High-bandwidth Digital Content Protection System v1.3

Amendment for Gigabit Multimedia Serial Link (GMSL)

Revision 1.0

March 1, 2010

Notice

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY
PROPOSAL, SPECIFICATION OR SAMPLE. Intel Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

The cryptographic functions described in this specification may be subject to export control by the United States, Japanese, and/or other governments.

Intellectual Property

Implementation of this specification requires a license from the Digital Content Protection LLC.

Contact Information

Digital Content Protection LLC
C/O VTM, Inc.
3855 SW 153rd Drive
Beaverton, OR 97006

Email: info@digital-cp.com
Web: www.digital-cp.com

Revision History

1 Introduction

1.1 Scope

This specification describes the amendment of the High-bandwidth Digital Content Protection (HDCP) system for the Gigabit Multimedia Serial Interface (GMSL), Revision 1.00, referred to as HDCP-GMSL 1.0. It is based on the High-bandwidth Digital Content Protection (HDCP) system, Revision 1.30, referred to as HDCP 1.3.

HDCP-GMSL 1.0 is designed for protecting Audiovisual content over the GMSL interface. In an HDCP System, two or more HDCP Devices are interconnected through an HDCP-protected Interface. The Audiovisual Content protected by HDCP, referred to as HDCP Content, flows from the Upstream Content Control Function into the HDCP System at the most upstream HDCP Transmitter. From there, the HDCP Content, encrypted by the HDCP System, flows through a tree-shaped topology of HDCP Receivers over HDCP-protected Interfaces. This specification describes a content protection mechanism for: (1) authentication of HDCP Receivers to their immediate upstream connection (i.e., an HDCP Transmitter), (2) revocation of HDCP Receivers that are determined by the Digital Content Protection, LLC, to be invalid, and (3) HDCP Encryption of Audiovisual Content over the HDCP-protected Interfaces between HDCP Transmitters and their downstream HDCP Receivers. HDCP Receivers may render the HDCP Content in audio and visual form for human consumption. HDCP Receivers may be HDCP Repeaters that serve as downstream HDCP Transmitters emitting the HDCP Content further downstream to one or more additional HDCP Receivers.

Except when specified otherwise, HDCP-GMSL 1.0-compliant Devices must interoperate with other HDCP-GMSL 1.0-compliant Devices attached to their HDCP-protected Interface Ports using the same protocol. HDCP Transmitters must support HDCP Repeaters.

The state machines in this specification define the required behavior of HDCP Devices. The link-visible behavior of HDCP Devices implementing the specified state machines must be identical, even if implementations differ from the descriptions. The behavior of HDCP Devices implementing the specified state machines must also be identical from the perspective of an entity outside of the HDCP System.

Implementations must include all elements of the content protection system described herein, unless the element is specifically identified as informative or optional. Adopters must also ensure that implementations satisfy the robustness and compliance rules described in the technology license. Additionally, HDCP Transmitters may be subject to additional robustness and compliance rules associated with other content protection technologies such as Advanced Access Control System (AACS) or Digital Transmission Content Protection (DTCP).

1.2 Definitions

The following terminology, as used throughout this specification, is defined as herein:

Audiovisual Content. Audiovisual works (as defined in the United States Copyright Act as in effect on January 1, 1978), text and graphic images, are referred to as *AudioVisual Content*.

Authorized Device. An HDCP Device that is permitted access to HDCP Content is referred to as an *Authorized Device*. An HDCP Transmitter may test if an attached HDCP Receiver is an Authorized Device by successfully completing the first and, when applicable, second part of the authentication protocol. If the authentication protocol successfully results in establishing authentication, then the other device is considered by the HDCP Transmitter to be an Authorized Device.

Control Channel. The bi-directional communications channel in GMSL used to exchange control and status data when implementing the HDCP cryptographic protocol. The Control Channel operates transparently over the same twisted pair wire that carries audio and video data.

Device Key Set. Each HDCP Device has a *Device Key Set,* which consists of a set of Device Private Keys along with the associated Key Selection Vector.

Device Private Keys. A set of Device Private Keys consists of 40 different 56-bit values. These keys are to be protected from exposure outside of the HDCP Device. A set of Device Private Keys is associated with a unique Key Selection Vector.

downstream. The term, *downstream*, is used as an adjective to refer to being towards the sink of the HDCP Content stream. For example, when an HDCP Transmitter and an HDCP Receiver are connected over an HDCP-protected Interface, the HDCP Receiver can be referred to as the *downstream* HDCP Device in this connection. For another example, on an HDCP Repeater, the HDCP-protected Interface Port(s) which can emit HDCP Content can be referred to as its *downstream* HDCP-protected Interface Port(s). See also, *upstream*.

Gigabit Multimedia Serial Link (GMSL). A serial link using digital electronic signaling to transport video, audio and bi-directional control data simultaneously over a twisted pair wire.

HDCP. *HDCP* is an acronym for High-bandwidth Digital Content Protection. This term refers to this content protection system as described by any revision of this specification and its errata.

HDCP 1.3. *HDCP 1.3* refers to, specifically, the variant of HDCP described by Revision 1.30 along with its associated errata, if applicable.

HDCP 1.3-compliant Device. An HDCP Device that is designed in adherence to HDCP 1.3 is referred to as an *HDCP 1.3-compliant Device*.

HDCP Content. *HDCP Content* consists of Audiovisual Content that is protected by the HDCP System. *HDCP Content* includes the Audiovisual Content in encrypted form as it is transferred from an HDCP Transmitter to an HDCP Receiver over an HDCP-protected Interface, as well as any translations of the same content, or portions thereof. For avoidance of doubt, Audiovisual Content that is never encrypted by the HDCP System is not *HDCP Content.*

HDCP Device. Any device that contains one or more HDCP-protected Interface Port and is designed in adherence to HDCP is referred to as an *HDCP Device*.

HDCP-GMSL 1.0. HDCP-GMSL 1.0 refers to, specifically, the variant of the amendment of HDCP for GMSL described by Revision 1.0 of this specification along with its associated errata, if applicable.

HDCP-GMSL 1.0-compliant Device. A Device that is designed in adherence to HDCP-GMSL 1.0 is referred to as a *HDCP-GMSL 1.0-compliant Device*

HDCP Encryption. *HDCP Encryption* is the encryption technology of HDCP when applied to the protection of HDCP Content in an HDCP System.

HDCP-protected Interface. An interface for which HDCP applies is described as an *HDCP-protected Interface*.

HDCP-protected Interface Port. A connection point on an HDCP Device that supports an HDCP-protected Interface is referred to as an *HDCP-protected Interface Port*.

HDCP Receiver. An HDCP Device that can receive and decrypt HDCP Content through one or more of its HDCP-protected Interface Ports is referred to as an *HDCP Receiver*.

HDCP Repeater. An HDCP Device that can receive and decrypt HDCP Content through one or more of its HDCP-protected Interface Ports, and can also re-encrypt and emit said HDCP Content through one or more of its HDCP-protected Interface Ports, is referred to as an *HDCP Repeater*. An *HDCP Repeater* may also be referred to as either an HDCP Receiver or an HDCP Transmitter when referring to either the upstream side or the downstream side, respectively.

HDCP System. An *HDCP System* consists of an HDCP Transmitter, zero or more HDCP Repeaters and one or more HDCP Receivers connected through their HDCP-protected interfaces in a tree topology; whereas the said HDCP Transmitter is the HDCP Device most upstream, and receives the HDCP Content from an Upstream Content Control Function. All HDCP Devices connected to other HDCP Devices in an *HDCP System* over HDCP-protected Interfaces are part of the *HDCP System*.

HDCP Transmitter. An HDCP Device that can encrypt and emit HDCP Content through one or more of its HDCP-protected Interface Ports is referred to as an *HDCP Transmitter*.

Key Selection Vector (KSV). Each HDCP Device contains a set of Device Private Keys. A set of Device Private Keys is associated with a *Key Selection Vector* (*KSV*). Each HDCP Transmitter is assigned a unique *KSV*. Also, each HDCP Receiver is assigned a unique *KSV*.

Interrupt. An interrupt in GMSL with the input at the Receiver and the output at the Transmitter. When an interrupt occurs the Transmitter checks the *Bstatus* register on the Receiver to determine the cause of the interrupt. The Interrupt operates transparently over the same twisted pair wire that carries audio and video data.

Lock Bit. A bit in the receiver indicating the receiver is locked to the serial input, ready to receive video data and process Control Channel transactions.

upstream. The term, *upstream*, is used as an adjective to refer to being towards the source of the HDCP Content stream. For example, when an HDCP Transmitter and an HDCP Receiver are connected over an HDCP-protected Interface, the HDCP Transmitter can be referred to as the *upstream* HDCP Device in this connection. For another example, on an HDCP Repeater, the HDCP-protected Interface Port(s) which can receive HDCP Content can be referred to as its *upstream* HDCP-protected Interface Port(s). See also, *downstream*. This term should not be confused as referring to the Upstream Specification.

