Universal Serial Bus
Device Class Definition
for
Content Security Devices

Content Security Framework

Release 2.0

June 12, 2012

Scope of This Release

This document is the Release 2.0 of this specification.

Contributors

Jason Hawken	AMD
Jim Hunkins	AMD
Kenneth Ma	Broadcom Corporation
Alec Cawley	DisplayLink
Dan Ellis	DisplayLink
Trevor Hall	DisplayLink
Jeff Foerster	Intel Corporation
Wey-Yi Guy	Intel Corporation
Steve McGowan	Intel Corporation
Abdul Rahman Ismail (Chair)	Intel Corporation
Geert Knapen	Intel Corporation
Barry O'Mahony (Editor)	Intel Corporation
Sridharan Ranganathan	Intel Corporation
Ygal Blum	Jungo
Yoav Nissim	Jungo
Joel Silverman	Kawasaki Microelectronics, Inc.
Chris Yokum	MCCI
Richard Petrie	Nokia Corporation
Yoram Rimoni	Qualcomm, Inc
Shannon Cash	SMSC
Morgan Monks	SMSC
John Sisto	SMSC
Guy Stewart	SMSC
Alexey Orishko	ST_Ericsson
Will Harris	Texas Instruments
Grant Ley	Texas Instruments
Paul Berg	USB-IF

INTELLECTUAL PROPERTY DISCLAIMER

Please send comments via electronic mail to cswg-chair@usb.org

Table of Contents

Scope of This Release 3

Contributors 3

Revision History ..... Error! Bookmark not defined.

1. Introduction 11
1.1. Scope 11
1.2. Purpose 11
1.3. Related Documents 11
1.4. Terms and Abbreviations 11

2. Management Overview 13

3. Functional Characteristics 15
3.1. Control Endpoint 15
3.2. Bulk and Isochronous Endpoints 15
3.3. Interrupt Endpoint 15
3.4. Content Security Methods 15
3.5. Channels 15

4. Operational Model 17

5. Descriptors 19
5.1. Device Descriptor 19
5.2. Configuration Descriptor 19
5.3. Content Security Interface Descriptors 19
5.3.1. Standard Content Security Interface Descriptor 19
5.3.2. Class-specific Content Security Interface Descriptors 19
5.3.3. Content Security Descriptor Topology 23
5.4. Content Security Endpoint Descriptors 23
5.5. Content Security Interface Notification Format 24
5.5.1. Change_Channel_Settings Notification 24

6. Requests 25
6.1. Standard Requests 25
6.2. Class-Specific Requests 25
6.2.1. Class Specific Request Layout 25
6.2.2. Get_Channel_Settings Request 26
6.2.3. Set_Channel_Settings Request 26

Appendix A. Content Security Device Class Codes 27
A.2. Interface Class Code 27
A.3. Descriptor Codes 27
A.4. Resource Type Codes 27
A.5. Request Codes 27
A.6. Notification Codes 27
A.7. CSM Method ID Codes 28

List of Tables

Table 1-1: Terms and Abbreviations 12
Table 5-1: CS_General Descriptor 20
Table 5-2: Channel Descriptor 21
Table 5-3: Endpoint Channel Descriptor 22
Table 5-4: AVData Channel Descriptor 23
Table 5-5: CSM Descriptor 23
Table 5-6: CSI Notification Format 24
Table 5-7: Change_Channel_Settings Notification 24
Table 6-1: Content Security Method-specific Request Field Definitions 26
Table 6-2: Get_Channel_Settings Request 26
Table 6-3: Set_Channel_Settings Request 26
Table A-1: Interface Class Code 27
Table A-2: Descriptor Codes 27
Table A-3: Resource Type Codes 27
Table A-4: Request Codes 27
Table A-5: Notification Codes 27
Table A-6: Method ID Codes 28

List of Figures

Figure 4-1: Content Security Audio Example 17

1. Introduction

The need for protected and controlled distribution of digital content is the basis of the USB Content Security Interface (CSI) described in this specification.

