TCG Storage
Enterprise SSC Feature Set
Locking LBA Ranges Control

Specification Version 1.00
Revision 1.00

May 2, 2014

Contacts:
admin@trustedcomputinggroup.org

TCG

TCG Published

Disclaimers, Notices, and License Terms

Table of Contents

1 Introduction

1.1 Document Purpose

This document introduces a new Feature Set with the intention to provide better control over LBA ranges locking state. The TCG Enterprise SSC spec has a provisioning option to lock LBA ranges upon some sort of Reset using the `LockOnReset` column of the TCG `Locking`table.

However, there could be events which don't fall into the Reset category from the point of view of any transport protocol but upon which it is still desirable to lock some LBA range(s).

This Feature Set expands the transport layer (SATA, SAS, etc…) dependent Resets definition by introducing Secure Events and Secure Conditions of a physical or logical nature which in general are made known to the system by entities other than the transport layer – so called Security Sensors.

1.2 Scope and Intended Audience

This specification defines the Locking LBA Ranges Control functionality for the Enterprise Security Subsystem Class (SSC). Any SD (Storage Device) that claims Enterprise SSC Locking LBA Ranges Control compatibility has to conform to this specification.

The intended audience for this document is storage device and peripheral device manufacturers and developers that wish to tie storage devices and peripherals into trusted platforms.

1.3 Key Words

Key words are used to signify SSC requirements.

The Key Words "**SHALL**", "**SHALL NOT**", "**SHOULD**," and "**MAY**" are used in this document. These words are a subset of the RFC 2119 key words used by TCG, and have been chosen since they map to key words used in T10/T13 specifications. These key words are to be interpreted as described in [1]. In addition to the above key words, the following are also used in this document to describe the requirements of particular features, including tables, methods, and usages thereof.

- **Mandatory (M):** When a feature is Mandatory, the feature SHALL be implemented. A Compliance test SHALL validate that the feature is operational.
- **Optional (O):** When a feature is Optional, the feature MAY be implemented. If implemented, a Compliance test SHALL validate that the feature is operational.
- **Excluded (X):** When a feature is Excluded, the feature SHALL NOT be implemented. A Compliance test SHALL validate that the feature is not operational.
- **Not Required (N)** When a feature is Not Required, the feature MAY be implemented. No Compliance test is required.

1.4 Document References

[1]. IETF RFC 2119, 1997, "Key words for use in RFCs to Indicate Requirement Levels"
[2]. Trusted Computing Group (TCG), "TCG Storage Architecture Core Specification", see [3] for the applicable specification version.
[3]. Trusted Computing Group (TCG), "TCG Storage Security Subsystem Class: Enterprise", Version 1.00.

1.5 Document Precedence

In the event of conflicting information in this specification and other documents, the precedence for requirements is:

1. This specification and [3] (these two documents are at the same level of precedence, and SHALL NOT conflict with each other).
2. TCG Storage Architecture Core Specification [2].

1.6 Dependencies on Other Feature Sets

None

1.7 Interactions with Other Feature Sets

None

1.8 Terminology

This feature defines the following new terms.	**Term Definition**
FSM	Finite State Machine
Secure Timeout	The maximum time between Secure Event and Secure Condition value being changed at which moment range lock action MAY be performed by SD on some LBA Range(s).
Secure Condition	SD state with respect to a particular Security Sensor. The Secure Condition MAY be part of Security Expression which SHALL be evaluated to determine SD behavior upon invocation by the host `Set` method, changing LBA Range `ReadLocked` or `WriteLocked` columns value in the `Locking` table from TRUE to FALSE or upon changing any of Secure Conditions the Secure Expression contains. SD MAY lock or unlock one or more LBA ranges if Secure Condition changes. *Begin Informative Content* At any given moment Secure Condition may or may not be the same as Security Sensor State of this sensor. It depends on the Secure Timeout value configured for particular Security Sensor: if the timeout value is not 0, the Secure Condition will retain previous value until Secure Timeout expiration. *End Informative Content*
Secure Event	Notification that the Secure State of a particular Security Sensor changed. *Begin Informative Content* If Secure Timeout is not 0, Secure Event by itself does not lead to any range being locked because Secure Condition value doesn't change right away. *End Informative Content*
Security Sensor	Physical device or SD software module capable of generating events which are considered to be important enough to be called Secure Events. *Begin Informative Content* There could be two types of Secure Events, associated with each sensor: ON (state changes from FALSE to TRUE) and OFF (state changes from TRUE to FALSE). *End Informative Content*
Security Sensor State	Security Sensor State is the Internal state of Security Sensor. When Security Sensor State changes, Secure Event is generated. *Begin Informative Content* Security Sensor State is calculated on current read value of its correspondent Security Sensor. *End Informative Content*

This feature defines the following new terms.	**Term Definition**
Security Expression	Boolean Expression consisting of any number of Secure Conditions, which MAY be connected by Boolean operations (“and”, “or” and “not”).
TCG Reset	A high-level reset type defined in the Core Spec.
TPer	The TCG security subsystem within a storage device.
Trusted Peripheral	A TPer.

1.9 Legend

The following legend defines SP table cell coloring coding. This color coding is informative only. The table cell content is normative.

Table 1. SP Table Legend

Table Cell Legend	R – W	Value	Access Control	Comment
Arial-Narrow	Read-only	Enterprise SSC Specified.	Fixed	• Cell content is Read Only. • Access control is fixed. • Value is specified by the Enterprise SSC
Arial Narrow bold-under	Read-only	(VU)	Fixed	• Cell content is Read-Only. • Access control is fixed. • Values are Vendor Unique (VU). A minimum or maximum value may be specified.
Arial-Narrow	Not Defined	(N)	Not Defined	• Cell content is (N). • Access control is not defined. • Any text in table cell is informative only. • A Get MAY omit this column from the method response.
Arial Narrow bold-under	Write	Preconfigured, user personalizable	Preconfigured, user personalizable	• Cell content is writable. • Access control is personalizable • Get Access Control is not described by this color coding
Arial-Narrow	Write	Preconfigured, user personalizable	Fixed	• Cell content is writable. • Access control is fixed. • Get Access Control is not described by this color coding

2 Security Sensors

2.1 Overview

The goal of this specification is to introduce additional criteria for changing the locking state of a LBA range. This functionality could be very helpful for customers who want better control over LBA ranges' locking states.

Begin Informative Content

There could be a need to lock certain LBA Ranges if certain events happen in the system. Equally, it could be necessary to reject an unlock request of some LBA ranges if the system is in a certain condition.

The SD can be notified of these events or can examine conditions using special entities (physical devices or software modules) called Security Sensors (see examples in the Appendix 1 - Security Sensors Examples).