Upstream Content Control Function. The HDCP Transmitter most upstream in the HDCP System receives HDCP Content from the *Upstream Content Control Function*. The *Upstream Content Control Function* is not part of the HDCP System, and the methods used, if any, by the *Upstream Content Control Function* to determine for itself the HDCP System is correctly authenticated or permitted to receive the Audiovisual Content, or to transfer the Audiovisual Content to the HDCP System, are beyond the scope of this specification. On a personal computer platform, an example of an *Upstream Content Control Function* may be software designed to emit Audiovisual Content to a display or other presentation device that requires HDCP.

1.3 Overview

HDCP is designed to protect the transmission of Audiovisual Content between an HDCP Transmitter and an HDCP Receiver. The system also allows for HDCP Repeaters that support downstream HDCP-protected Interface Ports. Figure 1-1 illustrates an example connection topology for HDCP Devices. The HDCP System allows up to seven levels of HDCP Repeaters and as many as 128 total HDCP Devices, including HDCP Repeaters, to be attached to an HDCP-protected Interface Port.

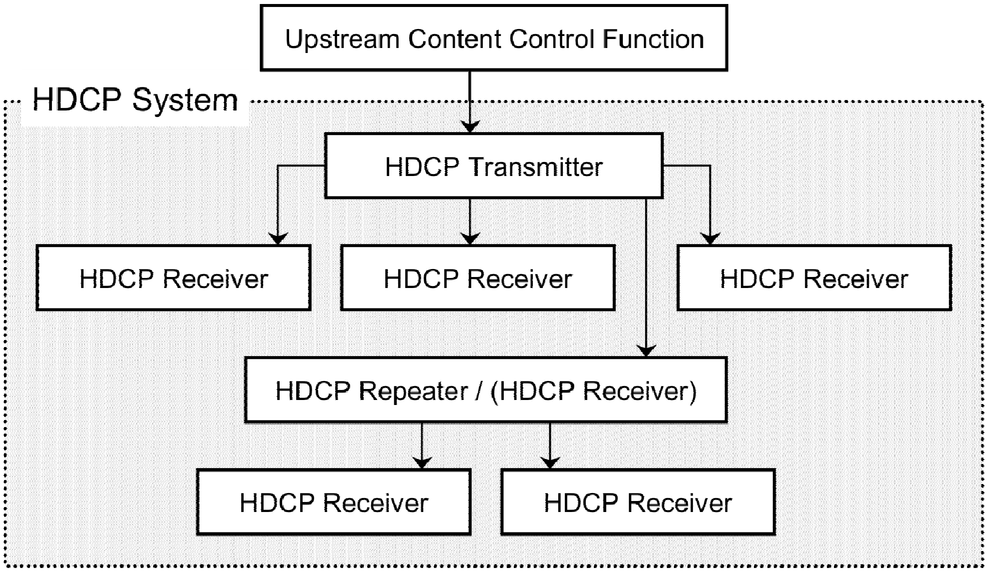


Figure 1-1. Sample Connection Topology of an HDCP System

There are three elements of the content protection system. Each element plays a specific role in the system. First, there is the authentication protocol, through which the HDCP Transmitter verifies that a given HDCP Receiver is licensed to receive HDCP Content. With the legitimacy of the HDCP Receiver determined, encrypted HDCP Content is transmitted between the two devices based on shared secrets established during the authentication protocol. This prevents eavesdropping devices from utilizing the content. Finally, in the event that legitimate devices are compromised to permit unauthorized use of HDCP Content, renewability allows a HDCP Transmitter to identify such compromised devices and prevent the transmission of HDCP Content.

This document contains chapters describing in detail the requirements of each of these elements. In addition, a chapter is devoted to describing the cipher that is used in both the authentication protocol and in the encryption of the HDCP Content.

1.4 Terminology

Throughout this specification, names that appear in italic refer to values that are exchanged during the HDCP cryptographic protocol. Names that appear in CAPS refer to status values from the receiver. C-style notation is used throughout the state diagrams and protocol diagrams, although the logic functions AND, OR, and XOR are written out where a textual description would be more clear.

The concatenation operator '||' combines two values into one. For eight-bit values *a* and *b*, the result of (*a* || *b*) is a 16-bit value, with the value *a* in the most significant eight bits and *b* in the least significant eight bits.

1.5 References

National Institute of Standards and Technology (NIST), *Digital Signature Standard (DSS)*, FIPS Publication 186-1, December 15, 1998.

National Institute of Standards and Technology (NIST), *Secure Hash Standard (SHS)*, FIPS Publication 180-1, April 17, 1995.

High-Bandwidth Digital Content Protection System, Revision 1.3, December 21, 2006, Digital Content Protection LLC

High-Bandwidth Digital Content Protection System, v1.3, Amendment for DisplayPort, Revision 1.0, Digital Content Protection LLC

2 Authentication

The HDCP Authentication protocol is an exchange between an HDCP Transmitter and an HDCP Receiver that affirms to the HDCP Transmitter that the HDCP Receiver is authorized to receive HDCP Content. This affirmation is in the form of the HDCP Receiver demonstrating knowledge of a valid set of secret device keys. Each HDCP Device is provided with a unique set of secret device keys, referred to as the Device Private Keys, from the Digital Content Protection LLC. The communication exchange, which allows for the receiver to demonstrate knowledge of such secret device keys, also provides for both HDCP Devices to generate a shared secret value that cannot be determined by eavesdroppers on this exchange. By having this shared secret formation melded into the demonstration of authorization, the shared secret can then be used as a symmetric key to encrypt HDCP Content intended only for the Authorized Device. Thus, a communication path is established between the HDCP Transmitter and HDCP Receiver that only Authorized Devices can access.

2.1 Overview

Each HDCP Device contains an array of 40, 56-bit secret device keys which make up its Device Private Keys, and a corresponding identifier, received from the Digital Content Protection LLC. This identifier is the Key Selection Vector (KSV) assigned to the device. The KSV is a 40-bit binary value.

An HDCP Device with multiple inputs can share the same keys (Receiver keys) across all its inputs. Similarly, an HDCP Device with multiple outputs can share the keys (Transmitter keys) across all its outputs.

The HDCP Authentication Protocol consists of two parts. The first part establishes shared values between the two HDCP Devices if both devices have a valid Device Key Set from the Digital Content Protection LLC. The second part allows an HDCP Repeater to report the KSVs of attached HDCP Receivers.

After successful completion of the first part of authentication, HDCP Encryption is enabled and encrypted content starts to flow between the HDCP Transmitter and the HDCP Receiver. Once encrypted content starts to flow, a periodic Link Integrity Check is performed to ensure cipher synchronization between the transmitter and the receiver. The Link Integrity Check process is explained in Section 2.2.3.

2.2 Protocol

2.2.1 First Part of Authentication Protocol

Figure 2-1 illustrates the first part of the authentication exchange. The HDCP Transmitter (*Device A*) can initiate authentication at any time, even before a previous authentication exchange has completed. Authentication is initiated by the HDCP Transmitter by sending an initiation message containing a 64-bit pseudo-random value (*An*) generated by the HDCP Cipher function hdcpRngCipher and its KSV (*Aksv*) to the HDCP Receiver (*Device B*). The HDCP Transmitter then uses the Control Channel to read the HDCP Receiver's KSV (*Bksv*) and REPEATER bit (*Bcaps[0]).* The REPEATER bit in the *Bcaps* register indicates if the receiver is an HDCP Repeater. The HDCP Transmitter verifies that the HDCP Receiver's KSV has not been revoked (Section 6), and that the received KSV contains 20 ones and 20 zeros.

The HDCP Transmitter can also initiate authentication by first reading the receiver's *Bksv* and the REPEATER bit before sending its *An* and *Aksv* value to the HDCP Receiver. However, throughout this specification it is assumed that the transmitter initiates authentication by first sending it's *An* and *Aksv* value to the HDCP Receiver.

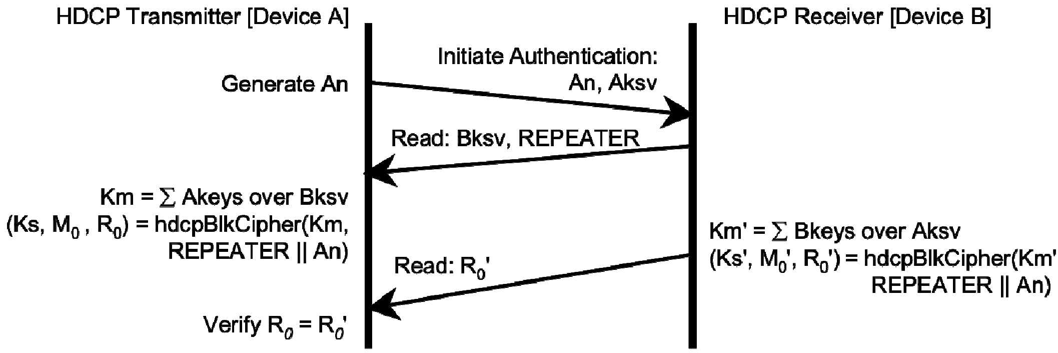


Figure 2-1. First Part of Authentication Protocol

At this point, if both HDCP Devices have a valid array of secret device keys and corresponding KSV from the Digital Content Protection LLC, then they can each calculate a 56-bit shared secret value, *Km* (*Km*' in the HDCP Receiver). Each device calculates *Km* (*Km*' in the HDCP Receiver) by adding a selection of its private device keys described by the other device's KSV, using 56-bit binary addition (i.e. unsigned addition modulo 2^{56}). The selection of secret device keys that are added together consists of those corresponding to the bit indexes of all of the 1-bits of the binary representation of the KSV.