1.1. Scope

This specification details the USB Content Security functionality, requests, and descriptors that support the various Content Security Methods (CSMs). CSMs, along with the corresponding CSM identification number assignment, are listed in Appendix A.6, "CSM Method ID Codes". Each CSM is detailed in a separate but associated Content Security Method Specification that contains data detailing its USB implementation. CSMs are added to the CS list by vote of USB Content Security Working Group (CSWG) and are subject to the USB specification promotion process.

The basic services needed to support the various CSMs are shown in the following list:

- Activating/De-activating a particular CSM.
- Associating/Dissociating a CSM to a data transport channel.
- A notification service that allows either the Host or Device or both to initiate asynchronous CSM related communication.
- Uniquely identifying each CSM for Host driver support.

1.2. Purpose

The purpose of the USB Content Security Class is to specify a common set of USB data transport requests and descriptors necessary to support the various Content Security Methods. CSMs may use some or all of the services detailed in this specification. A CSM defines the USB support given to a particular Content Protection Method (CPM). The intent is that each CSM will be detailed in a separate but associated USB CSM specification.

1.3. Related Documents

- [USB2.0] – Universal Serial Bus Specification, Revision 2.0, April 27, 2000 (referred to in this document as the USB 2.0 Specification).
- [USB3.0] – Universal Serial Bus 3.0 Specification, Revision 1, November 12, 2008 (referred to in this document as the USB 3.0 Specification).
- [USBAV1.0] – Universal Serial Bus Device Class Definition for Audio/Video Devices, Revision 1, December 07, , 2011 (referred to in this document as the AV Device Class Definition).
- [HDCP2.1] – High-bandwidth Digital Content Protection System, Interface Independent Adaptation; Revision 2.1; Digital Content Protection LLC; July 18, 2011. Available at: http://www.digital-cp.com/files/static_page_files/436E5E24-1A4B-B294-D0B95AAD084C773D/HDCP Interface Independent Adaptation Specification Rev2_1.pdf.
- [USBCS] – Universal Serial Bus Device Class Definition for Content Security Devices. Available at: http://www.usb.org/developers/devclass/
- [USBCC] – USB Common Class Specification Version 1.0. Available at: http://www.usb.org/developers/devclass/
- USBECNIAD – USB Engineering Change Notice: Interface Association Descriptors.
- [USBLANGIDS] – Universal Serial Bus Language Identifiers (LANGIDs), Revision 1.0, March 29, 2000.

1.4. Terms and Abbreviations

This section defines terms used throughout this document. For additional terms that pertain to the Universal Serial Bus, see Chapter 2, "Terms and Abbreviations," in [USB2.0] and [USB3.0].

Table 1-1: Terms and Abbreviations

Term	Description
AKE	Authentication and Key Exchange
Content Security Device	Any USB Device that contains a Content Security Interface.
Channel	A logical path over which secure data can be transmitted or received.
Content Provider	The owner of the content.
CPM	Content Protection Method, refers to a content provider protection scheme.
CS	Content Security. USB terminology for content protection.
CSC	Content Security Class. Refers to USB Device Class Definition for Content Security Devices.
CSI	Content Security Interface.
CSM	Content Security Method.
CSNS	Content Security Notification Service.
HDCP	High-bandwidth Digital Content Protection
Sink	The target of secure data transfers.
Source	The source of secure data transfers.

2. Management Overview

Protected Content typically refers to premium content. Content Protection Methods (CPM) have been developed for the controlled distribution of protected content. The Content Security Class was initiated to support CPM protocol exchange and protected content transport over the USB.

The Content Security (CS) Device Class provides a common set of extendable USB services, descriptors, and requests that are defined in this document. To support and provide for each CPM's specific USB needs, a corresponding USB Content Security Method (CSM) may be needed. Each CSM describes the use of the common CS services and defines additional CPM-specific USB transport services, descriptors, and requests.

The Content Security Interface (CSI) is either one of many interfaces that comprise a composite USB Device or it may be the only interface as in an authentication dongle.

