TCG Storage
Opal SSC Feature Set:
Additional DataStore Tables

Specification Version 1.00
Revision 1.00

February 24, 2012

Contact: admin@trustedcomputinggroup.org

TCG

TCG PUBLISHED

Disclaimers, Notices, and License Terms

Change History

Version 1.00	Date	Description
Rev 1.00	February 24, 2012	First publication

TABLE OF CONTENTS

Tables

1 Introduction

1.1 Document Purpose

The Storage Workgroup specifications provide a comprehensive architecture for putting Storage Devices under policy control as determined by the trusted platform host, the capabilities of the Storage Device to conform with the policies of the trusted platform, and the lifecycle state of the Storage Device as a Trusted Peripheral.

1.2 Scope and Intended Audience

This specification defines the Additional DataStore Tables for the Opal Security Subsystem Class (SSC). Any Storage Device that claims Opal SSC Additional DataStore Tables compatibility SHALL conform to this specification.

The intended audience for this specification is both trusted Storage Device manufacturers and developers that want to use these Storage Devices in their systems.

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", Version 2.00

[3]. Trusted Computing Group (TCG), "TCG Storage Security Subsystem Class: Opal", Versions 1.00 and 2.00

[4]. Trusted Computing Group (TCG), "TCG Storage Interface Interactions Specification", Version 1.02

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 TCG Storage Security Subsystem Class: Opal (these two documents are at the same level of precedence, and SHALL NOT conflict with each other)
2. Storage Interface Interactions Specification [4]
3. TCG Storage Architecture Core Specification [2]

1.6 Dependencies on Other Feature Sets

This feature set has no dependencies on other feature sets.

1.7 Interactions with Other Feature Sets

At the time of writing, this feature set has no interactions with other feature sets.

1.8 Terminology

This feature does not define any new terms.

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	Opal SSC specified	Fixed	• Cell content is Read-Only. • Access control is fixed. • Value is specified by the Opal 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 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 Additional DataStore Tables Overview

Begin Informative Content

The Additional DataStore Tables feature set defined in this document is an extension of the existing mechanism provided via the DataStore table defined in [3]. This specification extends the DataStore table concept by providing a mechanism allowing the host to partition the space that the TPer allocates for that table into a number of independently managed DataStore tables.

The goal of the Additional DataStore Tables feature set is to provide an ability to store application-specific metadata in the TPer in such a way that:

- It is possible to create multiple DataStore tables independent from each other
- It is possible to configure access control in such a way that multiple entities owning DataStore tables but unaware of each other are prevented from accessing each other's data
- The storage area of the DataStore tables does not overlap with the User Area in order to avoid potential compatibility issues with existing host software

The DataStore tables can, in some implementations, be used to store key material without which C_PIN credentials cannot be obtained. Because of that, corruption of DataStore tables may have the same consequences as corruption of C_PIN or K_AES tables. TPer implementers should keep this in mind while designing redundancy mechanisms for the DataStore tables.

End Informative Content

3 SSC Specific Functionality

This section defines the SSC-specific functionality (not contained in [2], [3], or [4]) required to support the Additional DataStore Tables feature set.

3.1 Modified methods

3.1.1 Activate

The Activate method of the Admin SP is modified to include an optional parameter called DataStoreTableSizes:

```
SPObjectUID.Activate[
        DataStoreTableSizes = list [ uintegers ]
]
=>
[ ]
```

If provided, the DataStoreTableSizes parameter (parameter number 0x060002) specifies the number of the DataStore tables that the TPer is requested to create and their sizes in bytes. The first element in the DataStoreTableSizes list corresponds to the DataStore table defined in [3]. The subsequent elements specify the desired sizes of the additional DataStore tables introduced by the Additional DataStore Tables feature set (refer to section 3.2.1).

The number and the sizes of the DataStore tables created via the Activate method are subject to certain constraints specific to the TPer. The constraints can be discovered via the Level 0 feature descriptor defined in section 4.1.1. If the TPer cannot satisfy a request for additional DataStore tables due to constraints related to the total number or size of the DataStore tables, the Activate method SHALL fail with the INSUFFICIENT_SPACE code. If the TPer cannot satisfy a request for additional DataStore tables due to constraints related to DataStore table size alignment, the Activate method SHALL fail with the INVALID_PARAMETER code.

If the DataStoreTableSizes parameter is not provided, the Activate method shall be processed as if this parameter were provided and consisted of a single element equal to the “Maximum total size of DataStore tables” field of the DataStore Table Feature Descriptor defined in section 4.1.1.

Refer to section 4.2.1 for additional requirements defined for the Activate method by the Additional DataStore Tables feature set.

3.1.2 Reactivate

If the Locking SP supports the Reactivate method, it is modified to include an optional parameter called DataStoreTableSizes:

The Reactivate method of the Locking SP is modified as follows:

```
ThisSP.Reactivate[
        DataStoreTableSizes = list [ uintegers ]
]
=>
[ ]
```

If provided, the DataStoreTableSizes parameter (parameter number 0x060003) specifies the number of the DataStore tables that the TPer is requested to create upon Locking SP reactivation and their sizes in bytes. The first element in the DataStoreTableSizes list corresponds to the DataStore table defined in [3]. The subsequent elements specify the desired sizes of the additional DataStore tables introduced by the Additional DataStore Tables feature set.