For example, suppose *Bksv* equals 0x5A3. For the binary representation of 0x5A3, bit positions 0, 1, 5, 7, 8, and 10 are ones and all other bit positions are zeros. Therefore, *Device A* will add it's own secret device keys at array indexes 0, 1, 5, 7, 8, and 10 together to calculate the shared secret value, *Km*. *Device B* will perform an analogous calculation using its own private key array and *Aksv* to get *Km*'.

If either device has an invalid set of secret device keys or corresponding KSV, then *Km* will not be equal to *Km*'.

The HDCP Cipher function hdcpBlockCipher is then used to calculate three values, *Ks, Mo*, and *Ro*. The cipher initialization values for this calculation are *Km* (or *Km*'), and the 65-bit concatenation of REPEATER with *An*. The HDCP Receiver's status bit REPEATER indicates that the HDCP Receiver supports retransmission of HDCP Content to additional HDCP Receivers. The session key *Ks* is a 56-bit secret key for the HDCP Cipher. *Mo* is a 64-bit secret value used in the second part of the authentication protocol, and as a supplemental HDCP Cipher initialization value. *Ro*' is a 16-bit response value that the HDCP Receiver returns to the HDCP Transmitter to provide an indication as to the success of the authentication exchange. *Ro*' must be available for the HDCP Transmitter to read within 100µs from the time that the HDCP Transmitter finishes writing *Aksv* to the HDCP Receiver.

If authentication was successful, then *Ro*' will be equal to *Ro*. If there is a mismatch between *Ro* and *Ro*', the HDCP Transmitter must re-read *Ro*' for comparison against *Ro* two additional times (for a total of three consecutive comparisons) in order to account for the possibility of Control Channel errors. The authentication protocol is deemed to have failed on three consecutive mismatches between *Ro* and *Ro*'. Authentication can be reattempted with the transmission of new *An* and *Aksv* on failure of the first part of authentication.

The HDCP Transmitter enables HDCP Encryption when the first part of the authentication protocol successfully completes. Section 5 explains in detail the encryption signaling protocol that is used to enable / disable HDCP Encryption.

2.2.2 Second Part of Authentication Protocol

Figure 2-2 illustrates second part of the authentication protocol. The HDCP Transmitter executes the second part of the protocol only when the REPEATER bit is set, indicating that the attached HDCP Receiver is an HDCP Repeater. This part of the protocol assembles a list of all downstream KSVs attached to the HDCP Repeater through a permitted connection tree, enabling revocation support upstream. The second part of the authentication protocol may be implemented in parallel with the Link Integrity Check (the Link Integrity Check is explained in Section 2.2.3).

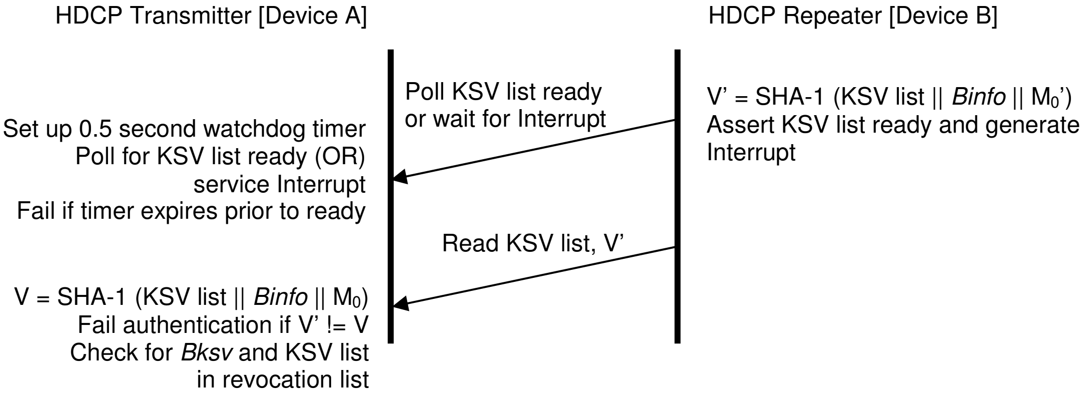

HDCP Transmitter [Device A]
HDCP Repeater [Device B]
Poll KSV list ready
or wait for Interrupt
V' = SHA-1 (KSV list || *Binfo* || M_0')
Assert KSV list ready and generate
Interrupt
Set up 0.5 second watchdog timer
Poll for KSV list ready (OR)
service Interrupt
Fail if timer expires prior to ready
Read KSV list, V'
V = SHA-1 (KSV list || *Binfo* || M_0)
Fail authentication if V' != V
Check for *Bksv* and KSV list
in revocation list

Figure 2-2. Second Part of Authentication Protocol

HDCP Repeaters assemble the list of all attached downstream HDCP Receivers as the downstream HDCP-protected Interface Ports of the HDCP Repeater complete the authentication protocol with attached HDCP Receivers. The list is represented by a contiguous set of bytes, with each KSV occupying five bytes stored in little-endian order. The total length of the KSV list is five bytes times the total number of attached and active downstream HDCP Devices, including downstream HDCP Repeaters. An HDCP-protected Interface Port with no active device attached adds nothing to the list. Also, the KSV of the HDCP Repeater itself at any level is not included in its own KSV list. An HDCP-protected Interface Port connected to an HDCP Receiver that is not an HDCP Repeater adds the *Bksv* of the attached HDCP Receiver to the list. HDCP-protected Interface Ports that have an HDCP Repeater attached add the KSV list read from the attached downstream HDCP Repeater, plus the *Bksv* of the attached downstream HDCP Repeater itself. In order to add the KSV list of the attached HDCP Repeater, it is necessary for the HDCP Repeater to verify the integrity of the list by computing *V* and checking this value against *V*' received from the attached HDCP Repeater. If *V* does not equal *V*', the downstream KSV list integrity check fails, and the HDCP Repeater must not assert its READY status and must not generate an Interrupt. Upstream HDCP Transmitters will detect this failure by the expiration of a watchdog timer set in the HDCP Transmitter.

When the HDCP Repeater has assembled the complete list of attached HDCP Devices' KSVs, it computes the verification value *V'*. This value is the SHA–1 hash of the concatenation of the KSV list, *Binfo*, and the secret value M_0'. When constructing the byte stream for the SHA-1 input, the KSV list is in the same little-endian byte order in which it is transmitted over the link, *Binfo* is appended in little-endian order, and M_0' is also appended in little-endian order. When both the KSV list and *V'* are available, the HDCP Repeater asserts its READY status indicator and generates an Interrupt.

The HDCP Transmitter, having determined that the REPEATER bit read earlier in the protocol is set, sets a half-second watchdog timer. It may either poll the HDCP Repeater's READY status bit or alternatively check the READY bit when a Interrupt is received. When READY is set, the HDCP Transmitter reads the KSV list and *V'* from the HDCP Repeater. The HDCP Transmitter verifies the integrity of the KSV list by computing the SHA–1 hash value *V* and comparing this value to *V'*. If *V* is not equal to *V'*, the HDCP Transmitter must re-read the KSV list, *Binfo* and *V'* two additional times (for a total of three consecutive *V'* checks) to account for the possibility of Control Channel errors. The authentication protocol is aborted on three consecutive mismatches between *V* and *V'* and authentication can be reattempted with the transmission of new *An* and the *Aksv*.

If the asserted READY status is not received by the HDCP Transmitter within a maximum permitted time of 0.5 seconds, authentication of the HDCP Repeater fails. With this failure, the HDCP Transmitter aborts the authentication protocol with the HDCP Repeater. Authentication can be reattempted with the transmission of a new *An* and the *Aksv*.

In addition to assembling the KSV list, an HDCP Repeater propagates topology information upward through the connection tree to the HDCP Transmitter. An HDCP Repeater reports the topology status variables DEVICE_COUNT and DEPTH. The DEVICE_COUNT for an HDCP Repeater is equal to the total number of attached downstream HDCP Receivers and HDCP Repeaters. The value is calculated as the sum of the number of attached downstream HDCP Receivers and HDCP Repeaters plus the sum of the DEVICE_COUNT read from all attached HDCP Repeaters. The DEPTH status for an HDCP Repeater is equal to the maximum number of connection levels below any of the downstream HDCP-protected Interface Ports. The value is calculated as the maximum DEPTH reported from downstream HDCP Repeaters plus one (accounting for the attached downstream HDCP Repeater).