The intended operation of the CSI when transferring protected content is for the CSI to process and transport protected content as prescribed by a CPM and its associated CSM. There is no need for additional Content Security specific endpoints to transport the protected content as the CSI uses the existing Device's data transport interfaces and channels to move the protected content. This is done with minimal impact to existing Device Classes and interfaces.

There are two common requests defined: the **Get_Channel_Settings** request and the **Set_Channel_Settings** request. The **Get_Channel_Settings** request is used to determine the CSM assigned to a channel. The USB Device may return a CSM value of zero indicating that there is no active CSM assigned to the specified channel. The **Set_Channel_Settings** request is used to assign a CSM to a channel or to deactivate a CSM assigned to a channel.

The definition and use of additional USB requests and interrupts is CSM dependent and is detailed in the associated CSM specification. For example, a CSM might need additional USB requests to exchange commands and responses between Host and Device. These requests may in turn support authentication, public-key formation, and System Renewal Messages.

The CSM also details USB Device Class interaction, notification format values (if used), and the format of content transferred over the associated USB data channel.

3. Functional Characteristics

Typically, the Content Security Interface is one part of a composite USB Device. A composite USB Device has several interfaces or groups of interfaces that serve different purposes. The Content Security Interface may be the only interface as in the case of an authentication dongle.

3.1. Control Endpoint

The Content Security Interface is addressed through the default control endpoint.

3.2. Bulk and Isochronous Endpoints

There is no need for additional Content Security specific endpoints to transport the protected content as the CSI uses the existing Device's data transport pipes to move protected data.

3.3. Interrupt Endpoint

The optional Interrupt IN endpoint in the Content Security Interface provides asynchronous notification of CSM-related events from a Device to the Host.

3.4. Content Security Methods

Protected content is distributed with a requirement that it be protected by a particular CPM. This content protection methodology allows compliant Devices to recognize and correctly process and/or transport the protected content. CSMs define the USB services, descriptors, and requests needed to support a given CPM.

3.5. Channels

In this specification, a channel is a logical construct representing a relationship between a transport resource, such as an interface or an endpoint or an AVData Entity, and one or more CSMs of which only one can be in use at any one time.

4. Operational Model

An example of an audio application is used to describe the operational model in the illustration below.

In this example, protected content is directed to a USB device (a pair of Audio speakers). The Host uses a **Set_Channel_Settings** request to the USB Device to activate the necessary CSM and prepare the Device to receive and correctly process the protected content. In the illustration, actual data flow is shown in dark gray arrows and logical data flow is shown in light gray arrows.