End Informative Content

2.2 Security Sensor State

Each Security Sensor has its own state (associated with some physical signal or software variable) and it SHALL send a Secure Event when its state changes. The Secure State SHALL be 0 (FALSE) or 1 (TRUE) and SHALL be always readable by Drive Controller FW.

Begin Informative Content

Inputs from some physical devices behind Security Sensor could require to be converted into one of two possible Security Sensor States – 0 or 1 – unless the device is already generating such "binary" signal. Thresholds can be used (see Table 9) for such a conversion but in general it's up to vendor how to convert physical device readings into 0/1 value.

Some physical devices may contain more than one such sensor – this is also Vendor dependent. While in general Security Sensors are not influenced by any transport layer resets, Vendor may decide otherwise for some sensors implementation.

End Informative Content

The SD MAY implement pure software Security Sensors, generating such events which don't need any special hardware for detection.

Begin Informative Content

Two examples of Software Security Sensors are Other Ranges Unlocked and Host Activity which can be found in sections 5.6 and 5.7 respectively. *End Informative Content*

2.3 Secure Timeout and Secure Condition

Secure Timeout shall be configured separately for both (ON and OFF) Secure Events of each Security Sensor. Secure Timeout determines when a Secure Condition value will be set to the Security Sensor's state. That MAY lead to re-evaluation of Security Expressions of some LBA range(s) and as a result to them being locked.

If Secure Timeout is set to zero, the action above SHALL be performed immediately after Secure Event.

The non-zero values of the Secure Timeout MAY be used to give the host an opportunity to lock ranges programmatically.

Begin Informative Content

The host may be able to lock ranges programmatically as it may have sensors similar to those of the SD itself. However if the SD is under an attack, which includes disconnecting the host, this is not an option.

End Informative Content

2.4 Security Sensors Table

Security Sensors are SD intrinsic features and they are defined in the SP-global `SecuritySensors` table in the Locking Template (see 4.3) – one row for each Security Sensor. The host MAY change parameters of these sensors (such as the Secure Timeout) as well as define which LBA ranges' locking states depend on each Security Sensor.

2.5 Security Expressions

For each LBA range a Security Expression SHALL be provisioned (MAY be empty). A Security Expression is a variable of the SEC_Expression type (see 3.2.4) and as such is a list of Boolean operations and reference(s) to Security Sensors. Its Boolean value SHALL be calculated by substitution of each Security Sensor reference by correspondent Secure Condition value. Empty or malformed Security Expression SHALL be assigned a value of 0 (FALSE).

3 SSC Specific Functionality

3.1 Overview

Chapter 3 defines the SSC-specific functionality required to support the Locking LBA Ranges Control Feature Set.

3.2 Additional Column Types

These column types describe system behavior with respect to Security Sensors.

3.2.1 Security Sensors Reference

The security_sensor_ref (see Table 2) is a row number in the `SecuritySensors` table (see Table 9).

Table 2. security_sensor_ref

UID	Name	Format
`00 00 00 05 00 00 0C 0E`	security_sensor_ref	Restricted_Reference_Type{5}, ref {SecuritySensorsTableUID}

3.2.2 Security Sensor Boolean

This enumeration is used to identify the Boolean operators "And", "Or", and "Not".

Table 3. security_sensor_boolean

UID	Name	Format
`00 00 00 05 00 00 0C 1E`	security_sensor_boolean	Enumeration_Type, 0, 2

The enumeration values are associated with Boolean operators as defined in Table 4.

Table 4. security_sensor_boolean Enumeration Values

Enumeration Value	Operator
0	And
1	Or
2	Not

3.2.3 Security Expression Element

This is an Alternative_Type where the options are either a uidref to a Security Sensor object or one of the security_sensor_boolean options. This type is used within the list to form a Boolean expression of SD Secure Conditions associated with any number of Security Sensors.

Table 5. SEC_Expression_Element

UID	Name	Format
`00 00 00 05 00 00 06 07`	SEC_Expression_Element	Alternative_Type, security_sensor_ref, security_sensor_boolean

3.2.4 Security Expression

The Security Expression is a list type made up of SEC_Expression_Element. The size of the Security Expression list is implementation-dependant.

There SHALL be a Boolean value associated with any SEC_Expression. To get this value, all SEC_Expression elements of security_sensor_ref type SHALL be replaced by Secure Condition values of the correspondent Security Sensors (if the correspondent row doesn't exist in the `SecuritySensors` table, Secure Condition value SHALL be assumed to be FALSE) and resulting Boolean expression SHALL be evaluated. The value obtained by this method is this SEC_Expression Boolean value. A SEC_Expression which results in malformed or empty Boolean expressions SHALL be assumed to have a value of 0 (FALSE).

Table 6. SEC_Expression

UID	Name	Format
`00 00 00 05 00 00 08 06`	SEC_Expression	List_Type, *, SEC_Expression_Element

Table 7 represents an example of SEC_Expression encoding. It uses the same precedence rules ("reversed polish" notation) as ACE_expression type – see section 5.1.3.3 of [2].

Table 7. SEC_Expression Encoding Example

Token	Meaning
F0	Start List
F2	Start Name
A4	00 00 0C 0E Half-UID – security_sensor_ref
A4	00 00 00 02 security_sensor_ref (row 2 in the `SecuritySensors` table)
F3	End Name
F2	Start Name
A4	00 00 0C 0E Half-UID – security_sensor_ref
A4	00 00 00 01 security_sensor_ref (row 1 in the `SecuritySensors` table)
F3	End Name
F2	Start Name
A4	00 00 04 0E Half-UID - security_sensor_boolean
00	security_sensor_boolean - AND
F3	End Name
F2	Start Name
A4	00 00 0C 0E Half-UID – security_sensor_ref
A4	00 00 00 03 security_sensor_ref (row 3 in the `SecuritySensors` table)
F3	End Name
F2	Start Name
A4	00 00 04 0E Half-UID - security_sensor_boolean
01	security_sensor_boolean - OR
F3	End Name
F2	Start Name
A4	00 00 0C 0E Half-UID – security_sensor_ref

A4	00 00 00 00 security_sensor_ref (row 0 in the `SecuritySensors` table)
F3	End Name
F1 F2	End List Start Name
A4	00 00 04 0E Half-UID - security_sensor_boolean
00	security_sensor_boolean - AND
F3	End Name

3.3 Modified Methods

This specification does not require any methods to be modified.

3.4 Additional Tables

SD SHALL implement the `SecuritySensors` table.

3.4.1 SecuritySensors Table

The `SecuritySensors` table belongs to the `Locking` Template (see 4.3). The table contains at least one row for each Security Sensor that the SD supports.

4 Feature Set Requirements

This section defines the Mandatory (M) and Optional (O) requirements for the Locking LBA Ranges Control Feature Set, when it is implemented in a TCG Enterprise SSC compliant SD.