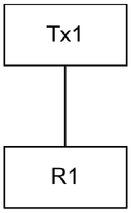


Figure 2-3. DEPTH and DEVICE_COUNT for HDCP Repeater

In Figure 2-3 above, R1 has zero downstream HDCP Devices and reports a value of zero for both the DEPTH and the DEVICE_COUNT.

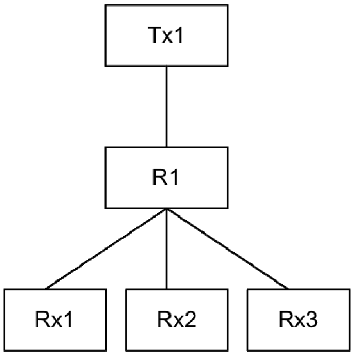


Figure 2-4. DEPTH and DEVICE_COUNT for HDCP Repeater

In Figure 2-4 above, R1 has three downstream HDCP Receivers connected to it. It reports a DEPTH of one and a DEVICE_COUNT of three.

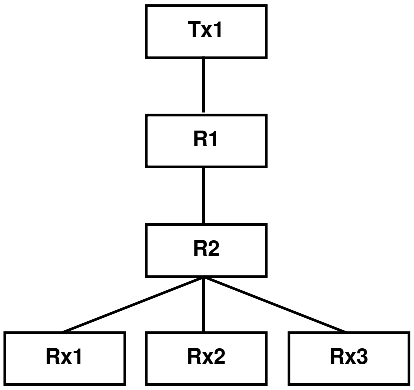


Figure 2-5. DEPTH and DEVICE_COUNT for HDCP Repeater

In Figure 2-5 above, R1 reports a DEPTH of two and a DEVICE_COUNT of four.

HDCP Repeaters must be capable of supporting DEVICE_COUNT values less than or equal to 127 and DEPTH values less than or equal to 7. If the computed DEVICE_COUNT for an HDCP Repeater exceeds 127, the HDCP Repeater must assert the MAX_DEVS_EXCEEDED status bit. If the computed DEPTH for an HDCP Repeater exceeds seven, the HDCP Repeater must assert the MAX_CASCADE_EXCEEDED status bit. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED status from a downstream HDCP Repeater, it must assert the corresponding status bits to the upstream HDCP Transmitter, assert the READY bit and generate an Interrupt.

The top-level HDCP Transmitter checks to see if the KSV of any attached device is found in the current revocation list, and, if present, the authentication fails. The HDCP Transmitter verifies the integrity of the current revocation list by checking the signature of the system renewability message (SRM) using the Digital Content Protection LLC public key. Failure of this integrity check constitutes an authentication failure.

The top-level HDCP Transmitter must complete the second phase of authentication within 1 minute after the assertion of READY by the downstream HDCP Repeater. When a new SRM version is received, the top-level HDCP Transmitter must complete SRM updates and must complete verification of KSVs of attached devices against the revocation list within 1 minute after the new SRM is received.

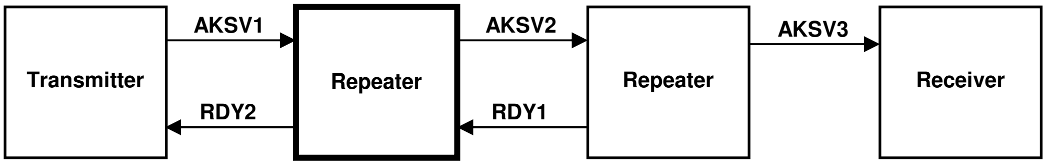
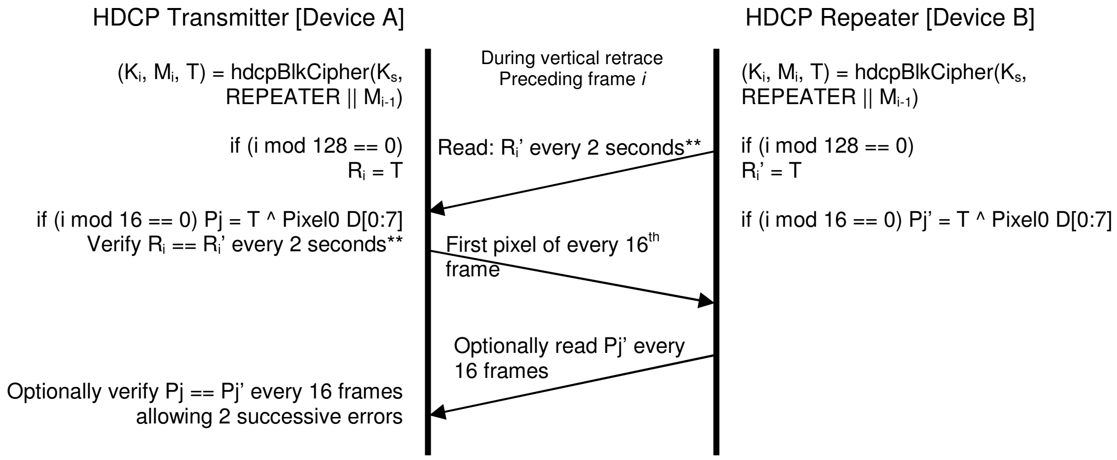


Figure 2-6. Multi-level Repeater Protocol Signals

From	To	Max Delay	Conditions and Comments
AKSV1 Upstream HDCP Transmitter *Aksv* received	AKSV2 HDCP Repeater's *Aksv* transmitted downstream	10 ms	Downstream propagation time. To latest *Aksv* transmission when more than one HDCP Receiver is attached.
AKSV3 *Aksv* transmitted to all downstream HDCP-protected Interface Ports	RDY1 Upstream READY asserted	50 ms	Upstream propagation time when no downstream HDCP Repeaters are attached. (no downstream KSV lists to process)
RDY1 Downstream READY asserted	RDY2 Upstream READY asserted	50 ms	Upstream propagation time when one or more HDCP Repeaters are attached. From latest downstream READY. (downstream KSV lists must be processed)
AKSV1 Upstream HDCP Transmitter transmits *Aksv*	RDY2 Upstream HDCP Transmitter polls asserted READY	0.42 seconds	For the Maximum of seven repeater levels, 7 * (10 ms + 50 ms)

Table 2-1. HDCP Repeater Protocol Timing Requirements

Table 2-1 specifies HDCP Repeater timing requirements that bound the worst-case propagation time for the KSV list. Note that because each HDCP Repeater does not know the number of downstream HDCP Repeaters, it must use the same half-second timeout used by the upstream HDCP Transmitter when polling for downstream READY.


HDCP Transmitter [Device A]
HDCP Repeater [Device B]
During vertical retrace Preceding frame i
(Ki, Mi, T) = hdcpBlkCipher(Ks, REPEATER || Mi-1)
(Ki, Mi, T) = hdcpBlkCipher(Ks, REPEATER || Mi-1)
if (i mod 128 == 0) Ri = T
Read: Ri' every 2 seconds**
if (i mod 128 == 0) Ri' = T
if (i mod 16 == 0) Pj = T ^ Pixel0 D[0:7]
Verify Ri == Ri' every 2 seconds**
First pixel of every 16th frame
if (i mod 16 == 0) Pj' = T ^ Pixel0 D[0:7]
Optionally read Pj' every 16 frames
Optionally verify Pj == Pj' every 16 frames allowing 2 successive errors

** Reading Ri synchronously every 128th frame is also acceptable in lieu of asynchronous polling every 2 seconds

Figure 2-7. Link Integrity Check and Enhanced Link Integrity Check

2.2.3 Link Integrity Check

After successful completion of the first part of authentication, HDCP Encryption is enabled and encrypted content starts to flow between the HDCP Transmitter and the HDCP Receiver. Once encrypted content starts to flow, a periodic Link Integrity Check or optional Enhanced Link Integrity Check is performed to maintain cipher synchronization between the HDCP Transmitter and the HDCP Receiver.

The Link Integrity Check or Enhanced Link Integrity Check, illustrated in Fig. 2-7, occurs during the vertical blanking interval preceding the frame for which it applies. Each of the two HDCP Devices calculates new cipher initialization values, K_i and M_i, and a third value R_i. The index, *i*, represents the frame number, starting with the value of one for the first video frame for which encryption is enabled after the completion of the first part of the authentication protocol, incrementing on each succeeding encrypted frame. K_i is a 56-bit key used to initialize the HDCP cipher for encryption or decryption of the HDCP Content. M_i is a new 64-bit initialization value for the HDCP cipher. R_i is a 16-bit value used for the link integrity check, and is updated for every 128th frame counter increment, starting with the 128th. The HDCP Transmitter verifies R_i' against its own calculations to insure that the video receiver is still able to correctly decrypt the information. This verification is made at a minimum rate of once every two seconds. Synchronous reading of R_i every time it changes (every 128th frame) is also acceptable in lieu of asynchronous polling. (Synchronous reading in the frame prior to R_i update and shortly after 1 millisecond of the R_i update also provides a method of detecting frame counter mismatch between HDCP Transmitter and HDCP Receiver when either device does not use the optional Enhanced Link Integrity Check). It is required that the R_i' read operation complete within 1 millisecond from the time that it is initiated by the HDCP Transmitter. Failure for any reason causes the HDCP Transmitter to consider the HDCP Receiver to be unauthenticated.