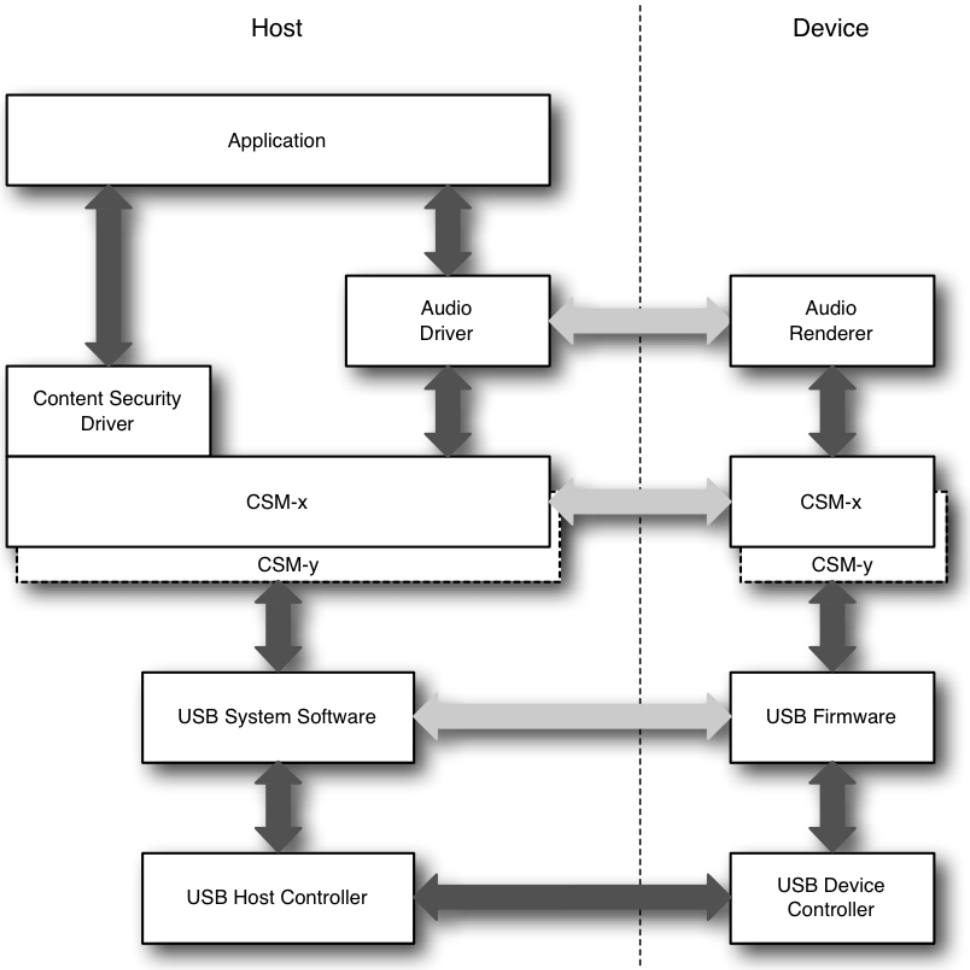

Figure 4-1: Content Security Audio Example

In the case where the USB Device is directed to send protected content to the Host, the USB Device notifies the Host and the Host responds with a **Get_Channel_Settings** request to determine the CSM and make the corresponding preparations for transporting the protected content.

5. Descriptors

This section defines all the Content Security class-specific interface and endpoint descriptors. The standard Device, Configuration, Interface and Endpoint descriptors are not duplicated here. Some of the fields in these descriptors are further explained when they take on values, specific to the Content Security Device Class.

5.1. Device Descriptor

The Device descriptor of a USB Device that incorporates a Content Security interface is almost always a composite Device and therefore it shall indicate that class information is to be found at the interface level. Different USB Device Classes use different methods to indicate this (for example, **the bDeviceClass, bDeviceSubClass** and **bDeviceProtocol** fields of the Device descriptor may contain a value of zero to force the Host enumeration software to search at the interface descriptor level for class-related information).

There is no class-specific Device descriptor for a Content Security Device.

5.2. Configuration Descriptor

The Configuration descriptor for a Content Security Device is identical to the standard Configuration descriptor defined in Section 9.6.2, "Configuration" of the *USB Specification*.

There is no class-specific Configuration descriptor.

5.3. Content Security Interface Descriptors

The Content Security Interface descriptors fully characterize the security capabilities of the Device. The standard Content Security Interface descriptor characterizes the USB Device as a Device that supports the Content Security services offered by the Content Security Device Class. The class-specific Interface descriptor describes the security capabilities of the particular Device by, for example:

- Identifying the CSM or CSMs implemented on the Device
- Providing CSM implementation details
- Associating secure content transfer channels with endpoints/Interfaces/AVData Entities on the Device

5.3.1. Standard Content Security Interface Descriptor

The standard Content Security Interface descriptor is identical to the standard Interface descriptor defined in Section 9.6.3, "Interface" of the *USB Specification*, with the **bInterfaceClass** field set to CONTENT_SECURITY and the **bInterfaceSubClass** and **bInterfaceProtocol** fields set to zero. The value of the CONTENT_SECURITY Interface Class code is specified in Appendix A.1, "Interface Class Code".

5.3.2. Class-specific Content Security Interface Descriptors

The class-specific Interface descriptors provide the information needed to describe the characteristics and behavior of the Content Security Interface on a Content Security-enabled Device.

