Universal Serial Bus
Device Class Definition
for
Content Security Devices

Content Security Method 5
High-bandwidth Digital Content Protection 2.1
(HDCP 2.1) Implementation

Release 1.0

June 12, 2012

Scope of This Release

This document is the Release 1.0 of this Content Security Method 5 Definition.

Contributors

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

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

Table of Contents

List of Tables

List of Figures

1. Introduction

This document describes the USB transport services and protocol formats that support High-bandwidth Digital Content Protection (HDCP) 2.1.

1.1. Scope

USB CSM5 describes the USB transport services, descriptors, and requests necessary to support HDCP2.1 protocols over USB. This document does not change or alter HDCP2.1 functionality.

The Content Security Class (CSC) specification allows Content Security Methods (CSM) to define additional requests as needed. CSM5 defines additional USB CSC requests in order to support HDCP2.1 AKE protocols between USB Host and Device. In addition, CSM5 implements the Content Security Notification Service and defines additional notifications that are needed to support HDCP2.1 protocols.

1.2. 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). Available at: http://www.usb.org/developers/docs/
- [USB3.0] – Universal Serial Bus 3.0 Specification, Revision 1, November 12, 2008 (referred to in this document as the USB 3.0 Specification). Available at: http://www.usb.org/developers/docs/
- [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.

1.3. 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	Congtent 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. CSM-5 Content Security Class Additions

The USB Device Class Definition for Content Security Devices (CSC) allows Content Security Methods to define additional services as needed. HDCP2.1 requires four USB Requests to transfer the Authentication Protocol commands and responses also referred to as Authentication Protocol Messages (APM). The CS Notification Service (CSNS) is used to allow USB devices to initiate HDCP2.1 Authentication Protocols.

HDCP Authentication Protocol transactions are performed in order to authenticate that the Device or Host that is receiving premium content, is a valid HDCP receiver. The keys exchanged during an Authentication session are used to protect the stream identified by the ChannelID. If there are multiple streams per Host/Device pair, then an Authentication session shall be established per ChannelID.

2.1. Authentication Protocol USB Requests

HDCP2.1 requires four USB requests to transfer the Authentication Protocol command frames and corresponding responses. This section details the structure of these requests. The General Request format for Authentication Protocol Command Response request is as follows:

Table 2-1: General Request Layout

Offset	Field	Size	Value	Description
0	**bmRequestType**	1	Bitmap	D4..0: Recipient 0b00001: Interface D6..5: Type 0b01: Class D7: Data Transfer Direction 0b0: Host to Device 0b1: Device to Host
1	**bRequest**	1	Number	CSM-5 Requests: PUT_COMMAND, GET_RESPONSE, GET_COMMAND, PUT_RESPONSE.
2	**wValue**	2	Number	HByte: Reserved. Shall be set to zero. LByte: CSM-5 Identifier. Shall be set to 0x05.
4	**wIndex**	2	Number	HByte: Channel ID. LByte: CSI Interface Number.
6	**wLength**	2	Number	Byte length of the Authentication Protocol Command or Resonse.

2.2. Command and Response Requests Format

The requests are paired together, one pair (PUT_COMMAND, GET_RESPONSE) is used to send Authentication Protocol commands to the Device and return the associated response. The other pair is used to retrieve an Authentication Protocol command from the Device and send the associated response.

Table 2-2: Command-Response Pairing

Command	Associated Response
PUT_COMMAND	GET_RESPONSE
GET_COMMAND	PUT_RESPONSE

2.2.1. PUT_COMMAND

The PUT_COMMAND request is used to transfer an Authentication Protocol Command from the Host to the Device.

Table 2-3: PUT_COMMAND Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0b00100001	PUT_COMMAND	HByte: 0x00 LByte: 0x05	HByte: Channel ID LByte: CSI interface number	Length of Command (in bytes)	HDCP 2.1 Command

2.2.2. GET_RESPONSE

The GET_RESPONSE request is used to transfer an Authentication Protocol Response from the Device to the Host.