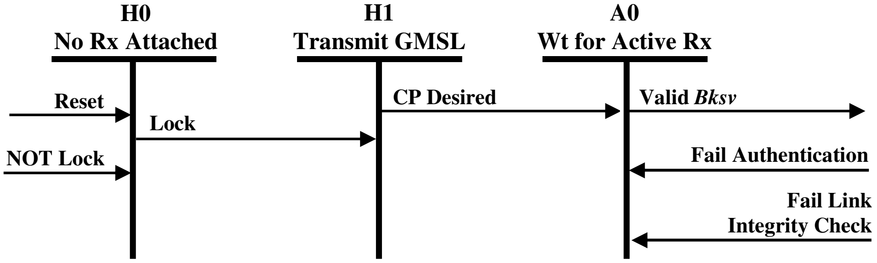
In order to enhance the detection of the loss of synchronization, the HDCP Transmitter and Receiver may optionally support the Enhanced Link Integrity Check, in which a computation to aid verification of cipher synchronization is performed when a specific video pixel is processed. Every 16th frame counter increment, the decrypted value of video data D[7:0] of the first pixel is combined with the least significant byte of R_i' using the XOR operation and is made available as P_j'. The Transmitter reads P_j' from the Receiver and compares it to its internally generated P_j. However, unless a minimum of three successive mismatches of stable values occur, this is considered to be a pixel transmission error and not an authentication or synchronization error. In addition, the mismatched P_j values must be sampled more than once in the same manner as the R_i value (see Appendix C in HDCP Rev1.3). In HDCP-GMSL Transmitters and Receivers, P_j and P_j' are always generated and available for the optional Enhanced Link Integrity Check.

2.3 HDCP Transmitter State Diagram

The HDCP Transmitter Link State Diagram and HDCP Transmitter Authentication Protocol State Diagram (Figure 2-8 and Figure 2-9) illustrate the operation states of the authentication protocol for an HDCP Transmitter that is not an HDCP Repeater. For HDCP Repeaters, the downstream (HDCP Transmitter) side is covered in Section 2.5.3.

The Transmitter's decision to begin authentication is dependent on events such as detection of an attached HDCP Receiver, availability of premium content or other implementation dependent details in the transmitter. HDCP Receivers are not required to authenticate unless the main link is initialized. When the Lock bit in a receiver is read high by the transmitter (using the Control Channel) the receiver must be ready to authenticate. In the event of authentication failure, the receiver must be prepared to process subsequent authentication attempts. In the case of an authentication failure, authentication can be reattempted with the transmission of new *An* and the *Aksv*. The HDCP Transmitter may cease to attempt authentication for transmitter-specific reasons.

The HDCP Transmitter reads HDCP registers of the receiver using Control Channel transactions. It handles HDCP register read failures (in terms of re-try attempts) in a manner consistent with other register read failures.



Note: Transition arrows with no connected state (e.g. Reset)
Indicate transitions that can occur from multiple states.

Figure 2-8. HDCP Transmitter Link State Diagram

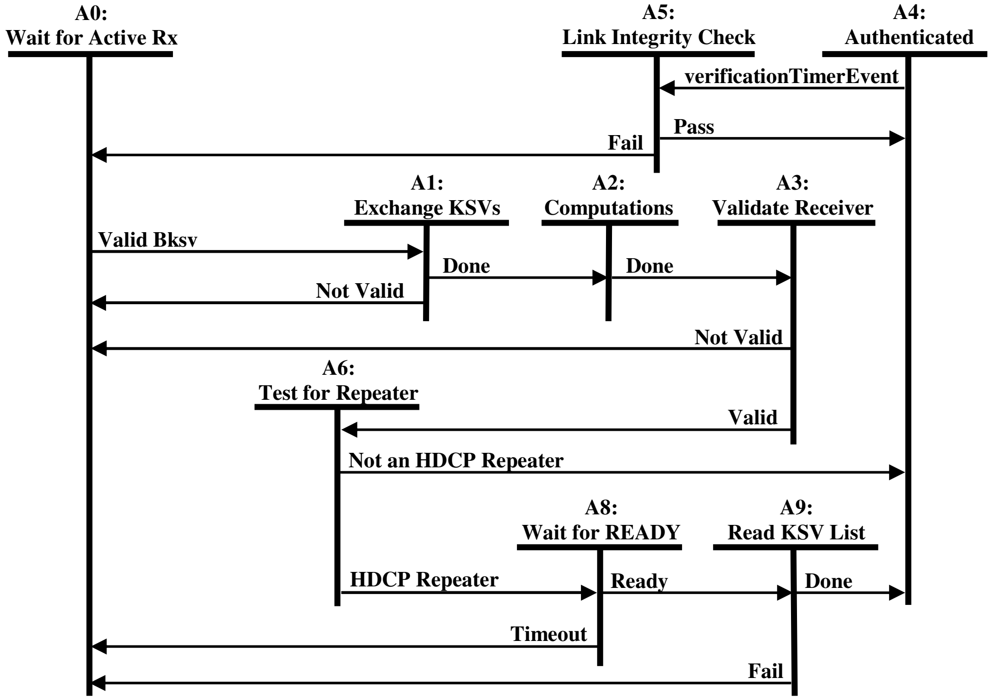


Figure 2-9. HDCP Transmitter Authentication Protocol State Diagram

Transition Any State:H0. Reset conditions at the HDCP Transmitter or failure to read the Lock bit in the high state of all HDCP capable receivers cause the HDCP Transmitter to enter the No Receiver Attached state.

Transition H0:H1. The detection of a sink device (by reading the Lock bit high) indicates to the transmitter that a sink device is attached. Reading Lock high is a sufficient indication to the transmitter that the receiver is available and active. When the receiver is no longer active, the transmitter reads Lock low.

State H1: Transmit GMSL. The transmitter should send video not requiring protection, such as an informative display, with HDCP encryption disabled. This ensures that a valid display is presented to the user before or during authentication.

Transition H1:A0. If content protection is desired by the Upstream Content Control Function, the HDCP transmitter waits for an active HDCP receiver.

State A0: Wait for Active Receiver. The transmitter continues to output video not requiring protection.

Transition A0:A1. The availability of an active HDCP Receiver through the detection of a valid *Bksv* causes authentication to begin.

State A1: Exchange KSVs. The HDCP Transmitter generates a 64-bit pseudo-random value (*An*) and writes it to the HDCP Receiver. The transmitter also writes its KSV (*Aksv)* to the receiver. The transmitter reads the HDCP Receiver's KSV (*Bksv*) and the REPEATER status bit which are necessary for cipher initialization.

Transition A1:A0. Failure to read *Bksv* containing 20 zeros and 20 ones is a protocol failure and causes a transition to State A0.

Transition A1:A2. The random value *An* and HDCP Transmitter KSV have been written, and a valid HDCP Receiver *Bksv* and REPEATER bit have been read. HDCP Transmitter has confirmed that *Bksv* contains 20 ones and 20 zeros.

State A2: Computations. In this state, the HDCP Transmitter computes the values *Km*, *Ks*, *M0,* and *R0*, using the HDCP Transmitter's Device Private Keys, *Bksv* read during State A1, and the random number *An* written to the HDCP Receiver during state A1.

Transition A2:A3. When the computed results from State A2 are available, the HDCP Transmitter proceeds to State A3.

State A3: Validate Receiver. The HDCP Transmitter reads *R0*' from the HDCP Receiver and compares it with the corresponding *R0* produced by the HDCP Transmitter during the computations of State A2. If *R0* is equal to *R0*' HDCP Encryption is enabled. The verification timer is set up to generate timer events at the nominal rate of once every two seconds, plus or minus one-half second. The HDCP Transmitter must allow the HDCP Receiver up to 100µs to make *R0*' available from the time that *Aksv* is written. The HDCP Transmitter also checks the current revocation list for the HDCP Receiver's KSV *Bksv*. If *Bksv* is in the revocation list the HDCP Receiver fails authentication. Note: checking the revocation list for *Bksv* may begin as soon as the *Bksv* has been read in State A1, asynchronously to the other portions of the protocol, but it must complete prior to the transition into the authenticated state (State A4).

The integrity of the current revocation list must be verified by checking the signature of the SRM using the Digital Content Protection LLC public key, as specified in Section 6.

Transition A3:A0. The response value *Ro'* received from the HDCP Receiver does not match the value calculated by the HDCP Transmitter or *Bksv* is in the current revocation list.

Transition A3:A6. The response value *Ro'* received from the HDCP Receiver matches the value calculated by the HDCP Transmitter and *Bksv* is not in the current revocation list.

State A4: Authenticated. The HDCP Transmitter has completed the authentication protocol. At this time, and at no time prior, the HDCP System makes available to the Upstream Content Control Function upon request, information that indicates that the HDCP System is fully engaged and able to deliver HDCP Content, which means (a) HDCP Encryption is operational on each downstream HDCP-protected Interface Port attached to an HDCP Receiver, (b) processing of valid received SRMs, if any, has occurred, as defined in this Specification, and (c) there are no HDCP Receivers on HDCP-protected Interface Ports, or downstream, with KSVs in the current revocation list.

