Universal Serial Bus

Content Security Method 2

USB Digital Transmission Content Protection Implementation

INTEL CORPORATION

USB 1.0 Release Candidate

Revision 0.9

January 25, 2000

Revision History

Revision	Date	Filename	Author	Description
.9	01/25/2000			Promotion to .9 at USB DWG.
.8b	12/23/1999	Csm2_v0_8b		Adjust to changes in CS class specification. Get_channel_setting, notification service,
.8a	11/9/1999	Csm2_v0_8a		Add requests to support transport of encrypted data over control endpoint.
.8	11/01/1999	Csm2_v0_8		Promoted to .8 at 10/22/1999 USB DWG. Corrected LByte of wValue of all requests to have bMethod value as denoted in Devices CS channel Descriptor.
.7	09/27/1999	csm2_v0_7		Separated CSM Appendices into individual CSM specification per Sept 1999 CSWG meeting

Contributors

Michael Andre	Intel
John Howard	Intel
Steve McGowan	Intel

Please send comments via electronic mail to michael.andre@intel.com

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Purpose

This paper describes the USB transport services and protocol formats that support Digital Transmission Content Protection (DTCP). Use of DTCP requires licensing by the Digital Transmission Licensing Administrator (DTLA). The details of this licensing can be found at www.dtcp.com.

1.2 Scope

USB CSM-2 describes the USB transport services, descriptors, and requests necessary to support DTCP protocols over USB. This document does not change or alter DTCP functionality.

The Content Security Class (CSC) specification allows Content Security Methods (CSM) to define additional requests as needed. CSM-2 defines additional USB CSC requests in order to support DTCP AKE protocols between USB Host and Device. In addition, CSM-2 implements the Content Security Notification Service and defines additional notifications that are needed to support DTCP protocols.

1.3 Related Documents

- Digital Transmission Content Protection Specification Volume 1 Revision 1.0, February 18, 1990
  - Appendix A, USB DTCP Specification
- Universal Serial Bus Device Class Definition for Content Security Devices
- Universal Serial Bus Specification Version 1.1
- USB Common Class Specification Version 1.0

1.4 Terms and Abbreviations

AKE	**Authentication and Key Exchange**
CCI	**Copy Control Information**
CS	**Content Security, USB terminology for Content Protection**
CSC	**Content Security Class, refers to USB Device Class Definition for Content Security Devices specification**
CSI	**Content Security Interface**
CSM	**Content Security Method**
DTCP	**Digital Transmission Content Protection**
DTLA	**Digital Transmission Licensing Administrator**
CSNS	**Content Security Notification Service**
USB	**Universal Serial Bus**

2 CSM-2 Content Security Class Additions

The USB Device Class Definition For Content Security Devices (CSC) allows Content Security Methods to define additional services as needed. DTCP requires four additional USB Requests to transfer the AKE commands and responses. The CS Notification Service (CSNS) is used to allow USB devices to initiate DTCP AKE protocols.

2.1 AKE USB Requests

DTCP requires four additional USB requests to transfer the AKE command frames rather than defining a unique USB request for each individual AKE Command and corresponding response. There are two additional requests that provide for the transport of encrypted data over the control endpoint. This section details the structure of these requests. The General Request format for AKE Command Response request is as follows:

Table 2-1 AKE General Request Format

Offset	Field	Size	Value	Description
0	*bmRequestType*	1	Bitmap	Characteristics of request: D7: Data transfer direction 0 = Host-to-device 1 = Device-to-host D6...5: Type 1 = Class D4...0: Recipient 1 = Interface
1	*bRequest*	1	Value	CSM-2 Requests ***PUT_COMMAND,GET_RESPONSE*** ***GET_COMMAND, PUT_RESPONSE*** ***PUT_DATA, GET_DATA***
2	*wValue*	2	Value	HByte: 0, Reserved LByte: 0x02 - CSM-2
4	*wIndex*	2	Value	HByte: Channel ID. LByte: CSI Interface number.
6	*wLength*	2	Count	Byte length of the AKE Command or Response Frame.