Table 2-4: GET_RESPONSE Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0b10100001	GET_RESPONSE	HByte: 0x00 LByte: 0x05	HByte: Channel ID LByte: CSI interface number	Length of Response (in bytes)	HDCP 2.1 Response

2.2.3. GET_COMMAND

The GET_COMMAND request is used to transfer an Authentication Protocol Command from the Device to the Host.

Table 2-5: GET_COMMAND Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0b10100001	GET_COMMAND	HByte: 0x00 LByte: 0x05	HByte: Channel ID LByte: CSI interface number	Length of Command (in bytes)	HDCP 2.1 Command

2.2.4. PUT_RESPONSE

The PUT_RESPONSE request is used to transfer an Authentication Protocol Response from the Host to the Device.

Table 2-6: PUT_RESPONSE Request

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0b00100001	PUT_RESPONSE	HByte: 0x00 LByte: 0x05	HByte: Channel ID LByte: CSI interface number	Length of Response (in bytes)	HDCP 2.1 Response

2.3. HDCP 2.1 Commands and Responses

HDCP messages are exchanged between Host and Device or vice-versa according to Authentication Protocol procedures specified in [HDCP2.1], using a USB request specified in Section 2.2, "Command and Response Requests Format". HDCP Messages sent from an HDCP Transmitter to an HDCP Receiver are designated as Commands for the purposes of this document. Messages sent from an HDCP Receiver to an HDCP Transmitter are designated as Responses. Both USB Hosts and USB Devices may act as either HDCP Transmitters or HDCP Receivers.

3. CSM-5 Descriptors

This section describes information relevant to the CSM-5 instantiation and the use of CSC descriptors. Each subsection corresponds to a CSC descriptor and only values pertinent to CSM-5 are listed in each subsection. Note that some subsections may not have any data and therefore the definition and use of the descriptor as specified in CSC is sufficient.

3.1. Device Descriptor

No additional definition needed.

3.2. Configuration Descriptor

No additional definition needed.

3.3. Content Security Interface Descriptor

No additional definition needed.

3.4. Channel Descriptor

Depending on the transport resource that needs content protection services, the appropriate Channel descriptor shall be selected (Interface, Endpoint, or AVData Channel descriptor).

The Channel descriptor shall indicate at least CSM-5 as one of the supported Content Security Methods, i.e. one of the bMethod[i] fields shall be set to 0x05.

Other Content Security Methods may be supported on the same Channel.

3.5. Content Security Method (CSM-5) Descriptor

Table 3-1: CSM-5 Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes: 6.
1	**bDescriptorType**	1	Constant	CSM. See Appendix A.2 in [USBCS].
2	**bMethodID**	1	Number	Method ID of a Content Security Method. Shall be set to 0x05.
3	**iCSMDescriptor**	1	Index	Index of the String descriptor that describes the Content Security Method. See Section 3.6, “CSM-5 String Descriptor”.
4	**bcdVersion**	2	BCD	CSM Descriptor Version number in Binary-Coded Decimal. This filed identifies the version of the HDCP that is supported. Shall be set to 0x0210.

The **CSMData** field is not used and shall therefore not be present.

3.6. CSM-5 String Descriptor

Table 3-2: CSM-5 String Descriptor

Offset	Field	Size	Value	Description
0	**bLength**	1	Number	Size of this descriptor, in bytes: 108.
1	**bDescriptorType**	1	Constant	STRING. See Table 9-5 in [USB2.0] or [USB3.0].
2	**bString**	106 (0x6A)	UTF-16LE	This field shall contain the following string (without the square brackets): [High-bandwidth Digital Content Protection Revision 2.1]

4. HDCP 2.1 Packet Format

The contents and structure of the APM Data are detailed in [HDCP2.1].