There are three types of Content Security class-specific Interface descriptors for the Security class:

- CS_General descriptor**:** identifies the Content Security Interface version number.
- Channel descriptor: identifies one or more Content Security Methods the Device can use on a channel. CS Devices shall contain at least one Channel Descriptor and each Channel Descriptor shall identify at least one CSM.
- Content Security Method descriptor: describes one CSM implemented on a Device. CS Devices shall contain at least one CSM descriptor.

5.3.2.1. CS_General Descriptor

This descriptor serves to identify the Content Security Interface Version number.

Table 5-1: CS_General Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes
1	**bDescriptorType**	1	Constant	CS_GENERAL. See Appendix A.2, “Descriptor Codes”.
2	**bcdVersion**	2	BCD	Content Security Interface version number in Binary-Coded Decimal (e.g., version 2.10 is 0x0210).

5.3.2.2. Channel Descriptor

This section defines the class-specific Channel descriptor. In this specification, a Channel is a logical construct that creates a relationship between a transport resource (such as a USB Interface or Endpoint or AVData Entity) used by the stream that requires protection and one or more CSMs of which only one can be used at any given time for that resource.

Each Channel descriptor defines one Channel on the Device. The Channel descriptor associates the Channel with the transport resource that needs content protection services and assigns a unique ChannelID to the Channel. The ChannelID is used in various Content Security requests to influence the behavior of the Channel, such as selecting the current CSM on the Channel, switching protection on and of, etc.

Within a single Device Configuration, each Channel descriptor shall specify a unique transport resource. Specifically, two Channel descriptors cannot reference the same transport resource in their transport resource-identifying fields (**bInterfaceNumber**, **bAlternateSetting**, and **bLogicalUnit** fields for the Interface Channel descriptor; **bEndpointAddress** field for the Endpoint Channel descriptor; and **wEntityID** field for the AVData Channel Descriptor – see below).

5.3.2.2.1. Interface Channel Descriptor

This section defines the class-specific Channel descriptor when the transport resource that needs content protection services is a USB Interface. In this case, the USB Interface Number and the Alternate Setting value as defined in the **bInterfaceNumber** and **bAlternateSetting** fields of its associated standard Interface descriptor together uniquely identify the transport resource. If the interface supports logical units, then the **bLogicalUnit** field shall also be included to uniquely identify the transport resource.

The Interface Channel descriptor is outlined in the following table:

Table 5-2: Interface Channel Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes
1	**bDescriptorType**	1	Constant	CHANNEL. See Appendix A.2, “Descriptor Codes”.
2	**bChannelID**	1	Number	ID of the Channel, shall be a non-zero value that is unique across the Device.
3	**bResourceType**	1	Number	INTERFACE. See Appendix A.3, “Resource Type Codes”.
4	**bInterfaceNumber**	1	Number	This field contains the interface number of the targeted interface.
5	**bAlternateSetting**	1	Number	This field contains the alternate setting value for the interface to which this channel applies.
6	**bLogicalUnit**	1	Number	This field indicates the Logical Unit within the protected interface to which this Channel applies. The definition of a Logical Unit is dependent upon the interface being protected. If the interface does not support Logical Units, then the **bLogicalUnit** field shall be set to 0.
7	**bMethod[0]**	1	Number	Method ID of a Content Security Method that can be activated on the Channel identified by the **bChannelID** field of this descriptor. See Table A-6, “Method ID Codes”.
8	**bReserved[0]**	1	Number	Reserved. Shall be set to zero. Deprecated from Content Security Specification 1.0.
-	**-**	-		
7 + 2(N-1)	**bMethod[N-1]**	1	Number	Method ID of a Content Security Method that can be activated on the Channel identified by the **bChannelID** field of this descriptor.
8 + 2(N-1)	**bReserved[N-1]**	1	Number	Reserved. Shall be set to zero. Deprecated from Content Security Specification 1.0.