The number and the sizes of the DataStore tables created via the Reactivate method are subject to certain constraints specific to the TPer. The constraints can be discovered via the Level 0 feature descriptor defined in

section 4.1.1. If the TPer cannot satisfy a request for additional DataStore tables due to constraints related to the total number or size of the DataStore tables, the Reactivate method SHALL fail with the INSUFFICIENT_SPACE code. If the TPer cannot satisfy a request for additional DataStore tables due to constraints related to DataStore table size alignment, the Reactivate method SHALL fail with the INVALID_PARAMETER code.

If the DataStoreTableSizes parameter is not provided, the Reactivate method shall be processed as if this parameter were provided and consisted of a single element equal to the “Maximum total size of DataStore tables” field of the DataStore Table Feature Descriptor defined in section 4.1.1.

Refer to section 4.3.1 for additional requirements defined for the Reactivate method by the Additional DataStore Tables feature set.

3.2 New tables

3.2.1 DataStoreXXXX

A number of new DataStoreXXXX byte tables are added to the Locking SP. The new DataStoreXXXX tables are functionally equivalent to the DataStore table defined in [3] and can be used by the host to persistently store implementation-specific information using the Set and Get methods on the DataStoreXXXX objects.

4 Feature Set Requirements

This section defines the Mandatory (M) and Optional (O) requirements for the Additional DataStore Tables feature set, when it is implemented in an Opal-compliant device.

4.1 Level 0 Discovery

An Opal-compliant SD that contains the Additional DataStore Tables feature set SHALL return the DataStore Table Feature Descriptor as described in 4.1.1, in addition to the Level 0 Discovery response requirements defined in [3].

4.1.1 DataStore Table Feature Descriptor (Feature Code = 0202h)

This feature descriptor SHALL be returned when the Opal-compliant SD supports the Additional DataStore Tables feature set. The contents of the feature descriptor are defined in Table 2.

Table 2 Level 0 Discovery - DataStore Table Feature Descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1				Feature Code				(LSB)
2				Version				Reserved
3				Length				
4								
5				Reserved				
6	(MSB)							
7				Maximum number of DataStore tables				(LSB)
8	(MSB)							
9								
10				Maximum total size of DataStore tables				
11								(LSB)
12	(MSB)							
13								
14				DataStore table size alignment				
15								(LSB)

4.1.1.1 Maximum number of DataStore tables

This field specifies the maximum number of the DataStore tables that the TPer supports, including the DataStore table defined in [3].

4.1.1.2 Maximum total size of DataStore tables

This field specifies the maximum total size in bytes of all of the DataStore tables that the TPer supports, including the DataStore table defined in [3].

4.1.1.3 DataStore table size alignment

This field specifies the size alignment in bytes for the DataStore tables other than the DataStore table defined in [3].

4.1.1.4 Level 0 requirements for the DataStore Table Feature Descriptor

- **Feature Code**: 0x0202
- **Version**: 0x1 or any version that supports the defined features in this specification
- **Length**: 0x0C

- **Maximum number of DataStore tables:** 1 or above.
- **Maximum total size of DataStore tables:** 0xA0000 or above.
- **DataStore table size alignment:** 1 or above

4.2 Admin SP

An Opal-compliant SD that contains the Additional DataStore Tables feature set SHALL contain the additions to the Admin SP specified in this section, in addition to the Admin SP requirements defined in [3].

4.2.1 Activate method

An Opal-compliant SD that contains the Additional DataStore Tables feature set supports the modifications to the Activate method defined in section 3.1.1.

The Activate method supports any DataStoreTableSizes parameter value that meets all of the following criteria:

- The number of elements in the DataStoreTableSizes parameter is less than or equal to the value of the “Maximum number of DataStore tables” field of the DataStore Table Feature Descriptor.
- The total sum of all elements in the DataStoreTableSizes parameter less than or equal to the value of the “Maximum total size of DataStore tables” field of the DataStore Table Feature Descriptor.
- The value of each element in the DataStoreTableSizes parameter is a multiple of the value of the “DataStore table size alignment” field of the DataStore Table Feature Descriptor.

If the Activate method succeeds, the TPer creates DataStore tables in the Locking SP as described in section 3.2.1 and ACE/AccessControl elements for those tables as described in sections 4.3.3 and 4.3.4. The sizes of the DataStore tables can be retrieved from the Rows column of the Table table after Locking SP activation.

4.3 Locking SP

An Opal-compliant SD that contains the Additional DataStore Tables feature set SHALL contain the additions to the Locking SP specified in this section, in addition to the Locking SP requirements defined in [3].

4.3.1 Reactivate method

An Opal-compliant SD that contains the Additional DataStore Tables feature set and implements the Reactivate method supports the modifications to the Reactivate method defined in section 3.1.2.