4.1 Level 0 Discovery

A SD that implements the Locking LBA Ranges Control Feature Set SHALL return the Locking LBA Ranges Control Feature Descriptor as described in 4.1.1, in addition to the Level 0 Discovery response requirements defined in [3].

4.1.1 Locking LBA Ranges Control Feature (Feature Code = 0401h)

The Locking LBA Ranges Control Feature descriptor SHALL be returned when the SD supports the Locking LBA Ranges Control. This descriptor contents are defined in the Table 8 below.

Table 8. Locking LBA Ranges Control Feature Descriptor Format

Byte \ Bit	7	6	5	4	3	2	1	0
0	Feature Code (0x0401)							
1								
2	Version				Reserved			
3	Length							
4	Reserved							
5 - 15	Reserved for future Locking LBA Ranges Control parameters							

4.1.1.1 Version

This field is the version of the Locking LBA Ranges Control Feature Set. The current version is 1.

4.1.1.2 Length

This field represents the number of following bytes. Fixed value is 12 (0x0C).

4.2 Admin SP

This feature set requires no additions to the Admin SP.

4.3 Locking SP

4.3.1 Overview

A SD that supports the Locking LBA Ranges Control Feature Set SHALL implement the additions specified in this section.

4.3.2 SecuritySensors Table

This Object table contains one row for each Security Sensor, supported by SD hardware and firmware. The host cannot add/delete rows to/from this table. The host MAY change all parameters of these sensors except `UID`, `Name` and `SecureCondition` as well as define which LBA ranges shall monitor each Security Sensor to which end `SecurityExpression` column is added to the TCG `Locking` table (see 4.3.3).

The table belongs to `Locking` Template and there is only one such table per SP.

The table UID is `00 00 00 01 00 00 08 06`.

4.3.2.1 SecuritySensors Table Columns

Begin Of Informative Content

Each column of the SecuritySensors table (see Table 9) has default value which it assumes at the time of manufacturing. The host MAY later modify values of those columns which it is allowed to modify and those modified values are retained over power cycles until changed again by the host.

The SecureCondition is an exception – after Power On its value should be set to FALSE.

End Of Informative Content

Table 9. SecuritySensors Table Description

Column Name	Manufacturing Default Value	IsUnique	Column Type
UID	00 00 08 06 00 00 *rr* *rr*		uid
Name	VU		name
Units	VU		name
Initial	VU		boolean
SensorEnabled	VU		boolean
EventOnThreshold	VU		uinteger_4
EventOnThresholdFraction	VU		Fraction
EventOnSecureTout	VU		uinteger_4
EventOffThreshold	VU		uinteger_4
EventOffThresholdFraction	VU		Fraction
EventOffSecureTout	VU		uinteger_4
SecureCondition	VU		boolean

SD SHALL populate the SecuritySensors table, creating at least one row for each physical and software sensor it supports. For all such rows the Initial column value SHALL be TRUE.

The SecuritySensors table MAY contain multiple rows with same Name/Unit values which refer to the same physical or software sensor. The host MAY create new rows in the table (the Initial column value for such new rows SHALL be FALSE) but it SHALL use Name/Units combination from some of the existing rows. The host MAY delete rows it created previously but SHALL NOT be able to delete rows which were created by SD (have their Initial column value TRUE).

Begin Of Informative Content

Multiple rows with the same Name column value represent multiple Security Sensors using the same physical device or software module. For example, there could be two instances of Orientation Security Sensor (see 5.3) with different EventOnThreshold column values. Two LBA ranges which refer to these Security Sensors will be locked which SD tilt reaches different angles.

End Of Informative Content

4.3.2.1.1 UID

This is the unique identifier of this row of the SecuritySensors table. This column SHALL NOT be modifiable by the host. Value is 00 00 08 06 00 00 rr rr where rr rr is row number.

4.3.2.1.2 Name

This is the manufacturer-defined name for this Security Sensor. This column SHALL NOT be modifiable by the host. Name is unique identifier of physical or software sensor behind this Security Sensor.

4.3.2.1.3 Units

The `Units` column specifies the units (such as "grad" or "m/sec") that quantify the output of a Security Sensor. A `Units` parameter is meaningful only for certain types of sensors like orientation or movement detectors. Otherwise the `Units` column value SHALL be a string of length zero. This column SHALL NOT be modifiable by the host.

4.3.2.1.4 Initial

The default value for this column SHALL be TRUE for all rows created by SD. The host MAY only create or delete rows with Initial value FALSE. The host SHALL NOT be able to modify this column value.

4.3.2.1.5 SensorEnabled

The sensor SHALL be examined only if this column value is TRUE.

4.3.2.1.6 EventOnThreshold

This is the value of the integer part of the input value to a Security Sensor that triggers an ON Secure Event when the input is increasing. `EventOnThreshold` is measured in `Units` and SHALL be ignored if `Units` is an empty string. The actual threshold value SHALL be calculated as

```
EventOnThreshold + EventOnThresholdFraction/1000
```

4.3.2.1.7 EventOnThresholdFraction

This is the value of the fractional part of the input value to a Security Sensor that triggers an ON Secure Event when the input is increasing. `EventOnThresholdFraction` is measured in thousandth parts of `Units` and SHALL be ignored if `Units` is an empty string.

4.3.2.1.8 EventOnSecureTout

This is a Secure Timeout (in milliseconds) caused by an ON Secure Event. After its expiration, Secure Condition value will become 1 (TRUE). If Secure Timeout has been started already (Secure Event ON did happen), `Set` method, invoked by Host on the `EventOnSecureTout` column SHALL not affect this timer instance.

4.3.2.1.9 EventOffThreshold

This is the value of the integer part of the input value to a Security Sensor that triggers an OFF Secure Event when the input is decreasing. `EventOffThreshold` is measured in `Units` and SHALL be ignored if `Units` is an empty string. The actual threshold value SHALL be calculated as

```
EventOffThreshold + EventOffThresholdFraction/1000
```

4.3.2.1.10 EventOffThresholdFraction

This is the value of the fractional part of the input value to a Security Sensor that triggers an OFF Secure Event when the input is decreasing. `EventOffThresholdFraction` is measured in thousandth parts of `Units` and SHALL be ignored if `Units` is an empty string. The actual threshold value SHALL be calculated as

```
EventOffThreshold + EventOffThresholdFraction/1000
```

4.3.2.1.11 EventOffSecureTout

This is a Secure Timeout (in milliseconds) caused by an OFF Secure Event. After its expiration, Secure Condition value will become 0 (FALSE). If Secure Timeout has been started already (Secure Event OFF did happen), `Set` method, invoked by the host on the `EventOffSecureTout` column SHALL not affect this timer instance.

4.3.2.1.12 SecureCondition

This is the current Secure Condition value. This column SHALL NOT be modifiable by the host.