2.1.1 Command and Response Requests Format

The requests are paired together, one pair is used to send AKE commands to the Device and return the associated response. The other pair is used to retrieve an AKE command from the Device and send the associated response.

Table 2-2 AKE Command Response Pairing

Command	Associated Response
PUT_COMMAND	***GET_RESPONSE***
GET_COMMAND	***PUT_RESPONSE***
PUT_DATA	***GET_DATA***

2.1.1.1 Command Requests

There are two Command requests, ***GET_COMMAND*** and ***PUT_COMMAND***.

The ***GET_COMMAND*** is used to transfer an AKE command from the Device to the Host.

Table 2-3 *GET_COMMAND* Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
1 01 00001B	***GET_COMMAND*** (0x80)	HByte – 0x00 Reserved LByte: 0x02 CSM-2	HByte: Channel ID LByte: CSI Interface Number	Byte Length of USB AKE Command	DTCP AKE Commands

PUT_COMMAND is used to send an AKE command from the Host to the Device.

Table 2-4 *PUT_COMMAND* Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0 01 00001B	***PUT_COMMAND*** (0x81)	HByte: 0x00 Reserved LByte: 0x02 CSM-2	HByte: Channel ID LByte: CSI Interface Number	Byte Length of Data	DTCP AKE Commands

2.1.1.2 Response Requests

There are two Response requests ***GET_RESPONSE*** and ***PUT_RESPONSE***. Response Requests are used to transport the AKE response frame.

The ***GET_RESPONSE*** is used to transfer the response to an AKE command from the Device to the Host.

Table 2-5 *GET_RESPONSE* Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
1 01 00001B	***GET_RESPONSE*** (0x82)	HByte: 0x00 Reserved LByte: 0x02 CSM-2	HByte: Channel ID LByte: CSI Interface Number	Byte Length AKE Response	AKE Response

The ***PUT_RESPONSE*** is used to transfer the response to an AKE command from the Host to the Device.

Table 2-6 *PUT_RESPONSE* Requests

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0 01 00001B	***PUT_RESPONSE*** (0x83)	HByte – 0x00 Reserved LByte: 0x02 CSM-2	HByte: Channel ID LByte: CSI Interface Number	Byte Length of AKE Response	AKE Response

2.1.1.3 Data Requests

There are two Data requests ***GET_DATA*** and ***PUT_DATA***. ***GET_DATA*** is used to transport data from the device to the host.

Table 2-7 *GET_DATA* Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
1 01 00001B	***GET_DATA*** (0x84)	HByte: 0x00 Reserved LByte: 0x02 CSM-2	HByte: Channel ID LByte: CSI Interface Number	Byte Length	

The ***PUT_DATA*** is used to transport Data form the host to the device.

Table 2-8 *PUT_DATA* Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0 01 00001B	***PUT_DATA*** (0x85)	HByte: 0x00 Reserved LByte: 0x02 CSM-2	Channel ID CSI Interface Number	Byte Length	

2.2 Content Security Notification Service (CSNS)

CSM-2 compliant devices will implement the CS notification service, support the **CHANGE_CHANNEL_SETTINGS** notification, and support the CSM-2 notifications defined in this section.

The CSM-2 CSNS allows the USB Device to send AKE commands, responses, and data as needed via the CSM-2 requests: **GET_COMMAND**, **GET_RESPONSE**, and **GET_DATA**. The CSM-2 host driver upon receiving a CSM-2 notification will issue the corresponding request to the device. The CSNS is started once a CS channel is established that links CSM-2 to an interface or endpoint via the **SET_CHANNEL_SETTINGS** request.