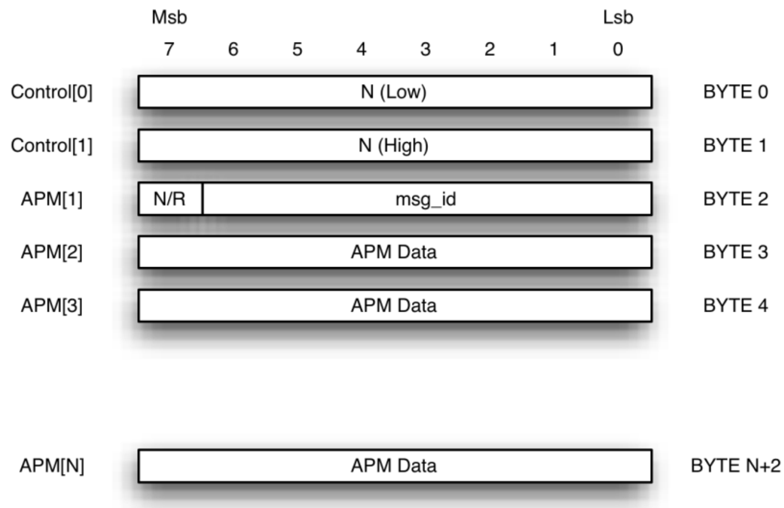

Figure 4-1: HDCP 2.1 Packet Format

As defined in Section 4.2 of the reference, the first byte, APM[1], is the **msg_id** field of the AKE message, with the remaining bytes in the field as defined for the particular message. Preceding the **msg_id** field is the **N** field, indicating the length in bytes of the APM data field, APM[1..N].

The N/R bit of the **msg_id** field shall be set to one when the USB Device is not yet ready to supply the data for the APM in question. Otherwise, the N/R bit shall be set to zero.

When the N/R bit is set to one then:

- If there is a pending Command or Response being processed, the **msg_id** field shall be set to the value of the pending Command or Response; otherwise it shall be set to zero.
- **N** field shall be set to one.

The USB device shall not transition to the next step in the sequence diagrams (see Section 195) unless the entire command/response has been successfully sent.

5. Authentication Sequence Diagrams

All of these are normative examples.

5.1. AKE Sequence – Host is HDCP Transmitter

Figure 5-1 shows an example Authentication and Key Exchange (AKE) sequence diagram for the case where the USB Host is the HDCP Transmitter and the USB Device is the HDCP Receiver. It is assumed that the Host and Device have been previously paired. See Figure 2.2 of [HDCP2.1] for an illustration of this transaction.

Note that in some cases, the Device returns a NOT_YET_READY (i.e. the device sets the N/R bit set to one in the **msg_id** field) in response to a GET_COMMAND or GET_RESPONSE request. The Host then attempts the request again at a later time.