4.3.2.2 Security FSM State Diagram

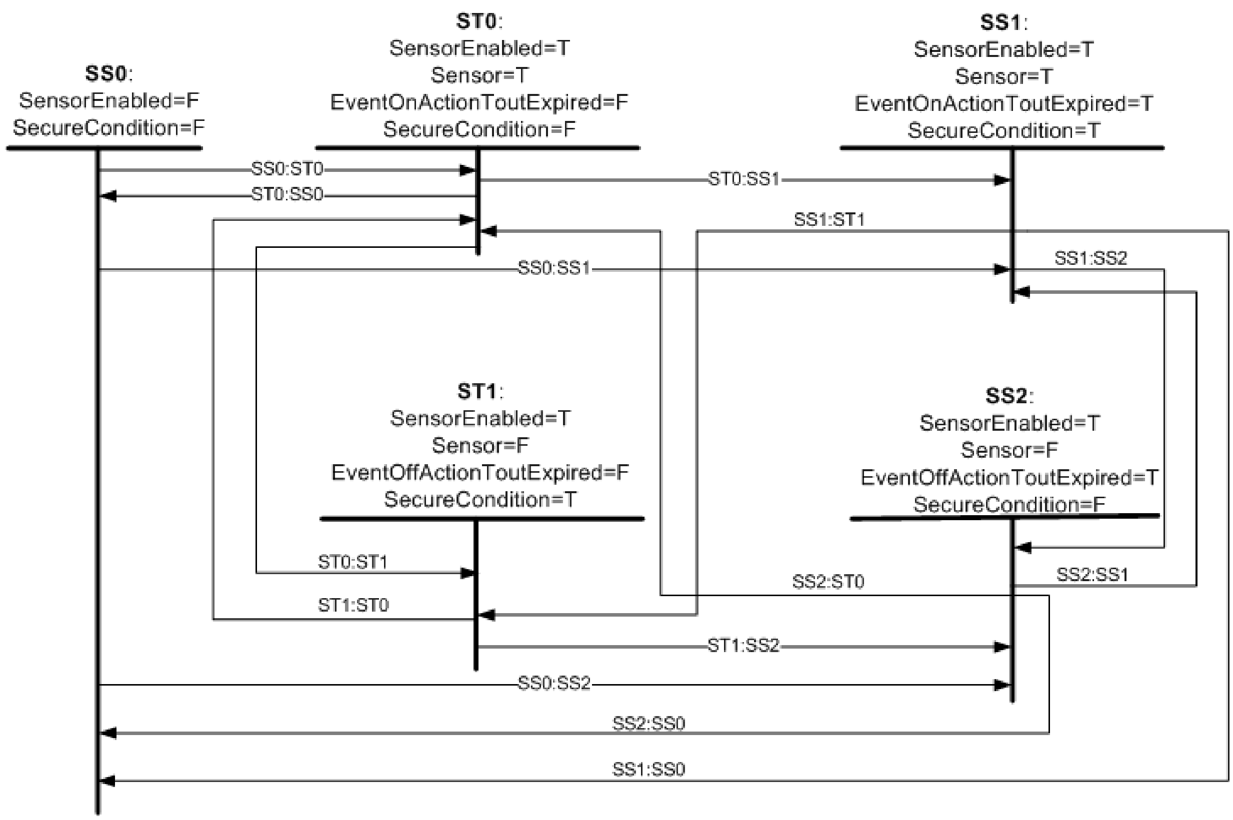
Figure 1a.Figure 1 displays the states and state transitions for Security FSM State. Transition is driven by enabling/disabling the Security Sensor, by its state changing and Secure Timeout expiration.

Begin Informative Content

This FSM determines what will be the Secure Condition of the system associated with a particular Security Sensor at any given moment. Note that Secure Condition(s) are used to evaluate `SecurityExpression` column of `Locking` table and therefore influence LBA range locking state.

End Informative Content

Figure 1 **Security FSM State Transition Diagram**



4.3.2.2.1 State Descriptions

This section describes the states that are used in Figure 1, and the column values that each state represents.

SS0 SensorEnabled=F
This describes the state where this particular Security Sensor is turned off. The sensor input is ignored.

ST0 SensorEnabled=T/Sensor=T/EventOnActionToutExpired=F
This describes the state where Security Sensor is enabled and its state is TRUE but Secure Timeout has not expired yet.

SS1 SensorEnabled=T/Sensor=T/EventOnActionToutExpired=T
This describes the state where Security Sensor is enabled, its state is TRUE and Secure Timeout has already expired.

ST1 SensorEnabled=T/Sensor=F/EventOnActionToutExpired=F
This describes the state where Security Sensor is enabled and its state is FALSE but Secure Timeout has not expired yet.

SS2 SensorEnabled=T/Sensor=F/EventOnActionToutExpired=T
This describes the state where Security Sensor is enabled, its state is FALSE and Secure Timeout has already expired.

4.3.2.2.2 State Change Descriptions

This section describes the state changes depicted in Figure 1. In parentheses next to each state transition identifier are the values that change to cause that transition.

SS0:ST0 (Enabling while Sensor=T, EventOnSecureTout > 0)

This state change occurs as a result of enabling a sensor after the sensor value has been read as TRUE.

SS0:SS1 (Enabling while Sensor=T, EventOnSecureTout = 0)

This state change occurs as a result of enabling a sensor after the sensor value has been read as TRUE and if `EventOnActionTout` column value is 0.

SS0:SS2 (Enabling while Sensor=F)

This state change occurs as a result of enabling a sensor after the sensor value has been read as FALSE.

ST0:SS1 (Secure Timeout Expired while Sensor=T)

This state change occurs when a Secure Timeout expires after the sensor value has been read as TRUE. The FSM State SHALL become **SS1** and the `SecurityExpression` column SHALL be examined for each row (LBA range) in the `Locking` table.

ST0:ST1 (Sensor Changes from TRUE to FALSE while waiting for Secure Timeout to expire)

This state change occurs when a sensor value changes from TRUE to FALSE prior to Secure Timeout expiration. This change SHALL set the timer equal to the `EventOffActionTout` column value.

ST1:SS2 (Secure Timeout Expired while Sensor=F)

This state change occurs when a Secure Timeout expires after the sensor value has been read as FALSE. FSM State SHALL become **SS2** and `SecurityExpression` column SHALL be examined for each row (LBA range) in the `Locking` table.

ST1:ST0 (Sensor Changes from FALSE to TRUE while waiting for Secure Timeout to expire)

This state change occurs when a sensor value changes from FALSE to TRUE prior to Secure Timeout expiration. This change SHALL set the timer value to the `EventOnSecureTout`.

SS1:ST1 (Sensor Changes from TRUE to FALSE, EventOffSecureTout > 0)

This state change occurs when a sensor value changes from TRUE to FALSE while `EventOffSecureTout` > 0. This change SHALL set the timer value to the `EventOffSecureTout`.