The Reactivate method supports any DataStoreTableSizes parameter value that meets all of the following criteria:

- The number of elements in the DataStoreTableSizes parameter is less than or equal to the value of the “Maximum number of DataStore tables” field of the DataStore Table Feature Descriptor.
- The total sum of all elements in the DataStoreTableSizes parameter less than or equal to the value of the “Maximum total size of DataStore tables” field of the DataStore Table Feature Descriptor.
- The value of each element in the DataStoreTableSizes parameter is a multiple of the value of the “DataStore table size alignment” field of the DataStore Table Feature Descriptor.

If the Reactivate method succeeds, the TPer replaces existing DataStore tables in the Locking SP with new DataStore tables as described in section 3.2.1. The content of the DataStore tables is not guaranteed to be preserved across a Reactivate method invocation even if the DataStoreTableSizes list provided in it is the same as in the previous Activate or Reactivate method invocation.

The existing ACE/AccessControl elements for the DataStore tables are replaced with new ACE/AccessControl elements as described in sections 4.3.3 and 4.3.4. This results in the BooleanExpr column of the ACEs related to the DataStore tables being reset to “Admins”. The sizes of the DataStore tables can be retrieved from the Rows column of the Table table after Locking SP activation.

4.3.2 Table table

A number of new entries are added to the Table table for the additional DataStore tables, resulting in the following changes in the Table table preconfiguration of the Locking SP:

*TT1 = The number of rows in the DataStore tables is specified via the Activate / Reactivate method and subject to the constraints defined in 4.1.1

*TT2 = The number of the DataStore tables is specified via the Activate / Reactivate method and subject to the constraints defined in 4.1.1

Table 3 Locking SP – Additions to Table Table Preconfiguration

UID	Name	CommonName	TemplateID	Kind	Column	NumColumns	Rows	RowsFree	RowBytes	LastID	MinSize	MaxSize
00 00 00 01 00 00 10 02	"DataStore2"			Byte			*TT1					
00 00 00 01 00 00 10 00 (+XXXX) *TT2	DataStoreXXXX			Byte			*TT1					

4.3.3 AccessControl table

A number of new entries are added to the AccessControl table for the additional DataStore tables, resulting in the following changes in the AccessControl table preconfiguration of the Locking SP:

*ACT = The number of DataStore related AccessControl entries depends on the number of DataStore tables, which is specified via the Activate / Reactivate method and subject to constraints defined in 4.1.1

Table 4 Locking SP - AccessControl Table Preconfiguration

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
ACE																
		00 00 00 08 00 03 FC 02 *ACT	ACE_DataStore2_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 00 08 00 03 FC 03 *ACT	ACE_DataStore2_Set_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 FC 00 (+ (XXXX – 1) * 2) *ACT	ACE_DataStoreXXXX_Get_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
		00 00 00 08 00 03 FC 01 (+ (XXXX – 1) * 2) *ACT	ACE_DataStoreXXXX_Set_All	Set		ACE_ACE_Set_BooleanExpression				ACE_Anybody						
DataStore																
		00 00 10 02 00 00 00 00 *ACT	DataStore2	Get		ACE_DataStore2_Get_All				ACE_Anybody						

Table Association - informative oly	UID	InvokingID	InvokingID Name - informative only	MethodID	CommonName	ACL	Log	AddACEACL	RemoveACEACL	GetACLACL	DeleteMethodACL	AddACELog	RemoveACELog	GetACLLog	DeleteMethodLog	LogTo
		00 00 10 02 00 00 00 00 *ACT	DataStore2	Set		ACE_DataStore2_Set_All				ACE_Anybody						
		00 00 10 00 00 00 00 00 (+XXXX << 32) *ACT	DataStoreXXXX	Get		ACE_DataStoreXXXX_Get_All				ACE_Anybody						
		00 00 10 00 00 00 00 00 (+XXXX << 32) *ACT	DataStoreXXXX	Set		ACE_DataStoreXXXX_Set_All				ACE_Anybody						

4.3.4 ACE table

A number of new ACEs are added for the additional DataStore tables, resulting in the following changes in the ACE table preconfiguration of the Locking SP:

*ACET = The number of DataStore related ACEs depends on the number of DataStore tables, which is specified via the Activate / Reactivate method and subject to constraints defined in 4.1.1

Table 5 Locking SP - ACE Table Preconfiguration

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	00 00 00 08 00 03 FC 02 *ACET	"ACE_DataStore2_Get_All"		Admins	All

Table Association -Informative Column	UID	Name	CommonName	BooleanExpr	Columns
	00 00 00 08 00 03 FC 03 *ACET	"ACE_DataStore2_Set_All"		Admins	All
	00 00 00 08 00 03 FC 00 (+ (XXXX – 1) * 2) *ACET	"ACE_DataStoreXXXX_Get_All"		Admins	All
	00 00 00 08 00 03 FC 01 (+ (XXXX – 1) * 2) *ACET	"ACE_DataStoreXXXX_Set_All"		Admins	All

4.4 Additional SPs

This feature set requires no additional SPs.