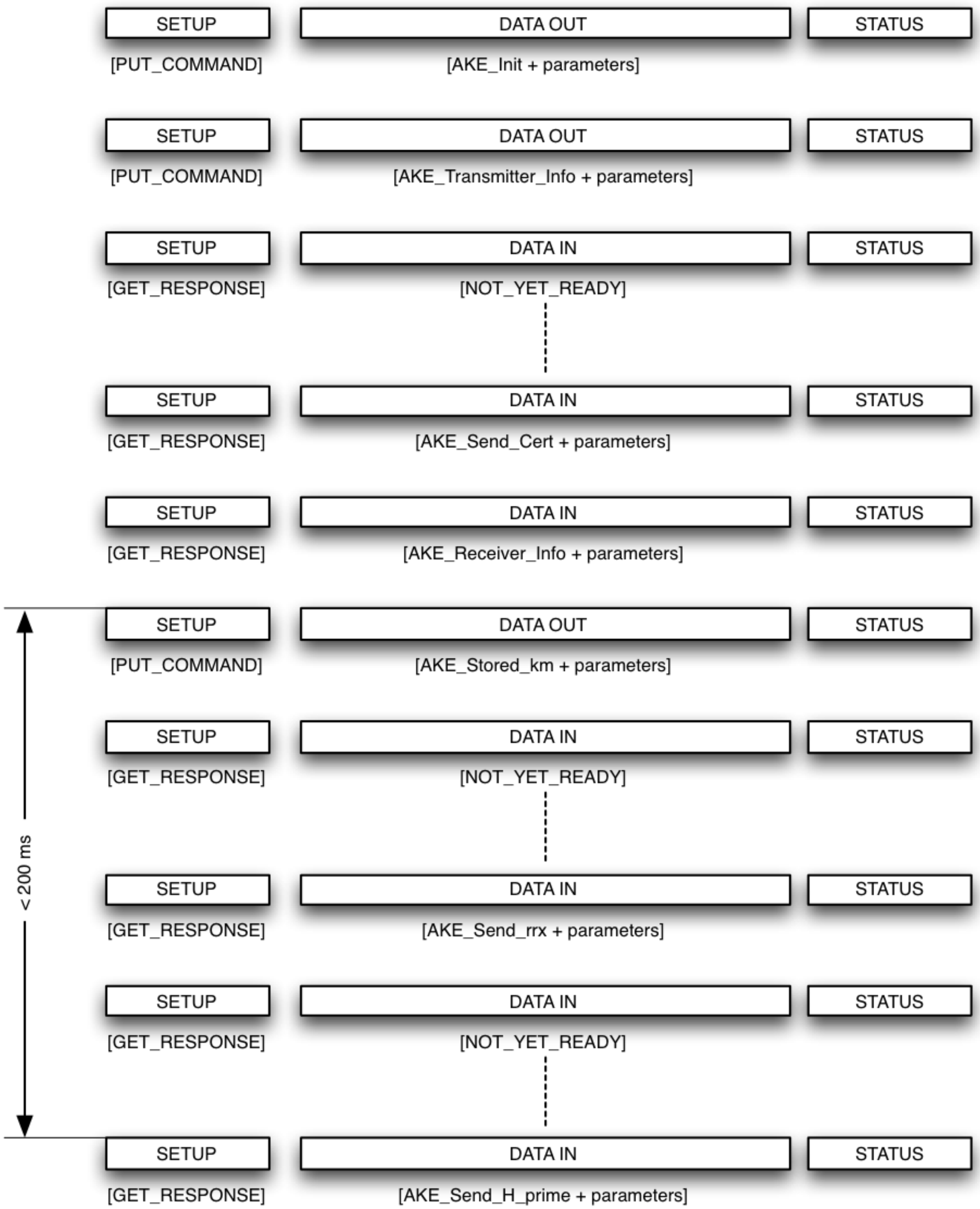

Figure 5-1: Example AKE Sequence, previously paired HDCP Transmitter (Host) and HDCP Receiver (Device)

Note: If the Host requests the AKE_Send_H_prime Response less than 200 ms after the AKE_Stored_km Command, the Device may return NOT_YET_READY. The AKE_Send_H_prime Response in a successful HDCP AKE transaction shall be returned no later than 200 ms after the AKE_Stored_km Command, but may be returned sooner.

5.2. AKE Sequence – Device is HDCP Transmitter

Figure 5-2 shows an example Authentication and Key Exchange (AKE) sequence diagram for the case where the USB Host is the HDCP Receiver and the USB Device is the HDCP Transmitter. It is assumed that the Host and Device have been previously paired. See Figure 2.2 of [HDCP2.1] for an illustration of this transaction.