SS1:SS2 (Sensor Changes from TRUE to FALSE, EventOffSecureTout = 0)

This state change occurs when a sensor value changes from TRUE to FALSE while `EventOffSecureTout` = 0.

SS2:ST0 (Sensor Changes from FALSE to TRUE, EventOnSecureTout > 0**)**

This state change occurs when a sensor value changes from FALSE to TRUE while `EventOnSecureTout`>0. This change SHALL set the timer to the `EventOnSecureTout` value.

SS2:SS0 (Sensor Changes from FALSE to TRUE, EventOnSecureTout = 0)

This state change occurs when a sensor value changes from FALSE to TRUE while `EventOnSecureTout` = 0.

ANY:SS0 (Disabling)

This state change occurs when the value of the `SensorEnabled` column of the `SecuritySensors` table is set to FALSE (a sensor is disabled). The `SecureCondition` value SHALL be set to FALSE.

4.3.3 Locking Table

The Locking table SHALL change as explained below.

4.3.4 SecurityExpression Column

Table 10. Locking Table Description

Column Name	Default	IsUnique	Column Type
SecurityExpression	VU		SEC_Expression

4.3.4.1 SecurityExpression

The SecurityExpression Column of the SEC_Expression type SHALL be added to the Locking table. The column contains the SEC_Expression that SHALL be evaluated for this LBA range and is modifiable by the host.

The SecurityExpression value SHALL be ignored if both ReadLockEnabled and WriteLockEnabled columns values are FALSE. Otherwise the SecurityExpression value SHALL be re-evaluated each time the SecurityExpression itself or any of its Security Sensors' parameters are modified by the host by invocation of a Set method as well as upon change of a Secure Condition of any Security Sensor the SecurityExpression includes.

This re-evaluation SHALL be completed immediately after this Set method completed prior to any other method being processed by SD.

Begin Informative Content

The following Set methods invoked by the host can result in changing of the SecurityExpression value:

A. Modifying any SecuritySensors table column for any Security Sensor which is a member of this SecurityExpression.

B. Adding/deleting rows in the SecuritySensors table.

C. Adding/removing elements (Boolean operations, grouping and Security Sensors' reference) in the SecurityExpression itself.

End Informative Content

An LBA range for which ReadLockEnabled=TRUE and ReadLocked=FALSE (LBA range is unlocked for Read operation) SHALL be locked for Read operation (ReadLocked value set to TRUE) at the moment its SecurityExpression value changes from 0 (FALSE) to 1 (TRUE).

An LBA range for which ReadLockEnabled=FALSE and ReadLocked=FALSE (LBA range is unlocked for Read operation) and SecurityExpression=FALSE SHALL be locked for Read operation (ReadLocked value set to TRUE) at the moment its ReadLockEnabled value changes from 0 (FALSE) to 1 (TRUE).

An LBA range for which WriteLockEnabled=TRUE and WriteLocked=FALSE (LBA range is unlocked for Write operation) SHALL be locked for Write operation (WriteLocked value set to TRUE) at the moment its SecurityExpression value changes from 0 (FALSE) to 1 (TRUE).

An LBA range for which WriteLockEnabled=FALSE and WriteLocked=FALSE (LBA range is unlocked for Read operation) and SecurityExpression=FALSE SHALL be locked for Read operation (WriteLocked value set to TRUE) at the moment its WriteLockEnabled value changes from 0 (FALSE) to 1 (TRUE).

A previously locked for Read or Write operation LBA Range SHALL NOT be unlocked either for Read or for Write operations if its `SecurityExpression` value is equal to `1` (TRUE). The `Set` method status NOT_AUTHORIZED (0x01) SHALL be returned to the host on an attempt to set `ReadLocked` or `WriteLocked` columns of the `Locking` table to FALSE for such an LBA range.

4.3.5 ACE Table

The Locking SP Access Control Elements (ACEs) SHALL be modified. Additional rows SHALL be added as specified in Table 11 below for the additional `SecurityExpression` column which was added to the `Locking` table.

Table 11. Locking SP – Additions to SP ACE table

UID	Name	Common Name	Boolean Expr	Row Start	Row End	Col Start	Col End
00 00 00 08 00 00 8E 00	Anybody_Get_SecSens		00 00 00 09 00 00 00 01 (Anybody)			"SecurityExpression"	"SecurityExpression"
00 00 00 08 00 00 8E 01	Anymaster_Set_SecSens		00 00 00 09 00 00 84 03 (BandMasters)			"SensorEnabled"	"EventOffSecureTout"
00 00 00 08 00 00 8E 02	"BandMaster0_SetSecSens"	""	00 00 00 09 00 00 80 01 (BandMaster0)	Null	Null	"SecurityExpression"	"SecurityExpression "
00 00 00 08 00 00 8E 03	"BandMaster1_ SetSecSens"	""	00 00 00 09 00 00 80 02 (BandMaster1)	Null	Null	"SecurityExpression"	"SecurityExpression "
•	•	•	•	•	•	•	•

00 00 00 08 00 00 93 01	"BandMaster1023_SetSecSens"	""	00 00 00 09 00 00 84 00 (BandMaster1023	Null	Null	"SecurityExpression"	"SecurityExpression "

4.3.6 AccessControl Table

The Locking SP access control definitions SHALL be modified if an additional `SecuritySensors` table and an additional `SecurityExpression` column in the `Locking` table is added.

Table 12. Locking SP – Additions to Access Control Table

RowNumber	UID	InvokingID	MethodId	CommonName	ACL	Log	ADDACE ACL	RemoveACE ACL	GetACE ACL
VU	VU	00 00 08 06 00 00 00 01 (Other Ranges Unlocked Sensor)	00 00 00 06 00 00 00 06 (Get)	Anybody_Get_SecuritySensors_Table	00 00 00 08 00 00 8C 05 (Anybody)	None	Null	Null	00 00 00 08 00 00 8C 05 (Anybody)
VU	VU	00 00 08 06 00 00 00 01 (Other Ranges Unlocked Sensor)	00 00 00 06 00 00 00 07 (Set)	AnyMaster_Set_SecuritySensors_Table	00 00 00 08 00 00 8E 01 (AnyMaster_Set_SecSens)	None	Null	Null	00 00 00 08 00 00 8C 05 (Anybody)
VU	VU	00 00 08 06 00 00 00 02 (Host Activity Sensor)	00 00 00 06 00 00 00 06 (Get)	Anybody_Get_SecuritySensors_Table	00 00 00 08 00 00 8C 05 (Anybody)	None	Null	Null	00 00 00 08 00 00 8C 05 (Anybody)

