1 TPMS_ACT_DATA

This structure is used in TPM2_GetCapability() to return the ACT data.

Table 7 — Definition of TPMS_ACT_DATA Structure <OUT>

Parameter	Type	Description
handle	TPM_HANDLE	a permanent handle
timeout	UINT32	the current timeout of the ACT
attributes	TPMA_ACT	the state of the ACT

3.7 Addition to 10.9 Lists

Add the following new TPML_ACT_DATA structure to clause 10.9 in Part 2.

3.7.1 TPML_ACT_DATA

This list is used to report the timeout and state for the ACT. This list may be generated by TPM2_GetCapabilty(). Only implemented ACT are present in the list

NOTE MAX_ACT_DATA = MAX_CAP_DATA / sizeof(TPMS_ACT_DATA).

Table 8 — Definition of TPML_TAGGED_POLICY Structure <OUT>

Parameter	Type	Description
count	UINT32	number of ACT instances A value of zero is allowed.
policies[count]{:MAX_ACT_DATA}	TPMS_ACT_DATA	array of ACT data

4 ACT Command Extension to Library Spec Part 3

4.1 TPM2_ACT_SetTimeout

4.1.1 General Description

This command is used to set the time remaining before an Authenticated Countdown Timer (ACT) expires.

This command sets TPMS_ACT_DATA.*timeout* (ACT Timeout) to *startTimeout*. The *startTimeout* value is an integer number of seconds and may be zero. The *startTimeout* parameter may be greater, equal, or less than the current value of ACT Timeout.

When ACT Timeout is non-zero, it will count down, once per second until it reaches zero, at which time the *signaled* attribute of the TPMA_ACT associated with *actHandle* is SET.

There are four states for ACT Timeout and *startTimeout*. The *signaled* attribute will be set as follows:

1) If ACT Timeout is zero and *startTimeout* is non-zero, then *signaled* will be CLEAR.

2) If ACT Timeout is non-zero and *startTimeout* is non-zero, then *signaled* will be CLEAR.

3) If ACT Timeout is zero and *startTimeout* is zero, then *signaled* will be CLEAR.

4) If ACT Timeout is non-zero and *startTimeout* is zero, then *signaled* will be SET.

NOTE The ACT signals on a transition from non-zero to zero. The transition can occur either due to TPM2_ACT_SetTimeout() or a decrement. The effect of *signaled* is platform dependent.

4.1.2 Command and Response

Table 9 — TPM2_ACT_SetTimeout Command

Type	Name	Description
TPMI_ST_COMMAND_TAG	tag	TPM_ST_SESSIONS
UINT32	commandSize	
TPM_CC	commandCode	TPM_CC_ACT_SetTimeout
TPMI_RH_ACT	@actHandle	Handle of the selected ACT Auth Index: 1 Auth Role: USER
UINT32	startTimeout	the start timeout value for the ACT in seconds

Table 10 — TPM2_ACT_SetTimeout Response

Type	Name	Description
TPM_ST	tag	see clause 6
UINT32	responseSize	
TPM_RC	responseCode	

4.2 Modification to 30.2.1 General Description (of TPM2_GetCapability)

Add TPM_CAP_ACT to the capability table in Part 3 and add the description for TPM_CAP_ACT as new bullet point.

capability	***property***	**Return Type**
TPM_CAP_AUTH_POLICIES	TPM_HANDLE(2)	TPML_TAGGED_POLICY
TPM_CAP_ACT	TPM_HANDLE(2)	TPML_ACT_DATA
TPM_CAP_VENDOR_PROPERTY	manufacturer specific	manufacturer-specific values
NOTES: (1) The TPM_ALG_ID or TPM_ECC_CURVE is cast to a UINT32 (2) The TPM will return TPM_RC_VALUE if the handle does not reference the range for permanent handles.		

- TPM_CAP_ACT – Returns a list of TPMS_ACT_DATA, each of which contains the handle for the ACT, the remaining time before it expires, and the ACT attributes.