5.3.2.2.2. Endpoint Channel Descriptor

This section defines the class-specific Channel descriptor when the transport resource that needs content protection services is a USB Endpoint. In this case, the endpoint address; i.e. endpoint number and direction as defined in the **bEndpointAddress** field of its associated standard Endpoint descriptor uniquely identifies the transport resource. The Endpoint Channel descriptor is outlined in the following table:

Table 5-3: Endpoint Channel Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes
1	**bDescriptorType**	1	Constant	CHANNEL. See Appendix A.2, “Descriptor Codes”.
2	**bChannelID**	1	Number	ID of the Channel, shall be a non-zero value that is unique across the Device.
3	**bResourceType**	1	Bitmap	ENDPOINT. See Appendix A.3, “Resource Type Codes”.
4	**bEndpointAddress**	1	Number	This field contains the endpoint address of the targeted endpoint, where: D7: Direction 0 = OUT 1 = IN D6..D4: Reserved and set to zero D3..D0: Endpoint number
5	**bReserved1**	1	Number	Reserved. Shall be set to zero.
6	**bReserved2**	1	Number	Reserved. Shall be set to zero.
7	**bMethod[0]**	1	Number	Method ID of a Content Security Method that can be activated on the Channel identified by the **bChannelID** field of this descriptor. See Table A-6, “Method ID Codes”.
8	**bReserved[0]**	1	Number	Reserved. Shall be set to zero. Deprecated from Content Security Specification 1.0.
-	**-**	-		
7 + 2(N-1)	**bMethod[N-1]**	1	Number	Method ID of a Content Security Method that can be activated on the Channel identified by the **bChannelID** field of this descriptor.
8 + 2(N-1)	**bReserved[N-1]**	1	Number	Reserved. Shall be set to zero. Deprecated from Content Security Specification 1.0.

5.3.2.2.3. AVData Channel Descriptor

This section defines the class-specific Channel descriptor when the transport resource that needs content protection services is an AVData Entity as defined by [USBAV1.0]. In this case, the AVData Entity ID as assigned by the AVFunction implementation together with its Alternate Setting uniquely identifies the transport resource. The AVControl Interface Number and its Alternate Setting is further included to identify the AVFunction within the USB Device.

Note that even in the case where the transport resource is an AVData Streaming Interface, the AVData Channel descriptor is used rather than the USB Interface Channel descriptor.

The AVData Channel descriptor is outlined in the following table:

Table 5-4: AVData Channel Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes.
1	**bDescriptorType**	1	Constant	CHANNEL. See Appendix A.2, “Descriptor Codes”.
2	**bChannelID**	1	Number	ID of the Channel, shall be a non-zero value that is unique across the Device.
3	**bResourceType**	1	Bitmap	AVDATA. See Appendix A.3, “Resource Type Codes”.
4	**bInterfaceNumber**	1	Number	This field contains the interface number of the AVFunction’s AVControl Interface.
5	**bAlternateSetting**	1	Number	This field contains the alternate setting value for the AVFunction’s AVControl Interface.
6	**wEntityID**	2	Number	EntityID of the targeted AVData Entity (AVData FrameBuffer Entity or AVData Streaming Interface.
8	**bAVDataAltSetting**	1	Number	Alternate Setting of the AVData Entity. If the resource is independent of the AVData Entity’s Alternate Setting, this field shall be set to zero.
9	**bMethod[0]**	1	Number	Method ID of a Content Security Method that can be activated on the Channel identified by the **bChannelID** field of this descriptor. See Table A-6, “Method ID Codes”.
…	…	…	…	…
9+2(N-1)	**bMethod[N-1]**	1	Number	Method ID of a Content Security Method that can be activated on the Channel identified by the **bChannelID** field of this descriptor.

5.3.2.3. Content Security Method (CSM) Descriptor

This section defines the class-specific Content Security Method (CSM) descriptor. A Device that supports Content Security services shall report one CSM descriptor for each Content Security Method implemented by the Device.