•	VU	VU	•	VU	VU	VU
•	VU	VU	•	VU	VU	VU
•	00 00 08 02 00 00 00 01 (Global Range Locking Object)	00 00 08 02 00 00 04 00 (Band1023_Locking object)	•	00 00 08 02 00 00 00 01 (Global Range Locking Object)	00 00 08 06 00 00 00 00 (SecuritySensors Table)	00 00 08 06 00 00 00 02 (Host Activity Sensor)
•	00 00 00 06 00 00 00 07 (Set)	00 00 00 06 00 00 00 06 (Get)	•	00 00 00 06 00 00 00 06 (Get)	00 00 00 06 00 00 00 08 (Next)	00 00 00 06 00 00 00 07 (Set)
•	BandMaster 0_SetBand-Set-Global_RangeLocking object	Anybody _GetBand-Get-Band1023_Locking object	•	Anybody _GetBand-Get-Global_Range Locking object	Anybody_Next_SecuritySensors_Table	AnyMaster_Set_SecuritySensors_Table
•	00 00 00 08 00 00 88 01 (BandMaster 0_SetBand) , 00 00 00 08 00 00 8E 01 (BandMaster0_ SetSecSens)	00 00 00 08 00 02 00 01 (Anybody _GetBand) , 00 00 00 08 00 00 8E 00 (Anybody_Get_SecSens)	•	00 00 00 08 00 02 00 01 (Anybody _GetBand) , 00 00 00 08 00 00 8E 00 (Anybody_Get_SecSens)	00 00 00 08 00 00 8C 05 (AnyMaster)	00 00 00 08 00 00 8E 01 (AnyMaster_Set_SecSens)
•	None	None	•	None	None	None
•	Null	Null	•	Null	Null	Null
•	Null	Null	•	Null	Null	Null
•	00 00 00 08 00 00 80 01 (BandMaster0)	00 00 00 08 00 00 84 00 (BandMaster1023)	•	00 00 00 08 00 00 80 01 (BandMaster0)	00 00 00 08 00 00 8C 05 (Anybody)	00 00 00 08 00 00 8C 05 (Anybody)

VU	VU	00 00 08 02 00 00 04 00 (Band1023_ Locking object)	00 00 00 06 00 00 00 07 (Set)	BandMaster1023_SetBand-Set-Band1023_Locking object	00 00 00 08 00 00 8C 00 (BandMaster1023_SetBand), 00 00 00 08 00 00 93 00 (BandMaster1023_ SetSecSens)	None	Null	Null	00 00 00 08 00 00 84 00 (BandMaster1023)

4.3.7 Locking States Description

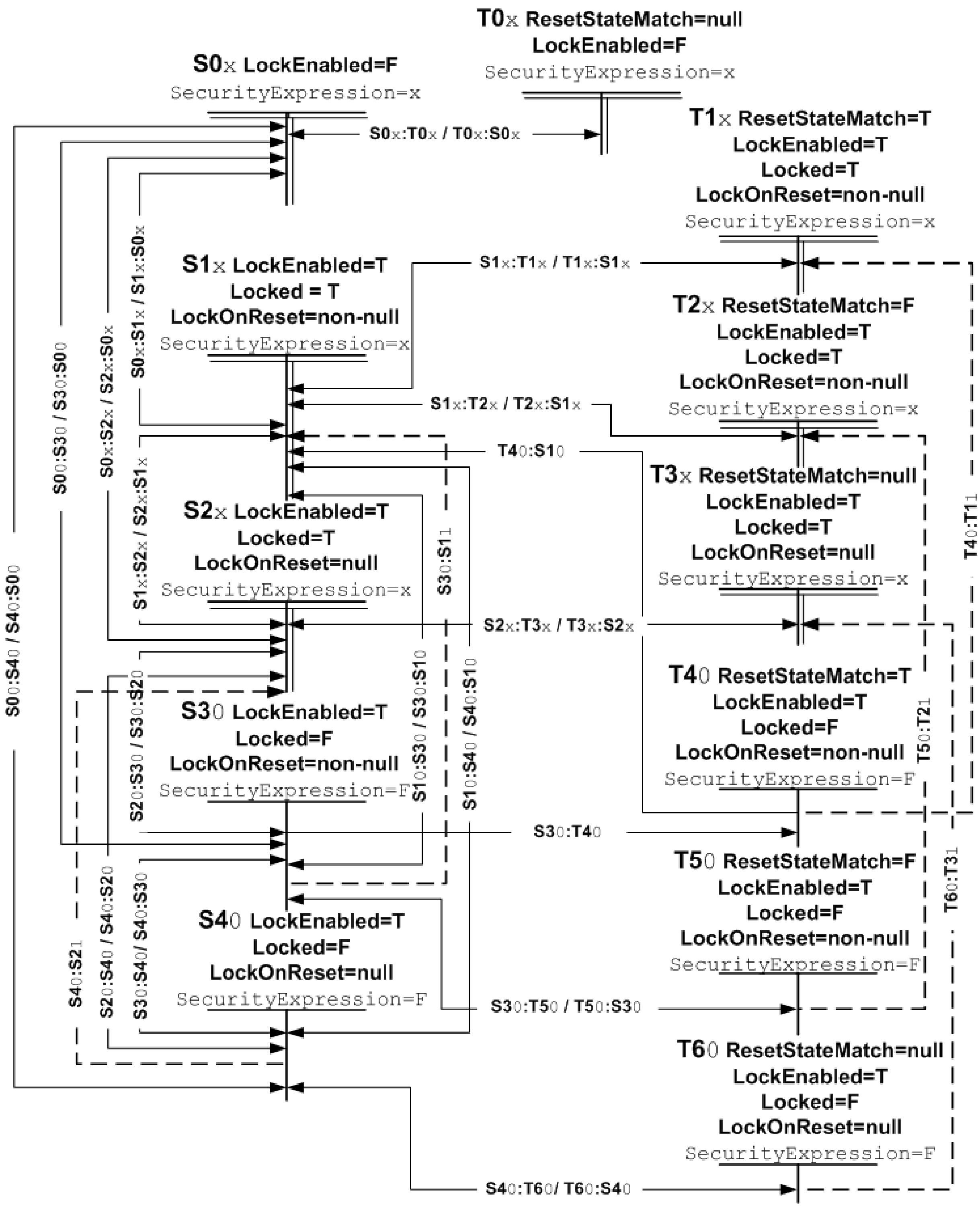
Locking SP behavior SHALL change as described in this chapter. This chapter is based on Locking State Description section of the TCG Storage Architecture Core Specification [2]. All differences are specifically outlined.

4.3.7.1 Locking State Descriptions

Figure 2 displays the states and state transitions for read lock and write lock. The differences are highlighted by distinctive font, and new transactions are represented by dashed lines (detailed description follows). For simplicity the diagram and the accompanying textual information describe the operation of locking in general rather than both read lock and/or write lock in particular.

When a reset is described in these state transitions, "reset" is used generically to refer to qualifying resets as determined by the value of the `LockOnReset` column and the reset behavior associated with particular resets as determined by the appropriate interface-specific description of that reset.