The CSC specification defines a general format for CSM notifications returned by the USB Device. CSM-2 notification format does not require a data field at offset three as described in CSC specification. The CSM-2 format is as follows:

Table 2-9 CSM-2 Notification Format

Offset	Field	Size	Value	Description
0	*bLength*	1	0x03	Byte length of this descriptor.
1	*bChannel*	1	SBD	Channel ID of CSM that generated the notification.
2	*bNotification*	1	Number	00_{16} – $7F_{16}$ = Set by CS specification. 80_{16} = ***Send_GET_COMMAND*** Request 81_{16} = ***Send_GET_RESPONSE*** Request 82_{16} = ***Send_GET_DATA*** Request 83_{16} – FF_{16} = Reserved

Note, USB Interrupt IN service is somewhat of a misnomer; it is implemented such that the Host periodically polls the USB Device. This provides the Device with an opportunity to send a notification to the Host. Recall that USB is designed so that the Host has total control of the USB.

2.3 CSM-2 Descriptors

This section describes information relevant to the CSM-2 instantiation and use of CSC descriptors. Each subsection corresponds to a CSC descriptor and only values pertinent to CSM-2 are listed in each subsection. Note, some subsections may not have any data and therefore the definition and use of the descriptor as specified in CSC is sufficient.

2.3.1 Device Descriptor

No additional definition needed.

2.3.2 Configuration Descriptor

No additional definition needed.

2.3.3 Content Security Interface Descriptor

No additional definition needed.

2.3.4 Content Security Method Descriptor

The **bMethodID** has a value of 0x02.

The **bcdVersion** field has a value of 0x0100

The CSMData Field is not used.

2.3.4.1 CSM-2 String Descriptor

Table 2-10 String Descriptor

Field	Size	Value	Description
bLength	1	Number	Byte length of this descriptor.
bDescriptorType	1	0x03	Specified by Table 9-5 of USB 1.1
bString	0x34	ASCII	The value of this field is as follows and contained within the square brackets **[Digital Transmission Content Protection Version 1.00]**

2.3.5 Content Security Method Variant Descriptor

Not used by CSM-2.

3 DTCP AKE Packet Formats

3.1 Control Packet Format

The Control Packet is used to exchange DTCP control frames between Host and USB Device via the default control pipe using the CSM Get and Put Requests.

	Msb							Lsb
Control[0]	C/R	Reserved(Zero)			Ctype			
Control[1]	AKE Control Data							
Control[2]								
Control[3]								
Control[4]								
Control[5]								
Control[6]								
Control[7]								
Control[8]	Byte Length N of AKE_Info Field							
Control[9]								
AKE_Info[1]	AKE_Info							
-								
AKE_Info[N]								

Figure 3-1 CSM-2 Control Packet Format

The contents and structure of the AKE Control Data and AKE_Info fields are detailed in DTCP specification appendix A.

3.2 Status Packet Format

The Status Packet is used to query and determine DTCP status and state.

	msb							Lsb
Control[0]	C/R	Reserved(Zero)			Ctype			
Control[1]	AKE Control Data							
Control[2]								
Control[3]								
Control[4]								
Control[5]								
Control[6]								
Control[7]								

Figure 3-2 Status Packet Format

The contents and structure of the AKE Control Data and AKE_Info fields are detailed in DTCP specification appendix A.

4 CSM-2 Protected Content Header

This header is used to transfer content protected data over the USB data transport pipe of the associated audio or video class and provides the functionality described in subsections of section 6 of DTCP specification. The header format is defined in Appendix A of the DTCP specification.

	msb							Lsb
Header[0]								
Header[1]								
PC[0]	Protected Content							
-								
-								
-								
PC[N]								

Figure 4-1 Protected Content Packet

Appendix A. CSM-2 Specific Request Codes

A.1 CSM-2 Specific Request Codes

Table A-1: CSM-2 Specific Request Codes

Request Code	Value
Get_Command	0x80
Put_Command	0x81
Get_Response	0x82
Put_Response	0x83
Get_Data	0x84
Put_Data	0x85
Reserved	0x86..0xFF

A.2 CSM-2 Notification Values

Table A-2: CSM-2 Notification Values

bNotification	Value
Send_Get_Command	0x80
Send_Get_Response	0x81
Send_Get_Data	0x82
Reserved	0x83..0xFF