Table 5-5: CSM Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes
1	**bDescriptorType**	1	Constant	CSM. See Appendix A.2, “Descriptor Codes”.
2	**bMethodID**	1	Number	Method ID of a Content Security Method. See Table A-6, “Method ID Codes”.
3	**iCSMDescriptor**	1	Index	Index of a string descriptor that describes the Content Security Method. A value of zero indicates that there is no string descriptor associated.
4	**bcdVersion**	2	BCD	CSM Descriptor Version number in Binary-Coded Decimal (e.g., version 2.10 is 0x0210).
6	**CSMData**	N		Optional field(s) that provides Device-specific implementation details for the CSM identified by the **bMethodID** field.

5.3.3. Content Security Descriptor Topology

All the CS related descriptors on the Device are available to the Host after successful completion of a standard GET_DESCRIPTOR (configuration) request. All Channel and CSM descriptors follow the standard Content Security interface descriptor and precede any Content Security interface endpoint descriptors or any other interface descriptors.

5.4. Content Security Endpoint Descriptors

The Content Security Interface on a Device is addressed through the default Control endpoint (endpoint 0), which every Device shall implement. However, if the optional Interrupt IN endpoint is required for any CSM implemented by

the device they shall be present and described by an Endpoint descriptor. If implemented, a Content Security Interface Interrupt Endpoint descriptor is a standard Endpoint descriptor with the **bmAttributes** field set to Interrupt.

5.5. Content Security Interface Notification Format

A standard Content Security Notification is defined to allow security notifications from multiple channels on a single interrupt endpoint. Devices that generate notification events can implement a single Interrupt IN endpoint for all channels.

Table 5-6: CSI Notification Format

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this Notification, in bytes
1	**bChannel**	1	Number	ID of the channel that generated the notification.
2	**bNotification**	1	Number	Identifies the type of notification. The value assignments are defined in Appendix A.5, “Notification Codes”.
3	**Data**	N	data	CSM-defined notification data

5.5.1. Change_Channel_Settings Notification

The **Change_Channel_Settings** notification is used to request that a CSM be activated and linked to the given channel. The Host upon receiving this notification will issue a **Set_Channel_Settings** request that actually causes the channel settings to be changed. Note that a CS channel setting can only be changed by a **Set_Channel_Settings** request.

If a Device supports the **Change_Channel_Settings** notification, it shall implement the CS notification Service. Otherwise the use of the notification service is optional and CSM dependent.

Table 5-7: Change_Channel_Settings Notification

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this Notification, in bytes
1	**bChannelID**	1	Number	ID of the channel that generated the notification.
2	**bNotification**	1	Number	CHANGE_CHANNEL_SETTINGS. Refer to Appendix A.5, “Notification Codes”.
3	**Data**	1	data	Data contains the CSM MethodID that the USB Device wants activated on the given Channel.

6. Requests

This section specifies the requests a Device that offers Content Security services can receive from the Host at its Content Security Interface.

6.1. Standard Requests

The Content Security Device Class supports the standard requests described in Section 9, "USB Device Framework," of the *USB Specification*. The Content Security Device Class places no specific requirements on the values for the standard requests.

6.2. Class-Specific Requests

All Content Security class-specific requests are directed to the Content Security Interface. The basic Content Security class-specific request layout is the same as defined in Section 9.3 of the *USB Specification, Version 1.1*. The meaning of each request field is defined in the next paragraphs.

For most Content Security class-specific requests, the content of all the request fields except for the **bmRequestType** and **wIndex** fields, is CSM-specific and defined in the respective CSM Definition documents. For more information, see Section 6.2.1, "Class Specific Request Layout".

Two **bRequest** field values that all Content Security Device Class Devices shall accept, in addition to the requests specific to each of the Content Security Methods the Device implements, are:

- **Get_Channel_Settings** request
- **Set_Channel_Settings** request

The Host uses the **Get_Channel_Settings** request to determine the CSM currently associated to an interface or endpoint (Channel). The **Set_Channel_Settings** request assigns a new CSM to a Channel.

Devices receiving unsupported requests shall stall the control pipe upon receipt of an unsupported request.

6.2.1. Class Specific Request Layout

This section details the general structure of the Content Security class-specific requests.