4.3 Modification to 24.3 TPM2_SetPrimaryPolicy

Enhance the General Description of TPM2_SetPrimaryPolicy(), change the type of the *authHandle* parameter from TPMI_RH_HIERARCHY_AUTH to TPMI_RH_HIERARCHY_POLICY and add TPMI_RH_ACT to the description of the *authHandle* parameter.

4.3.1 General Description

This command allows setting of the authorization policy for the lockout (*lockoutPolicy*), the platform hierarchy (*platformPolicy*), the storage hierarchy (*ownerPolicy*), and the endorsement hierarchy (*endorsementPolicy*). On TPMs implementing Authenticated Countdown Timers (ACT), this command may also be used to set the authorization policy for an ACT.

The command requires an authorization session. The session shall use the current *authValue* or satisfy the current *authPolicy* for the referenced hierarchy or the ACT.

The policy that is changed is the policy associated with *authHandle*.

If the enable associated with *authHandle* is not SET, then the associated authorization values (*authValue* or *authPolicy*) may not be used, and the TPM returns TPM_RC_HIERARCHY.

NOTE The enable associated with the *authValue* of an ACT is *phEnable*. The *authPolicy* of an ACT has no associated enable.

When *hashAlg* is not TPM_ALG_NULL, if the size of *authPolicy* is not consistent with the hash algorithm, the TPM returns TPM_RC_SIZE.

4.3.2 Command and Response

Table 11 — TPM2_SetPrimaryPolicy Command

Type	Name	Description
TPMI_ST_COMMAND_TAG	tag	TPM_ST_SESSIONS
UINT32	commandSize	
TPM_CC	commandCode	TPM_CC_SetPrimaryPolicy {NV}
TPMI_RH_HIERARCHY_POLICY	@authHandle	TPM_RH_LOCKOUT, TPM_RH_ENDORSEMENT, TPM_RH_OWNER, TPMI_RH_ACT or TPM_RH_PLATFORM+{PP} Auth Index: 1 Auth Role: USER
TPM2B_DIGEST	authPolicy	an authorization policy digest; may be the Empty Buffer If *hashAlg* is TPM_ALG_NULL, then this shall be an Empty Buffer.
TPMI_ALG_HASH+	hashAlg	the hash algorithm to use for the policy If the *authPolicy* is an Empty Buffer, then this field shall be TPM_ALG_NULL.

4.4 Modification to 24.8 TPM2_HierarchyChangeAuth

Enhance the general description of TPM2_HierarchyChangeAuth() with:

The ACT has no associated *authValue* that can be changed by this command.

4.5 Modification to 9.3.1 General Description (of TPM2_Startup)

Enhance the list of actions on TPM Reset, TPM Restart, and TPM Resume by the ACT actions.

On TPM Reset and on TPM Restart,

- ACT timeout is reset to zero, and the ACT specific *authPolicy* is set to Empty Buffer

On TPM Resume

- ACT timeout and the ACT specific *authPolicy* are preserved

5 ACT Command Extension to Library Spec Part 1 Annex F

5.1 ACT

The list specifies the information a platform spec or vendor needs to define:

- Number of supported ACT instances (usually 0 or 1).
- Trigger event when ACT times out.
- Whether TPM2_ClockRateAdjust() may affect the ACT rate, and maximum adjustment.
- ACT behavior if TPM is in low power state (sleep mode) (typically, ACT must advance).
- Whether the ACT state must be preserved over a power cycle (setting of *preserveSignaled* attribute).
- Whether clearing the *signaled* attribute (when ACT Timeout is zero and *startTimeout* is zero) must also clear the associated trigger event.
- Whether the remaining ACT timeout must be retrievable in TPM Failure Mode.

6 References

[1] TPM 2.0 Library Specification, Revision 1.55

https://trustedcomputinggroup.org/specifications-public-review/