Figure 2 Locking State Transition Diagram



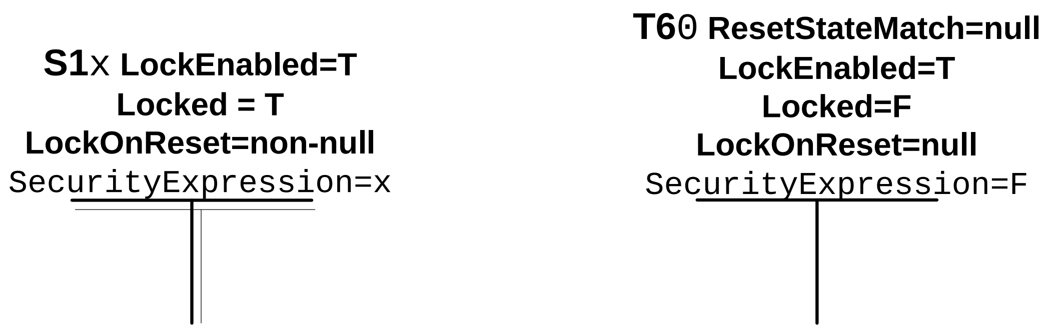
4.3.7.1.1 State Descriptions

This section describes the state changes depicted in the picture above. In parentheses next to each state transition identifier are the values that change to cause that transition.

Comparing to [2], the `SecureExpression` value for this LBA range is added to each locking state. State names are updated accordingly by adding `SecureExpression` value (0 or 1) to the state name. Thus, previous S0 state becomes pair of states S00 (`SecureExpression`=F) and S01 (`SecureExpression`=T). **ZZx** will be used to name such "generic" state with ZZ being S0-S4, T0-T6 and x being `SecureExpression` value: 0 (FALSE) or 1 (TRUE).

In many cases both states of the ZZx pair behave similarly and often presented on the Figure 2 as one entity like S0x or T1x with x being 0 or 1. Double "T lines" on the Figure 2 represent such ZZx pairs rather than single state which are represented by single "T lines" – see Figure 3 (a) and (b) respectively.

Figure 3 Double and single Locking States.



(a) ZZx pair of Locking States (ZZ=S1 in this example)

(a) Single Locking State

Transitions between states in the ZZx pair are not shown on the Figure 2 but described later in the text.

However for some locking states (S30, S40, T40, T50 and T60 when LBA range is not locked) `SecureExpression=T` pair doesn't exist because LBA range cannot be in unlocked state if `SecureExpression=T`.

The Table 13 below contains all Locking State descriptions. Each state description is exactly the same as that of its "ancestor" from Core Specification [2] except the fact that `SecureExpression` value is also specified.

Table 13. Locking States Description

State Common Name	Ancestor State Name from Core Spec [2]	`SecureExpression=F` State Name	`SecureExpression=T` State Name	State Description (applies to both states in the ZZx pair except stuation where ZZ1 state doesn't exist in which case applies to the ZZ0 state only).
S0x	S0	S00	S01	`LockEnabled`**=F** This describes the state where the TPer's Locking feature is turned off. Locking is not possible. The `Locked` column, `LockOnReset` column and `SecurityExpression` column values are disregarded.
S1x	S1	S10	S11	`LockEnabled`**=T/**`Locked`**=T/**`LockOnReset`**=non-null** This describes the state where the TPer's Locking feature is turned on. Locking is possible. The `Locked` state is currently TRUE, indicating that the range is locked. `LockOnReset` is non-null, indicating that, upon any of the listed reset events, the range SHALL lock.

S2x	S2	S20	S21	LockEnabled**=T/**Locked**=T/**LockOnReset**=null** This describes the state where the TPer's Locking feature is turned on. Locking is possible. The Locked state is currently TRUE, indicating that the range is locked. LockOnReset is "FALSE" (null set), indicating that reset events do not cause the range to lock. The range SHALL maintain current locking state (the value of the Locked column remains the same, TRUE) through all resets and Secure Events.
S3x	S3	S30	n/a	LockEnabled**=T/**Locked**=F/**LockOnReset**=non-null** This describes the state where the TPer's Locking feature is turned on. Locking is possible. The Locked state is currently FALSE, indicating that the range is not locked. LockOnReset is "TRUE" (non-null set), indicating that the listed reset events cause the range to lock. Changing SecurityExpression to TRUE will also cause the range being locked.
S4x	S4	S40	n/a	LockEnabled**=T/**Locked**=F/**LockOnReset**=null** This describes the state where the TPer's Locking feature is turned on. Locking is possible. The current Locked state is FALSE, indicating that the range is not locked. LockOnReset is "FALSE" (null set), indicating that reset events do not cause the range to lock. The range SHALL maintain current locking state (FALSE in this case) through all reset events. However changing SecurityExpression to TRUE will cause the range being locked.
T0x	T0	T00	T01	ResetStateMatch**=null/**LockEnabled**=F** This is the transition state where a reset is occurring and the Locking feature is disabled. The Locked column, LockOnReset column and SecurityExpression column values are disregarded.
T1x	T1	T10	T11	ResetStateMatch**=T/**LockEnabled**=T/**Locked**=T/** LockOnReset**=non-null** This describes a transition state where a reset is occurring, and the range had the accompanying attributes - the locking feature is turned on, the range is locked, and the LockOnReset value applies to the currently occurring reset state.
T2x	T2	T20	T21	ResetStateMatch**=F/**LockEnabled**=T/**Locked**=T/** LockOnReset**=non-null** This describes a transition state where a reset is occurring, and the range had the accompanying attributes - the locking feature is turned on, the range is locked, and the LockOnReset value does not apply to the currently occurring reset state. This state is functionally equivalent to T3x.
T3x	T3	T30	T31	ResetStateMatch**=null/**LockEnabled**=T/**Locked**=T/** LockOnReset**=null** This describes a transition state where a reset is occurring, and the range had the accompanying attributes - the locking feature is turned on, the range is locked, and the LockOnReset value is null. This state is functionally equivalent to T2x.
T4x	T4	T40	n/a	ResetStateMatch**=T/**LockEnabled**=T/**Locked**=F/** LockOnReset**=non-null** This describes a transition state where a reset is occurring, and the range had the accompanying attributes - the locking feature is turned on, the range is not locked, and the LockOnReset value applies to the currently occurring reset state.
T5x	T5	T50	n/a	ResetStateMatch**=F/**LockEnabled**=T/**Locked**=F/** LockOnReset**=non-null** This describes a transition state where a reset is occurring, and the range had the accompanying attributes - the locking feature is turned on, the range is not locked, and the LockOnReset value does not apply to the currently occurring reset state. This state is functionally equivalent to T60.
T6x	T6	T60	n/a	ResetStateMatch**=null/**LockEnabled**=T/**Locked**=F/** LockOnReset**=null** This describes a transition state where a reset is occurring and the range had the accompanying attributes - the locking feature is turned on, the range is not locked, and the LockOnReset value is null. This state is functionally equivalent to T50.