6.2.1.1. Content Security Method-Specific Requests

The basic Content Security class-specific request layout is the same as defined in Section 9.3 of the *USB Specification, Version 1.1*. The meaning of each request field for Content Security Method-specific requests is defined in Table 6-1.

Table 6-1: Content Security Method-specific Request Field Definitions

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	Specific request. See requests in this document. CSMs are allowed to define additional requests as needed.
2	**wValue**	2	Value	Word-sized field that is bRequest dependent.
4	**wIndex**	2	Value	Word-sized field that is bRequest dependent.
6	**wLength**	2	Count	Word-sized field that specifies the byte length of the associated data field.

6.2.2. Get_Channel_Settings Request

The **Get_Channel_Settings** request returns the ID of the Content Security Method (CSM) currently selected for a specified channel.

Table 6-2: Get_Channel_Settings Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0b10100001	GET_CHANNEL_SETTINGS	0	HByte: Channel ID LByte: CSI interface number	2	HByte: 0 LByte: Method ID of the currently running CSM

Bit D7 of the **bmRequestType** field specifies that this is a Get request (D7 = 0b1). It is a class-specific request (D6..5 = 0b01), directed to the Content Security interface (D4..0 = 0b00001).

The **bRequest** field is set to GET_CHANNEL_SETTINGS. The value of this constant is defined in Appendix A.4, "Request Codes".

6.2.3. Set_Channel_Settings Request

The **Set_Channel_Settings** request sets the current CSM for a channel and is the only method for assigning a CSM to an interface or endpoint.

Table 6-3: Set_Channel_Settings Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0b00100001	SET_CHANNEL_SETTINGS	HByte: 0 LByte: Method ID of the new CSM	HByte: Channel ID LByte: CSI interface number	0	-

Bit D7 of the **bmRequestType** field specifies that this is a Set request (D7 = 0b0). It is a class-specific request (D6..5 = 0b01), directed to the Content Security interface (D4..0 = 0b00001).

The **bRequest** field is set to SET_CHANNEL_SETTINGS. The value of this constant is defined in Appendix A.4, "Request Codes".

If the LByte of the **wValue** field is set to zero, this causes the specified interface or endpoint (via its Channel ID) to deactivate the currently active CSM, effectively disabling all Content Security services for that Channel.

Appendix A. Content Security Device Class Codes

A.1. Interface Class Code

Table A-1: Interface Class Code

Interface Class Code	Value
CONTENT_SECURITY	0x0D

A.2. Descriptor Codes

Table A-2: Descriptor Codes

Descriptor Code	Value
CS_GENERAL	0x21
CHANNEL	0x22
CSM	0x23

A.3. Resource Type Codes

Table A-3: Resource Type Codes

Descriptor Code	Value
REOURCE_TYPE_UNDEFINED	0b00000000
INTERFACE	0b00000001
ENDPOINT	0b00000010
AVDATA_ENTITY	0b10000000

A.4. Request Codes

Table A-4: Request Codes

Request Code	Value
REQUEST_UNDEFINED	0x00
GET_CHANNEL_SETTINGS	0x01
SET_CHANNEL_SETTINGS	0x02
Reserved for future extensions	0x03..0x7F
CSM-defined	0x80..0xFF

A.5. Notification Codes

Table A-5: Notification Codes

Notification Code	Value
NOTIFICATION_UNDEFINED	0x00
CHANGE_CHANNEL_SETTINGS	0x01
Reserved for future extensions	0x02..0x7F
CSM-defined	0x80..0xFF

A.6. CSM Method ID Codes

Table A-6: Method ID Codes

CSM	Method ID	Comments
CSM_UNDEFINED	0x00	
Basic Authentication Protocol (CSM-1)	0x01	Deprecated. Do not use.
USB Digital Transmission Content Protection (CSM-2)	0x02	
Open Copy Protection System (CSM-3)	0x03	Deprecated. Do not use.
Elliptic Curve Content Protection Protocol (CSM-4)	0x04	Deprecated. Do not use.
High-bandwidth Digital Content Protection System (CSM-5)	0x05	