Transition A4:A5. A verification timer event causes this transition to state A5.

State A5: Link Integrity Check. The HDCP Transmitter reads R_i' from the HDCP Receiver and compares that value against its value R_i. If the values are not equal, the HDCP Receiver is incorrectly decrypting the transmitted stream. The R_i' value may be re-read to allow for synchronization and Control Channel errors. The HDCP Transmitter may also check for a loss of synchronization with the HDCP Receiver (see Appendix C in the HDCP 1.3 specification). If the Enhanced Link Integrity Check option is used, the P_j values may be used to check for loss of synchronization more frequently than the R_i check.

Transition A5:A4. R_i' from the HDCP Receiver correctly matches the expected value, R_i .

Transition A5:A0. R_i' from the HDCP Receiver does not match the expected value, R_i , or the value was not returned to the HDCP Transmitter within 1 millisecond from the initiation of the read operation, or the loss of synchronization was detected using the R_i or P_j values.

State A6: Test for Repeater. The HDCP Transmitter evaluates the state of the HDCP Repeater capability bit (REPEATER) that was read in State A1.

Transition A6:A4. The REPEATER bit is not set (the HDCP Receiver is not an HDCP Repeater).

Transition A6:A8. The REPEATER bit is set (the HDCP Receiver is an HDCP Repeater).

State A8: Wait for Ready. The HDCP Transmitter sets up a half-second watchdog timer and either polls the HDCP Receiver's READY bit or resumes further processing while waiting for an Interrupt.

Transition A8:A0. The watchdog timer expires before the READY indication is received.

Transition A8:A9. READY is asserted and is detected by the HDCP Transmitter when polling, or When the *Bstatus* register is read while processing the Interrupt.

State A9: Read KSV List. The watchdog timer is cleared. The HDCP Transmitter reads the list of attached KSVs from the KSV memory, reads *V'*, computes *V,* and verifies $V == V'$. The KSVs from the list are compared against the current revocation list. If the size of the KSV list exceeds the capacity of the HDCP Transmitter, the authentication protocol is aborted.

The integrity of the current revocation list must be verified by checking the signature of the SRM using the Digital Content Protection LLC public key, as specified in Section 6.

Transition A9:A0. This transition is made if *V* != *V'*, verification of the SRM fails, or if any of the KSVs in the list are found in the current revocation list. If *V* is not equal to *V'*, the HDCP Transmitter must re-read the KSV list, *Binfo* and *V'* two additional times (for a total of three consecutive *V'* checks) to account for the possibility of Control Channel errors. Two additional status bits cause this transition when asserted. These are MAX_CASCADE_EXCEEDED and MAX_DEVS_EXCEEDED.

Transition A9:A4. This transition is made if *V* == *V'*, the SRM is valid, none of the reported KSVs are in the current revocation list, and the downstream topology does not exceed specified maximums.

Note that in some implementations, the trip from the point in state A3 where encryption is enabled to State A4 may be sufficiently long to miss one or more verification timer events. For improved usability, such implementations may alternatively handle the Link Integrity Check (State A5) asynchronously from the rest of the state diagram. In such cases, the transition into state A5 may occur from any state for which encryption is currently enabled. Also, the transition from state A5 returns to the appropriate state to allow for undisrupted operation.

2.4 HDCP Receiver State Diagram

The operation states of the authentication protocol for an HDCP Receiver that is not an HDCP Repeater are illustrated in Figure 2-10. For HDCP Repeaters, the upstream (HDCP Receiver) side is covered in Section 2.5.4.

The HDCP Receiver must be ready to re-authenticate with the HDCP Transmitter at any time. In particular, the only indication to the HDCP Receiver of a re-authentication attempt by the HDCP Transmitter is the reception of *An* and *Aksv* from the HDCP Transmitter.

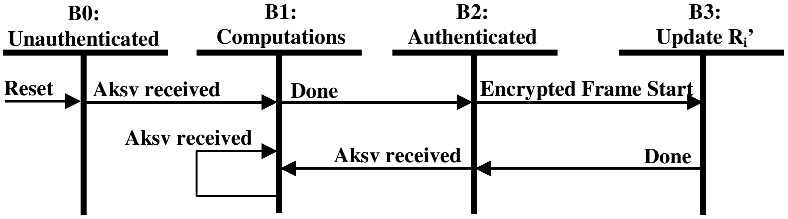


Figure 2-10. HDCP Receiver Authentication State Diagram

Transition Any State:B0. Reset conditions at the HDCP Receiver cause the HDCP Receiver to enter the unauthenticated state.

State B0: Unauthenticated. The HDCP Receiver is awaiting the reception of *An* and *Aksv* from the HDCP Transmitter to trigger the authentication protocol.

Transition B0:B1. The final byte of *Aksv* is received from the HDCP Transmitter.

State B1: Computations. In this state, the HDCP Receiver calculates the values *Km'*, *Ks'*, *M0'*, and *R0'* using the HDCP Receiver's Device Private Keys and the received values of *An* and *Aksv*. The HDCP Receiver must complete the computations within 100μs and make *R0'* available to the HDCP Transmitter.

Transition B1: B1. Should the HDCP Transmitter write a new *An* and its *Aksv* while the HDCP Receiver is in State B1, the HDCP Receiver abandons intermediate results and restarts the computations.

Transition B1:B2. The computations are complete and the results are available for reading by the HDCP Transmitter.

State B2: Authenticated. The HDCP Receiver has completed the authentication protocol and is ready to generate the first frame key when signaled by the HDCP Transmitter.

Transition B2:B1. A new authentication is forced any time a new *An* and the *Aksv* are written by the attached HDCP Transmitter.

Transition B2:B3. This transition is made during the vertical blank interval preceding encrypted frames. The Link Integrity Check protocol requires periodic updates to the R_i' value.

State B3: Update R_i'. During the vertical blank interval preceding each encrypted frame, the HDCP receiver determines whether or not to update the response value R_i' with the HDCP Cipher output value available during the frame key calculation. The R_i' value is updated when (i mod 128 == 0). The updated R_i' value must be available through the HDCP-protected Interface Port no more than 128 pixel clocks from the time that encryption enable is indicated for the next frame by the Encryption Enable (EE) bit. When the Enhanced Link Integrity Check option is used, the HDCP Receiver similarly makes P_j' available if (j mod 16 == 0) no more than 128 pixel clocks after it receives the first pixel of the frame.

Transition B3:B2. Once R_i' has been updated the receiver returns to the authenticated state.

2.5 HDCP Repeater State Diagrams

The HDCP Repeater has one HDCP-protected Interface connection to an upstream HDCP Transmitter and one or more HDCP-protected Interface connections to downstream HDCP Receivers as permitted in the Digital Content Protection LLC license. The state diagram for each downstream connection (Figure 2-13 and Figure 2-14) is substantially the same as that for the host HDCP Transmitter (Section 2.3), with two exceptions. First, the HDCP Repeater is not required to check for downstream KSVs in a revocation list. Second, the HDCP Repeater initiates authentication downstream when it receives an authentication request from upstream, rather than at detection of an HDCP Receiver on the downstream HDCP-protected Interface Port.

The HDCP Repeater signals the detection of an active downstream HDCP Receiver to the upstream HDCP Transmitter by setting the *Bstatus[1]* new device connected bit (NEW_DEV_CONN) high and generating an Interrupt.

HDCP Repeaters that have no active downstream HDCP devices must be considered. The HDCP Repeater may authenticate as an HDCP Receiver with *Bcaps* REPEATER bit set to 0 if it wishes to receive HDCP Content, but may not pass HDCP Content to downstream devices. If an HDCP Transmitter encounters a downstream HDCP Repeater reporting zero DEVICE_COUNT and sends it HDCP Content, it must complete the second phase of authentication successfully, computing V over an empty KSV list.

2.5.1 Propagation of Topology Errors

MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED: If the computed DEVICE_COUNT for an HDCP Repeater exceeds 127, the HDCP Repeater must assert the MAX_DEVS_EXCEEDED status bit. If the computed DEPTH for an HDCP Repeater exceeds seven, the HDCP Repeater must assert the MAX_CASCADE_EXCEEDED status bit. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED status from a downstream HDCP Repeater, it is required to assert the corresponding status bits to the upstream HDCP Transmitter, set the READY bit and generate an Interrupt.

2.5.2 Interrupt Propagation

An authenticated upstream connection of the HDCP Repeater becomes unauthenticated when a downstream HDCP-protected Interface Port that previously had no active downstream HDCP Receiver attached senses an attached active HDCP Receiver.

The authenticated upstream connection of the HDCP Repeater must not enter an unauthenticated state if an authenticated HDCP Receiver is disconnected from the downstream HDCP-protected Interface Port of the repeater. Also, if an authenticated HDCP Receiver attached to the downstream connection of the repeater is disconnected and reconnected (i.e. the downstream HDCP Repeater connection sees the same KSV stored in its KSV list at the HDCP protected interface port), the upstream HDCP connection of the repeater must not become unauthenticated. The downstream side is required to only re-authenticate the attached HDCP Receiver.