4.3.7.1.2 State Change Descriptions

This section describes the state changes depicted in the Figure 2. In parentheses next to each state transition identifier are the values that change to cause that transition. Most transitions are the same as in the TCG Storage Architecture Core Specification [2], those which are different are described below.

SZ0:SZ1 (SecurityExpression changed from FALSE to TRUE)

For ease of reading Figure 2 doesn't contain arrows, corresponding to this transition. If SZ1 state doesn't exist (like for SZ equal to S4 and S5) this transition also doesn't exist. For all other SZx pairs (SZ equal to S0, S1, S2 or S3) SZ0 becomes SZ1 when LBA range `SecurityExpression` column value changes from FALSE (0) to TRUE (1) due to the host changing Security Sensor(s) parameters or `SecurityExpression` itself as well as because of change in Security Sensor(s) Secure Condition(s).

TZ0:TZ1 (SecurityExpression changed from FALSE to TRUE)

For ease of reading Figure 2 doesn't contain arrows, corresponding to this transition. If TZ1 state doesn't exist (like for TZ equal to T4, T5 and T6) this transition also doesn't exist. For all other TZx pairs (TZ equal to T0, T1, T2 or T3) TZ0 becomes ST1 when LBA range `SecurityExpression` column value changes from FALSE (0) to TRUE (1) because of change in Security Sensor(s) Secure Condition(s). The host activity cannot cause this transition because the host interface is in Reset state.

SZ1:SZ0 (SecurityExpression changed from TRUE to FALSE)

For ease of reading Figure 2 doesn't contain arrows, corresponding to this transition. If SZ1 state doesn't exist (like for SZ equal to S4 and S5) this transition also doesn't exist. For all other SZx pairs (SZ equal to S0, S1, S2 or S3) SZ1 becomes SZ0 when LBA range `SecurityExpression` column value changes from TRUE (1) to FALSE (0) because of the host changing Security Sensor(s) parameters or `SecurityExpression` itself as well as because of change in Security Sensor(s) Secure Condition(s).

TZ1:TZ0 (SecurityExpression changed from TRUE to FALSE)

For ease of reading Figure 2 doesn't contain arrows, corresponding to this transition. If TZ1 state doesn't exist (like for TZ equal to T4, T5 and T6) this transition also doesn't exist. For all other SZx pairs (TZ equal to T0, T1, T2 or T3) TZ1 becomes ST0 when LBA range `SecurityExpression` column value changes from TRUE (1) to FALSE (0) because of change in Security Sensor(s) Secure Condition(s). The host activity cannot cause this transition because the host interface is in Reset state.

S30:S11, S40:S21 (SecurityExpression changed from FALSE to TRUE)

The LBA range `SecurityExpression` column value changes from FALSE (0) to TRUE (1) when the host changes Security Sensor(s) parameters or the `SecurityExpression,` as well as changes in Security Sensor(s) Secure Condition(s). The LBA range will be locked.

T40:T11, T50:T21, T60:T31 (SecurityExpression changed from FALSE to TRUE)

The LBA range `SecurityExpression` column value changes from FALSE (0) to TRUE (1) because of changes in Security Sensor(s) Secure Condition(s) change. The host activity cannot cause this transition because the host interface is in Reset state. The LBA range will be locked.

ZZx:YYx

For all other transitions shown on Figure 2, see description of correspondent ZZ:YY transition in the section 5.7.3.1.2 of the TCG Storage Architecture Core Specification [2]. Because `SecurityExpression` value doesn't change (x->x) these transitions are the same as in the Core Spec.

4.4 Additional SPs

This feature set requires no additional SPs.

5 Appendix 1 - Security Sensors Examples

5.1 Overview

This section describes examples of conditions that might be detected by security sensors.

5.2 Tampering Attempt Condition

A SD may have a physical connection (GPIO, I2C, etc…) to a sensor of any type which indicates that an attempt to tamper with TPer's contents may be in progress. Two examples are:

- A SD could be placed into a secure enclosure, which generates a tampering signal each time the secure enclosure's door is opened. This could detect somebody trying to connect his laptop in an attempt to impersonate a valid host and gain access to the SD in unlocked state.
- Remote sensors in a building could provide a tampering attempt signal if any sort of secure perimeter is penetrated (doors opened, alarms tripped, etc…).

5.3 Unsecure Orientation Condition

LBA range(s) may be prevented from being unlocked if SD is in some sort of unnatural position (tilted beyond certain angle for example) or already unlocked range(s) may be locked if SD's position becomes unnatural. A simple accelerometer sensor on SD itself may be able to detect such a situation.

5.4 Outside Secure Area Condition

Some ranges may be allowed to be unlocked only if the SD is located in some sort of Secure Area (building, site, geographical location, etc…) and should be locked if the SD leaves that area. There could be a location sensor of some sort on the host or SD, for example:

- Some sensor constantly receiving an encrypted radio signal on a certain frequency that is only available in particular building(s).
- Some sensor constantly receiving an encrypted radio signal on a certain frequency that is only available when a particular person is nearby. When this person leaves a SD's vicinity (the SD itself doesn't move) certain ranges on the SD may be locked.
- A GPS device, reporting whether the geographical location of the SD is inside or outside the predefined secure area.

5.5 Motion Detection Condition

SDs might always be static or should always be locked while they are being moved. Therefore moving an unlocked SD might indicate that the host or the SD is being attacked.

Precise definitions are beyond the scope of this specification. SDs installed in server racks must not be moved at all while those in laptops should only lock ranges if dropped on the ground, meaning that acceleration and/or speed would be rather high.

5.6 Other Ranges Unlocked Condition

Sometimes it is necessary to ensure that an LBA range is locked whenever any other LBA range (**except those which are unlocked by default**) is unlocked. This is essential for multi-user systems with remote access where not more than one user is allowed to work with the SD at the time. An LBA Range is considered to be **unlocked by default** if its `ReadLockEnabled` is FALSE and `LockOnReset` is empty.

Note that this condition is unlike previous ones as it is purely logical and doesn't require any physical sensors to be installed.

5.7 Host Activity Condition

This condition requires a sensor that detects when there is no host activity (read or write operations) for LBAs in a particular range for more than a predefined Secure Timeout. This type of sensor may be useful for a SD which is connected to a system where several users have access to the SD. If a user left without locking the range, and inactivity is not detected, the range could remain unlocked for an indefinite time and open to attack.

Timeout detection might also be used in more complex attacks, where an attacker tries to keep a SD in an unlocked state by halting the host in an unrecoverable error state ("blue screen") prior to physically tampering with the SD.