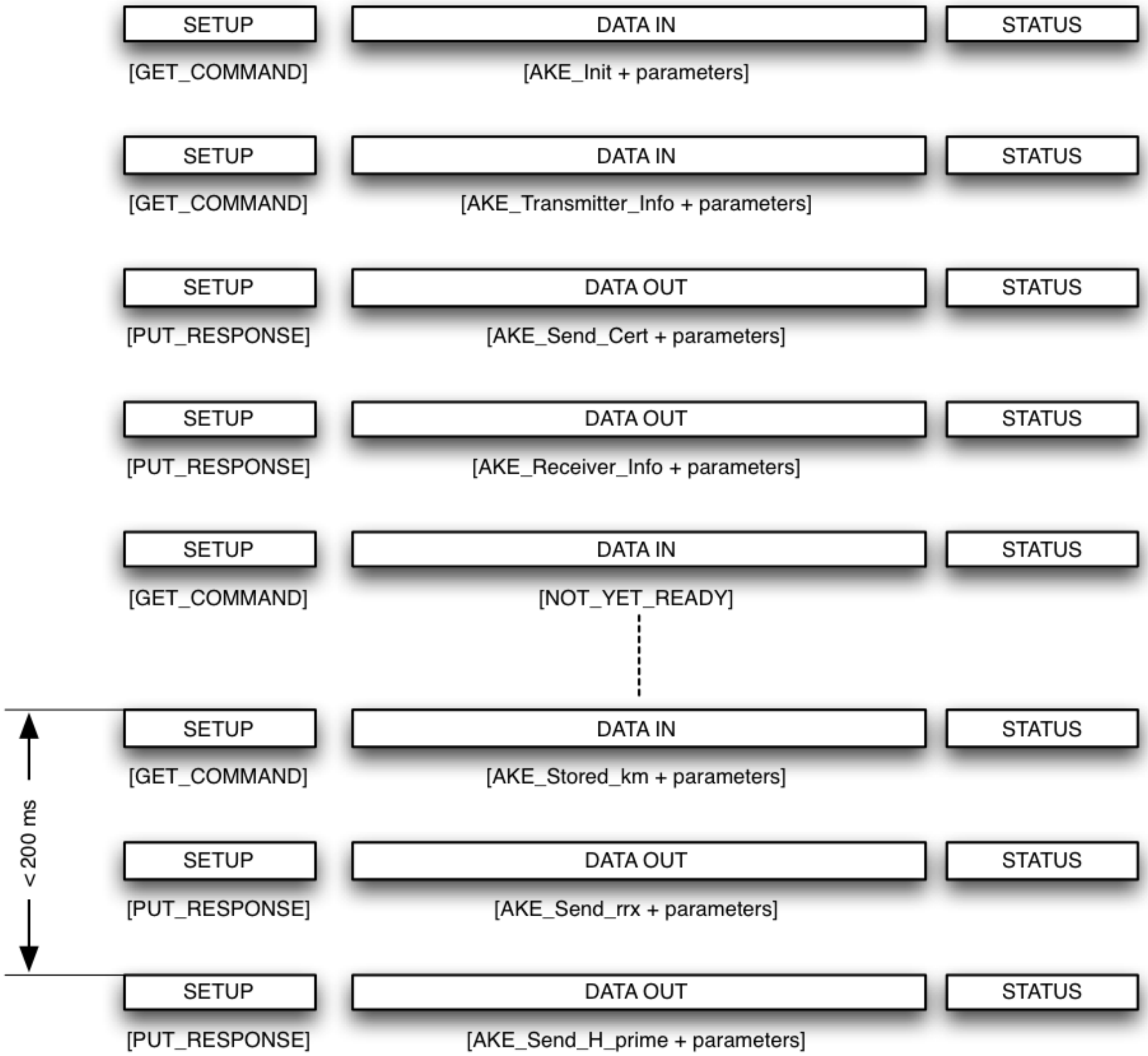

Figure 5-2: Example AKE Sequence, previously paired HDCP Transmitter (Device) and HDCP Receiver (Host)

5.3. Locality Check Sequence – Host is HDCP Transmitter

Figure 5-3 shows an example Locality Check (LC) sequence diagram for the case where the USB Host is the HDCP Transmitter and the USB Device is the HDCP Receiver. See Figure 2.5 of [HDCP2.1] for an illustration of this transaction.