When an active HDCP Receiver is connected to a downstream HDCP Repeater connection that previously had no active downstream HDCP Receivers, an Interrupt must immediately be generated to the upstream HDCP Transmitter. On detecting the Interrupt, the upstream HDCP Transmitter must initiate re-authentication. When an HDCP Repeater receives an Interrupt generated by the downstream HDCP Repeater, it must immediately propagate the Interrupt upstream.

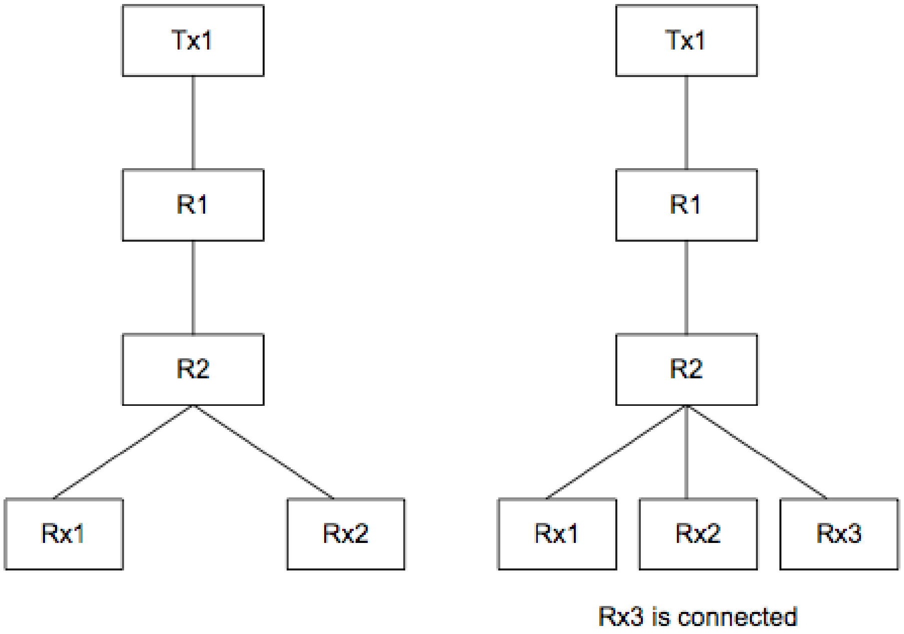


Figure 2-11. Interrupt Propagation on Connection of Active HDCP Receiver

In Figure 2-11, the devices are authenticated and HDCP Content is flowing. Connection of an active HDCP Receiver Rx3 must result in an Interrupt to Tx1. Tx1 must immediately initiate re-authentication. When an Interrupt is generated from R2 to R1, R1 must immediately propagate the Interrupt upstream to Tx1.

Unplug or re-connect of an active, authenticated HDCP Receiver attached to the downstream HDCP Repeater connection must not result in an Interrupt to the upstream HDCP Transmitter when HDCP Content is flowing. The Interrupt must be propagated to the upstream HDCP Transmitter once the flow of HDCP Content stops.

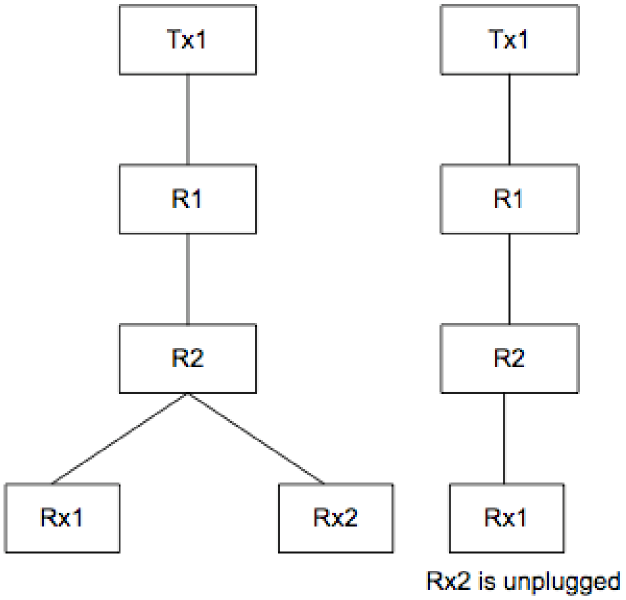


Figure 2-12. Interrupt Propagation on Unplug or Re-connect

In Figure 2-12 the devices are authenticated and HDCP Content is flowing. An Interrupt generated at R2 as a result of unplug or re-connect of Rx2 must not be propagated upstream by R2 when HDCP Content is flowing. The Interrupt must be propagated to the upstream HDCP Transmitter once the flow of HDCP Content stops.

On a new authentication request from the upstream HDCP Transmitter, the HDCP Repeater need not initiate re-authentication of all its authenticated downstream ports provided there have been no changes to the topology (i.e. the HDCP Repeater has not received an Interrupt) and all the downstream ports are either in an authenticated or unconnected state. The upstream HDCP Repeater connection may reuse the KSV list and topology information collected during the previous authentication session to complete the second part of authentication with the upstream HDCP Transmitter.

2.5.3 HDCP Repeater Downstream State Diagram

In this state diagram and its following description, the downstream (HDCP Transmitter) side refers to the HDCP Transmitter functionality within the HDCP Repeater for its corresponding downstream HDCP-protected Interface Port.

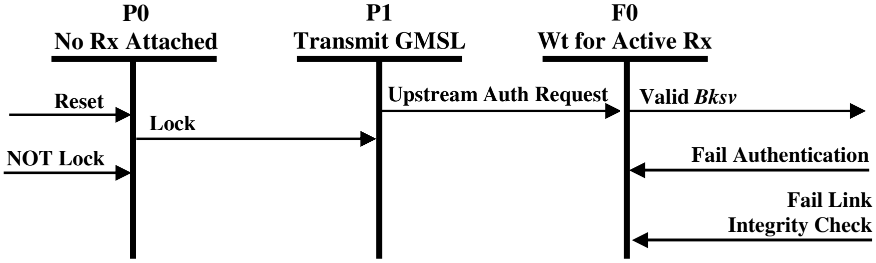


Figure 2-13. HDCP Repeater Downstream Link State Diagram

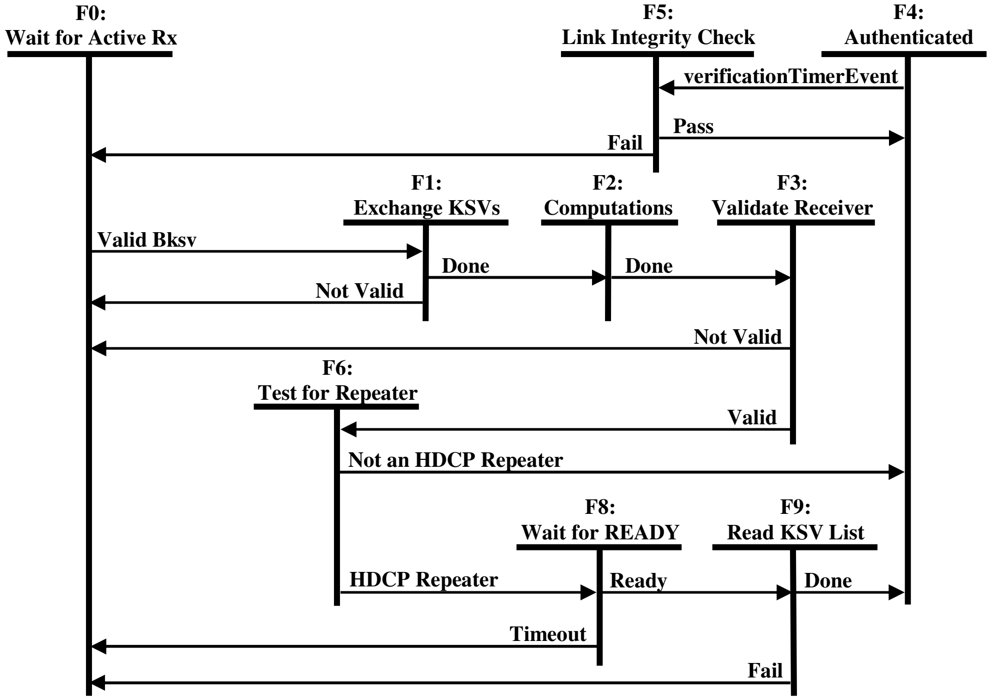


Figure 2-14. HDCP Repeater Downstream Authentication Protocol State Diagram

Transition Any State:P0. Reset conditions at the HDCP Repeater or failure to read Lock in the low of all HDCP Receivers cause the HDCP Repeater to enter the No Receiver Attached state.

Transition P0:P1. The detection of a sink device (by reading Lock high) indicates that a sink device is attached. Reading Lock high is sufficient indication that the receiver is available and active.(ready to display received content). When the receiver is no longer active, the downstream transmitter (HDCP Transmitter) reads Lock low.

