Encrypted Hard Drive Device Guide

September 13, 2011

Abstract

This paper provides information about Trusted Computing Group (TCG) protocol support in Windows® operating systems. It provides guidelines for storage device vendors to implement the subset of TCG protocol methods available in Windows.

This information applies to the following operating systems:  
 Windows Developer Preview

References and resources discussed here are listed at the end of this paper.

The current version of this paper is maintained on the Web at:   
 [Encrypted Hard Drive Device Guide](http://go.microsoft.com/fwlink/p/?LinkId=227287)

**Disclaimer**: This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet website references, may change without notice. Some information relates to pre-released product which may be substantially modified before it’s commercially released. Microsoft makes no warranties, express or implied, with respect to the information provided here. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2011 Microsoft. All rights reserved.



Contents

[Introduction 3](#_Toc302379623)

[Architecture 3](#_Toc302379624)

[Interface Support 4](#_Toc302379625)

[IEEE 1667 Support 4](#_Toc302379626)

[TCG Support 4](#_Toc302379627)

[TCG Data Stream Encoding 8](#_Toc302379628)

# Introduction

The purpose of this device guide is to speed up device development and prototyping by listing the subset of TCG methods and data that Windows uses. It is recommended that device vendors implement all the mandatory features in the Opal specification in their final product. This assures future compatibility and interoperability with other platforms.

In addition to the [TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935) and the [OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376), the device must also support the following feature sets:

* <https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php/14895/TCG_Storage-Opal_Feature_Set-Additional_DataStore_Tables_v1_00_r1_05-clean.00016.001.2011.02.10.pdf>
* <https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php/15888/TCG_Storage-Opal_Feature_Set_Single_User_Mode_1-00_1-04_20110706.00004.004.2010.09.23.pdf>

Encrypted Hard Drive is built on top of the [Enhanced Storage](http://msdn.microsoft.com/en-us/library/ff576682(v=VS.85).aspx) framework and provides the platform support for these new device types. For the device to seamlessly integrate and work with the next version of Windows (code-named Windows 8), it must meet the minimum requirements described in this document.

# Architecture

# Diagram showing the TCG Protocol Encapsulation

Figure 1: TCG Protocol Encapsulation

Windows supports only the following device configuration:

* SSC type must be Opal.
* The device must have a LockingSp.
* The device must support a minimum of eight bands including the global band.
* The maximum number of DataStore tables must be at least the number of bands supported by the device.
* The maximum total size of DataStore tables must be at least 128 Kbytes\*NumberOfBands.
* The maximum total size of DataStore tables must be aligned with the Minimum DataStore table size alignment.

# Interface Support

* The device must support interface security commands such as ATA (T13) TRUSTED SEND/TRUSTED RECEIVE or SCSI (T10) SECURITY PROTOCOL OUT/SECURITY PROTOCOL IN [[TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376) section 2.3.1]. Windows communicates internally only in SCSI. Windows translates SCSI to ATA depending on the interface type that the system supports.

# IEEE 1667 Support

* [IEEE 1667 Protocol Probing](http://standards.ieee.org/findstds/standard/1667-2009.html):
  + Must support ZeroLengthSecurityProtocolIn with INC\_512 either set or cleared.
  + Must support ProtocolProbe (Protocol=0, Silo=0, Command=0) and return SECURITY\_PROTOCOL\_IEEE1667 (0xee) as one of the supported protocols.
* [IEEE 1667 Probe Silo Support](http://standards.ieee.org/findstds/standard/1667-2009.html):
  + Must support the PROBE\_COMMAND (0x01) to the Probe Silo (0x00) using SECURITY\_PROTOCOL\_IEEE1667 (0xee) and return a Silo Command Status of either SUCCESS (0x00) or DEFAULT\_BEHAVIOR (0x01) with the probe data of the STID array that contains STID\_PROBE (0x100) and STID\_TCG (0x104).
* [IEEE 1667 TCG Silo Support](http://www.ieee1667.com/index.php?option=com_docman&task=doc_view&gid=363&tmpl=component&format=raw&Itemid=5):
  + Get Silo Capabilities Command (0x01)
  + Transfer Command (0x02)
  + Reset Command (0x03)
  + Get Transfer Results Command (0x04)
  + TPer Reset Command (0x05)

# TCG Support

Windows 8 supports only the Opal SSC. The following key features of the TCG Storage Architecture Core Specification and TCG Opal SSC Specification must be implemented by the device:

* Security Protocol Discovery (TcgProtocolId=0, TcgComId=0): Must support protocols 0, 1, and 2.
* Programmatic TPer Reset (TcgProtocolId=2, TcgComId=4): This command is used by Windows and system firmware to reset the TPer so that the protected ranges will be locked on operating system crashes, system sleep, or system reboot.
* ComID: Windows communicates with the device through IEEE1667 protocol. Therefore, it uses the ComID returned by the IEEE1667 TCG Silo command Get Silo Capabilities.
* Level 0 Discovery [[OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376) section 3.1.1]: Through the IEEE 1667 Get Silo Capabilities command (or native TCG command), Windows performs level 0 discovery. At the minimum, the device must support the following Feature Codes:
  + Locking Feature (0x0002) [[OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376) section 3.1.1.3]
  + Opal SSC Feature (0x0200) [[OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376) section 3.1.1.4]
  + [Single User Fixed ACL Feature](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php/15888/TCG_Storage-Opal_Feature_Set_Single_User_Mode_1-00_1-04_20110706.00004.004.2010.09.23.pdf) (0x0201)
  + [DataStore Table Feature](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php/14895/TCG_Storage-Opal_Feature_Set-Additional_DataStore_Tables_v1_00_r1_05-clean.00016.001.2011.02.10.pdf%20section%202.5) (0x0202)
* Device State Discovery: Windows invokes the following methods to determine the state of the device:
  + UID\_LOCKINGSP\_OPAL.Get() method (0x0000020500000002.0000000600000016): Determines the locking state of the device.
  + UID\_SID\_AUTHORITY.Get() method (0x0000000900000006.0000000600000016): Determines the SID authority state.
  + UID\_LOCKING\_INFO\_ROW.Get() method (0x0000080100000001.0000000600000016): Retrieves locking information. The device must support the additional columns required by the [Single User Mode Fixed Access Configuration Proposal](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php/15888/TCG_Storage-Opal_Feature_Set_Single_User_Mode_1-00_1-04_20110706.00004.004.2010.09.23.pdf) section 3.9.
  + UID\_SECRET\_PROTECT\_KAES128\_KEY.Get() method (0x0000001D0000001D.0000000600000016) or UID\_SECRET\_PROTECT\_KAES256\_KEY.Get() method (0x0000001D0000001E.0000000600000016): Determines the secret protection mechanism. Windows invokes one of these methods according to the ActiveKey (0x0A) column of the Locking Table.
* Transactions [[TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935)section 3.3.7.3]: Some band management operations involve issuing multiple commands to the device. If the operation is stopped before completion, the device may be in an inconsistent state. To prevent this type of scenario, Windows issues StartTransaction and EndTransaction commands around the group of commands to be protected as an atomic transaction. If any failure occurs with any of the commands, the transaction stops with an Abort and all the previously successful commands are undone.
* Session Manager Properties Method [[OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376) section 4.1.1.1, [TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935) section 5.2.2]: Windows invokes this method to determine the ComPacket sizes. Windows recommends both MaxComPacketSize and MaxResponseComPacketSize to be at least 16 Kbytes. ComPacket sizes smaller than 16 Kbytes severely impact boot performance. Windows also invokes this method to set the HostProperties such as MaxComPacketSize and MaxResponseComPacketSize for subsequent communications with the device.
* Session Manager Start Session Method [Spec: [TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935) section 5.2.3]: When Windows must communicate with the device, it invokes this method to open a session. The SPSessionID returned by SMUID.SyncSession is used for subsequent communications with the device within that session.
* Session Manager Close Session Method [Spec: [TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935) section 5.2.3.5]: After performing a band management operation, Windows always ends the session by sending an EndOfSession token.
* Base Template Authenticate Method [[TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935)section 5.3.3.12]: Windows invokes this method to authenticate the current AdminSP or LockingSP session with appropriate authority and corresponding pin. The authorities can be one of the following. Note that because Windows may disable SID authority after activating the security of the device, it is important to implement the PSID authority. It is the only way to revert the device in that scenario.
  + SID\_AUTHORITY (0x0000000900000006)
  + PSID\_AUTHORITY (0x000000090001FF01)
  + ADMIN\_AUTHORITY (0x0000000900010001)
  + USER\_AUTHORITY (0x0000000900030001 + n)
* Base Template C\_PIN Objects [[TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935)section 5.3.2.12]: Windows invokes the following methods to get or set the pins of various authorities:
  + UID\_MSID\_PIN.Get() method (0x0000000B00008402.0000000600000016): Retrieves the MSID pin.
  + CPinObject.Set() method: Sets the pin of a CPinObject. CPinObject can be one of the following:
    - UID\_SID\_PIN (0x0000000B00000001)
    - UID\_ADMIN\_PIN (0x0000000B00010001)
    - UID\_USER\_PIN (0x0000000B00030001 + n)
* Get band state or information: Windows invokes the following methods to get information about a band:
  + UID\_LOCKING\_ROW\_0.Get() method (0x0000080200000001.0000000600000016): Retrieves information for the global band.
  + UID\_LOCKKING\_ROW\_n.Get() method (0x0000080200030000+n.0000000600000016): Retrieves information for band *n*.
  + ActiveKeyUID.Get() method: Retrieves media key info. The UID for ActiveKey is obtained from column 10 of the LockingRow table.
  + UID\_USER\_AUTH\_n.Get() method (0x0000000900030001+n.0000000600000016): Retrieves the band name from the COMMON\_NAME column of the USER\_AUTH table.
  + UID\_DATASTORE\_BASE+n.Get() method (0x0000100100000000+(n<<32).0000000600000016): Retrieves band metadata from the data store.
* Set band state or information: Windows invokes the following methods to set the state or other information of a band:
  + UID\_LOCKING\_ROW\_0.Set() method (0x0000080200000001.0000000600000017): Sets information for the global band.
  + UID\_LOCKKING\_ROW\_n.Set() method (0x0000080200030000+n.0000000600000017): Sets information for band *n*.
  + UID\_USER\_AUTH\_n.Set() method (0x0000000900030001+n.0000000600000017): Stores the band name to the COMMON\_NAME column of the USER\_AUTH table.
  + UID\_DATASTORE\_BASE+n.Set() method (0x0000100100000000+(n<<32).0000000600000017): Stores band metadata to the data store.
  + UID\_LOCKING\_ROW+n.Erase() method (0x0000080200030000+n.0000000600000803): Erases a band.
* Device Provisioning: Windows invokes the following methods to provision or un-provision the device:
  + UID\_LOCKING\_SP\_OPAL.Activate() method (0x0000020500000002.0000000600000203): Activates security on the device. Windows invokes this method with the following parameters [[Single User Mode Fixed Access Configuration Proposal](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php/15888/TCG_Storage-Opal_Feature_Set_Single_User_Mode_1-00_1-04_20110706.00004.004.2010.09.23.pdf), section 2.2.1; Additional DataStore Table Proposal, section 2.1]:
    - SingleUserModeSelectionList (0x060000) = 0x0000080200000000 (EntireLockingTable)
    - RangeStartRangeLengthPolicy (0x060001) = 0 (User authority has sole ownership)
    - DataStoreTableSizes (0x060002) = List of per band DataStoreTable sizes
  + UID\_ADMIN\_SP.Revert() method (0x0000020500000001.0000000600000202) or UID\_LOCKING\_SP\_OPAL.Revert() method (0x0000020500000002.0000000600000202): Reverts security on the device to return it to the original factory state.
  + UID\_TPERINFO\_OBJECT.Set() method (0x0000020100030001.0000000600000017): Enables or disables TPer Reset.
  + UID\_SID\_AUTH.Set() method (0x0000000900000006.0000000600000017): Enables or disables SID authority.
  + UID\_ACE\_LOCKING\_RANGE\_ERASE+n.Set() method (0x0000000800043000+n.0000000600000017): Sets the authorities that can erase bands.
  + UID\_ACE\_DATASTORE\_GETALL+n.Set() method (0x000000080003FC00+n.0000000600000017): Sets the authorities that can read from the data stores.
  + UID\_ACE\_DATASTORE\_SETALL+n.Set() method (0x000000080003FC01+n.0000000600000017): Sets the authorities that can write to the data stores.
  + UID\_ADMIN\_AUTH.Set() method (0x0000000900010001.0000000600000017): Enables or disables ADMIN\_AUTHORITY.
* Resets
  + TPer Reset (SiloCommand=5 or TcgProtocolID=2, TcgComID=4) [Programmatic TPer Reset [OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376), section 3.2.3 and [IEEE 1667 P35](http://www.ieee1667.com/index.php?option=com_docman&task=doc_view&gid=363&tmpl=component&format=raw&Itemid=5) section A.X.7]: This command is used by Windows or BIOS to reset the TPer so that the protected ranges will be locked on operating system crashes or system reboots.
  + Stack Reset (SiloCommand=3 or TcgProtocolID=2, TcgComID=SessionCmdID) [[OPAL SSC 2.10](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=15376) section 3.2.2 and [IEEE 1667 P35](http://www.ieee1667.com/index.php?option=com_docman&task=doc_view&gid=363&tmpl=component&format=raw&Itemid=5) section A.X.5]: This command is used by Windows to reset the device when encountering device I/O errors or any unexpected errors.

# TCG Data Stream Encoding

As described by the [TCG Core Spec 2.0](https://members.trustedcomputinggroup.org/apps/org/workgroup/storagewg/download.php?document_id=14935), section 3.2.2, data tokens can be encoded with various lengths. If a method returns data tokens, Windows expects these tokens to be encoded using either the minimum encoding format or an encoding format that does not exceed the expected length of the data type. For example, Windows expects Boolean data to be 1 byte in length. Therefore, the device should encode a Boolean data token using the Tiny Atom encoding. Other encodings can also work as long as the length of the data is 1 byte. If the data type is a 32-bit DWORD, any encoding works as long as the length of the data is 4 bytes or less. The TCG parser implemented in Windows can parse data tokens with the following restrictions:

* Integer data type is limited to 8 bytes.
* Byte sequence length is limited only by the length of the ComPacket.