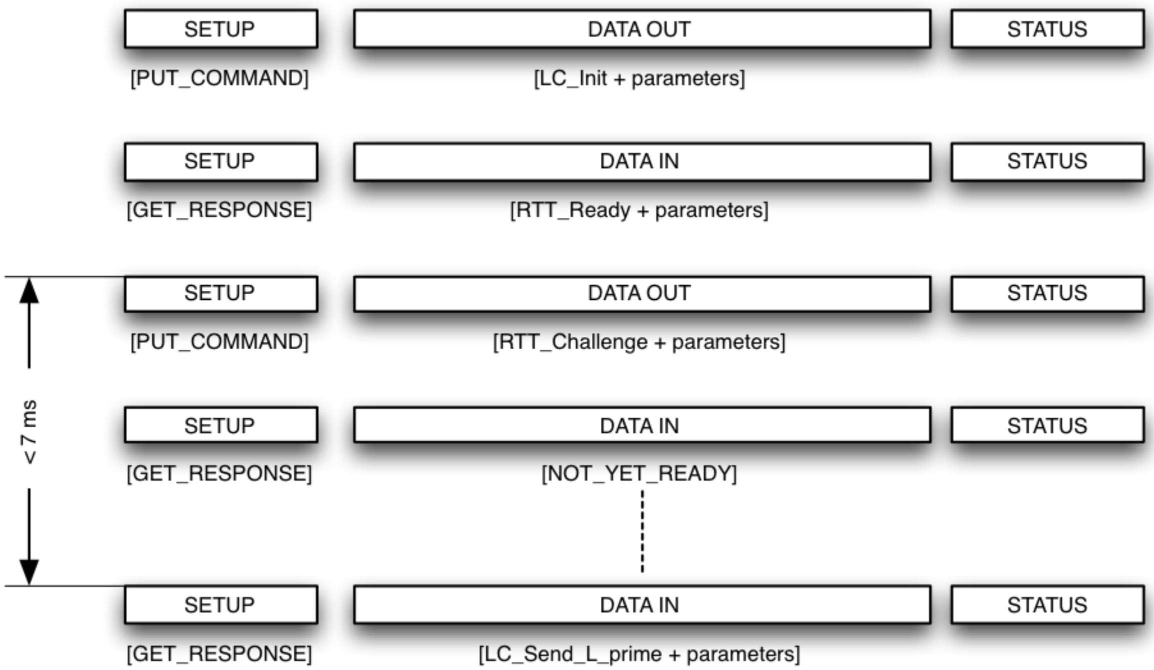

Figure 5-3: Example Locality Check (LC) Sequence, HDCP Transmitter (Host) and HDCP Receiver (Device)

5.4. Locality Check Sequence – Device is HDCP Transmitter

Figure 5-4 shows an example Locality Check (LC) sequence diagram for the case where the USB Host is the HDCP Receiver and the USB Device is the HDCP Transmitter. See Figure 2.5 of [HDCP2.1] for an illustration of this transaction.

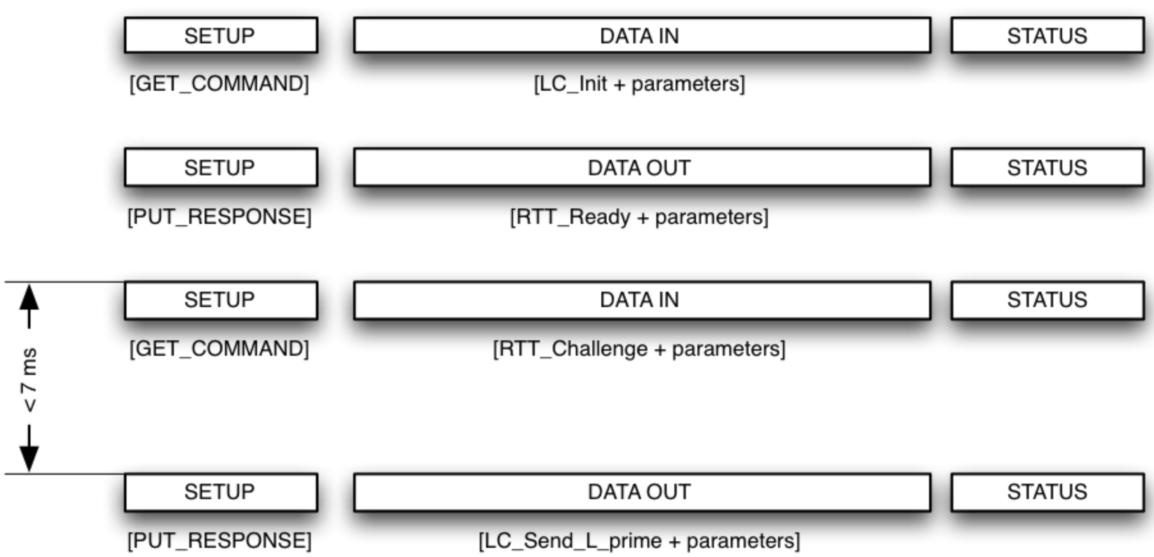

Figure 5-4: Example Locality Check (LC) Sequence, HDCP Transmitter (Device) and HDCP Receiver (Host)

5.5. SKE Exchange Sequence – Host is HDCP Transmitter

Figure 5-5 shows an example SKE sequence diagram for the case where the USB Host is the HDCP Transmitter and the USB Device is the HDCP Receiver.

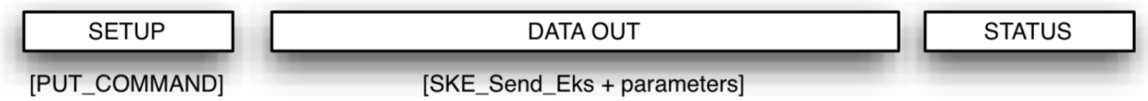

Figure 5-5: Example SKE Exchange Sequence, HDCP Transmitter (Host) and HDCP Receiver (Device)

5.6. SKE Exchange Sequence – Device is HDCP Transmitter

Figure 5-6 shows an example SKE sequence diagram for the case where the USB Host is the HDCP Receiver and the USB Device is the HDCP Transmitter.

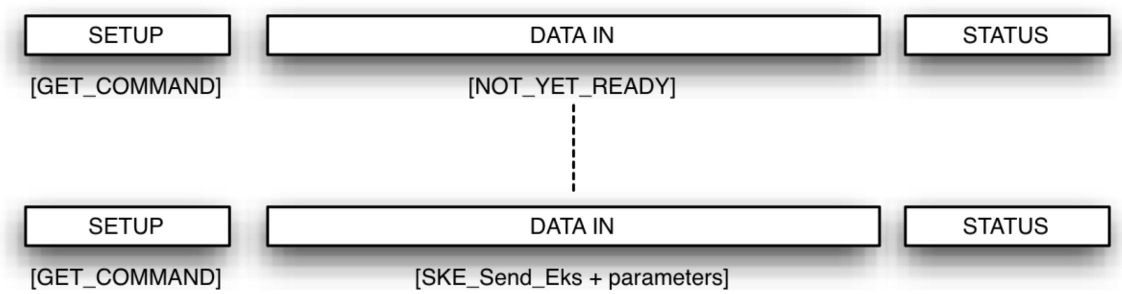

Figure 5-6: Example SKE Exchange Sequence, HDCP Transmitter (Device) and HDCP Receiver (Host)

5.7. Repeater Authentication Report Sequence – Host is HDCP Transmitter

Figure 5-7 shows an example Repeater Authentication Report sequence diagram for the case where the USB Host is the HDCP Transmitter and the USB Device is the HDCP Receiver. See Figure 2.6 of [HDCP2.1] for an illustration of this transaction.

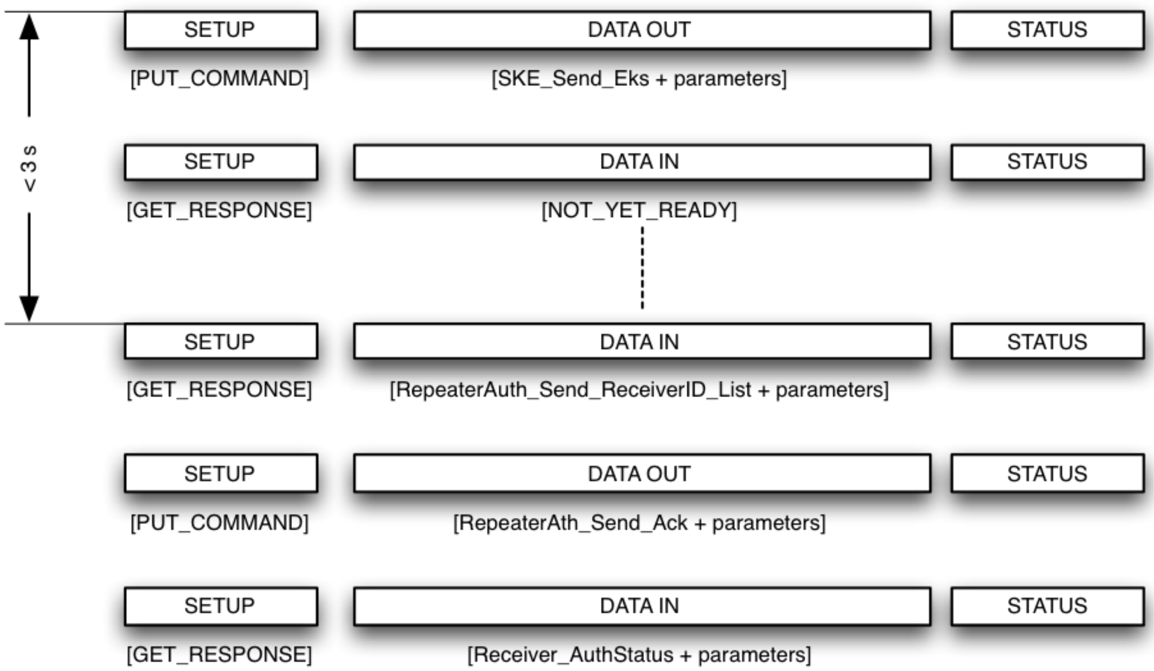

Figure 5-7: Example Repeater Authentication, HDCP Transmitter (Host) and HDCP Receiver (Device)

5.8. Repeater Authentication Report Sequence – Device is HDCP Transmitter

Figure 5-8 shows an example Repeater Authentication Report sequence diagram for the case where the USB Host is the HDCP Receiver and the USB Device is the HDCP Transmitter. See Figure 2.6 of [HDCP2.1] for an illustration of this transaction.

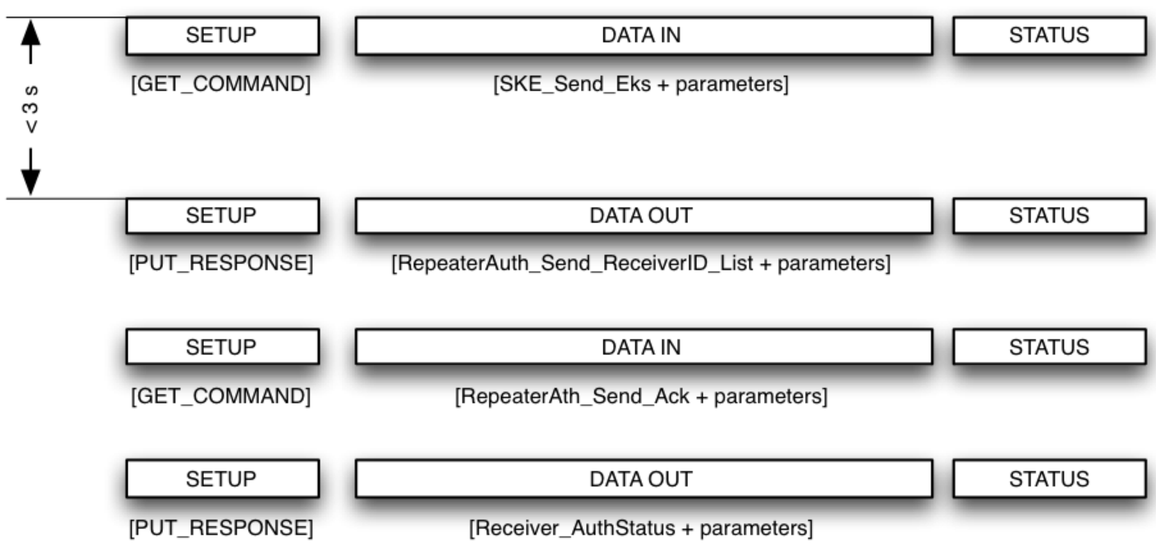

Figure 5-8: Example Repeater Authentication, HDCP Transmitter (Device) and HDCP Receiver (Host)

6. HDCP 2.1 Stream Parameters

Each stream protected by HDCP2.1 shall contain parameters to indicate real-time encryption state (enabled or disabled), and to maintain encryption synchronization between HDCP Transmitter and HDCP Receiver. As described in [HDCP2.1], each stream has an associated 4-byte parameter *streamCtr* and an 8-byte parameter *inputCtr*.