State P1: Transmit GMSL. In this state the downstream side should begin sending the unencrypted video signal received from the upstream HDCP Transmitter with HDCP Encryption disabled.

Transition P1:F0. Upon an Upstream Authentication Request, the HDCP Repeater waits for the availability of an active HDCP Receiver on this port.

State F0: Wait for Active Receiver. In this state, the HDCP Repeater must not decrypt or encrypt video. If a video signal is being transmitted by the HDCP Transmitter, a valid video screen is displayed to the user. Since state F0 is reached upon an Upstream Stream Authentication Request, authentication should be started immediately by the downstream side.

Transition F0:F1. The availability of an active HDCP Receiver through the detection of a valid Bksv causes authentication to begin.

Transition F1:F2. The random value *An* and downstream side KSV have been written, and a valid HDCP Receiver *Bksv* and REPEATER bit have been read. The downstream side is required to validate that *Bksv* contains 20 ones and 20 zeros.

State F2: Computations. In this state, the downstream side computes the values *Km*, *Ks*, M_0, and R_0, using its Device Private Keys, *Bksv* read during State F1, and the random number *An* written to the HDCP Receiver during State F1.

Transition F2:F3. When the computed results from State F2 are available, the downstream side proceeds to State F3.

State F3: Validate Receiver. The downstream side reads *R0*' from the HDCP Receiver and compares it with the corresponding *R0* produced by the HDCP Transmitter during the computations of State F2. If *R0* is equal to *R0*', then HDCP Encryption is immediately enabled. The downstream side must allow the HDCP Receiver at least 100µs to make *R0*' available from the time that *Aksv* is written. The HDCP Receiver's *Bksv* is added to the KSV list for this HDCP Repeater.

Transition F3:F0. The response value *R0'* received from the HDCP Receiver does not match the value calculated by the downstream side.

Transition F3:F6. The response value *R0'* received from the HDCP Receiver matches the expected value calculated by the downstream side.

State F4: Authenticated. At this time, and at no prior time, the downstream side has completed the authentication protocol and is fully operational, able to deliver HDCP Content. The verification timer is set up to generate timer events at the nominal rate of once every two seconds, plus or minus one-half second.

Transition F4:F5. A verification timer event causes this transition to State F5.

State F5: Link Integrity Check. The downstream side reads R_i' from the HDCP Receiver and compares that value against its value R_i. If the values are equal then the HDCP Receiver is correctly decrypting the transmitted stream. The R_i' value may be re-read to allow for synchronization and Control Channel errors. The HDCP Transmitter may also check for a loss of synchronization with the HDCP Receiver (see Appendix C of the HDCP 1.3 specification). If the Enhanced Link Integrity Check option is used, the P_j values may be used to check for loss of synchronization more frequently than the R_i check.

Transition F5:F4. R_i' from the HDCP Receiver correctly matches the expected value, R_i .

Transition F5:F0. R_i' from the HDCP Receiver does not match the expected value, R_i , or the value was not returned to the downstream side within 1millisecond from the initiation of the read operation, or the loss of synchronization was detected using the R_i or P_j values.

State F6: Test for Repeater. The HDCP Repeater evaluates the state of the video repeater capability bit (REPEATER) read in State F1.

Transition F6:F4. The REPEATER bit is not set (the HDCP Receiver is not an HDCP Repeater).

Transition F6:F8. The REPEATER bit is set (the HDCP Receiver is an HDCP Repeater).

State F8: Wait for Ready. The downstream side sets up a half-second watchdog timer and either polls the HDCP Receiver's READY bit or resumes further processing while waiting for an Interrupt.

Transition F8:F0. The watchdog timer expires before the READY indication is received.

Transition F8:F9. READY is asserted and is detected by the downstream side when polling, or When the *Bstatus* register is read while processing the Interrupt.

State F9: Read KSV List. The watchdog timer is cleared. The downstream side reads the list of attached KSVs through the KSV memory, reads *V'*, computes *V*, and verifies *V* == *V'*, and the KSVs from this port are added to the KSV list for this HDCP Repeater. Additional status bits (MAX_CASCADE_EXCEEDED and MAX_DEVS_EXCEEDED) from the downstream HDCP Repeater are read and if asserted, cause the HDCP Repeater to also assert them upstream.

Transition F9:F0. This transition is made if *V* != *V'*. A retry of the entire KSV memory read operation may be implemented if *V* != *V'*. It is also made if either MAX_CASCADE_EXCEEDED or MAX_DEVS_EXCEEDED are asserted.

Transition F9:F4. This transition is made if *V* == *V'*, the downstream topology does not exceed specified maximums.

Note that in some implementations, the trip from the point in state F3 where encryption is enabled to State F4 may be sufficiently long to miss one or more verification timer events. For improved usability, such implementations may alternatively handle the Link Integrity Check (State F5) asynchronously from the rest of the state diagram. In such cases, the transition into state F5 may occur from any state for which encryption is currently enabled. Also, the transition from state F5 returns to the appropriate state to allow for undisrupted operation.

2.5.4 HDCP Repeater Upstream State Diagram

The HDCP Repeater upstream state diagram, illustrated in Figure 2-15, makes reference to states of the HDCP Repeater downstream state diagram. A link integrity check failure on a downstream HDCP protected Interface Port should not cause the upstream HDCP-protected Interface Port to move into an unauthenticated state.

In this state diagram and its following description, the upstream (HDCP Receiver) side refers to the HDCP Receiver functionality within the HDCP Repeater for its corresponding upstream HDCP protected Interface Port.

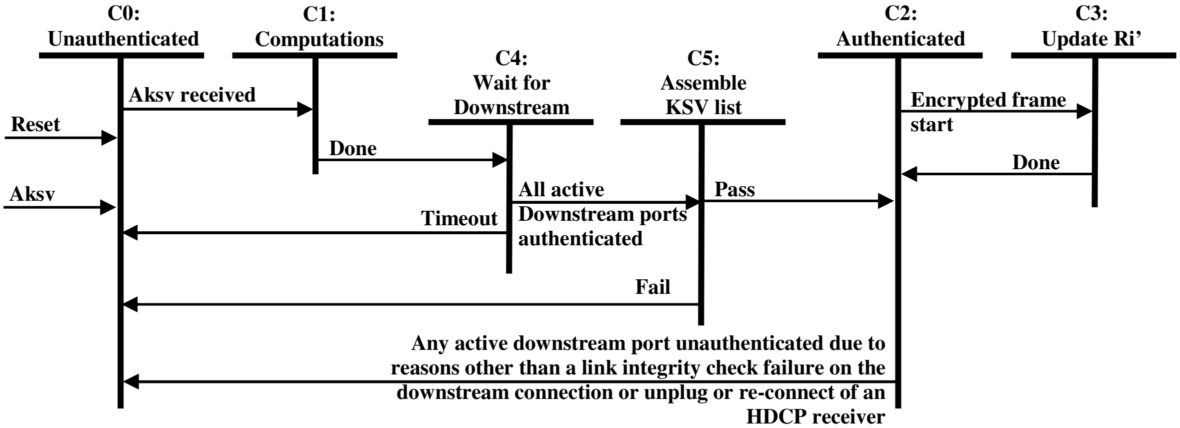


Figure 2-15. HDCP Repeater Upstream Authentication Protocol State Diagram

Transitions Any State:C0. Reset conditions at the HDCP Repeater cause the HDCP Repeater to enter the unauthenticated state. Re-authentication is forced any time the *Aksv* is written by the attached HDCP Transmitter, with a transition through the unauthenticated state.

State C0: Unauthenticated. The device is idle, awaiting the reception of *An* and *Aksv* from the HDCP Transmitter to trigger the authentication protocol. The READY status bit, in the HDCP-protected Interface Port, is de-asserted. The upstream R_i' and P_j' values are not updated in this state.

Transition C0:C1. The final byte of *Aksv* is received from the HDCP Transmitter.

State C1: Computations. In this state, the upstream (HDCP Receiver) side of the HDCP Repeater calculates the values *Km*', *Ks*', *M0*', and *R0*' using its Device Private Keys and the received values of *An* and *Aksv*. The upstream side is allowed a maximum of 100µs to complete the computations and make *R0*' available to the HDCP Transmitter. If the HDCP Transmitter writes a new *An* and *Aksv* while the HDCP Repeater is in this state (State C1), the HDCP Repeater abandons intermediate results and restarts the computations.

Transition C1:C4. The computations are complete and the results are available for reading by the HDCP Transmitter.

State C2: Authenticated. The upstream side has completed the authentication protocol and is ready to generate the first frame key when signaled by the HDCP Transmitter. The READY status bit is asserted.

Transition C2:C0. The upstream side becomes unauthenticated when any downstream side enters the unauthenticated state due to reasons other than a link integrity check failure on the downstream connection or unplug or reconnect of an active, authenticated HDCP Receiver attached to the downstream HDCP Port. When the upstream side becomes unauthenticated it signals the upstream HDCP Transmitter to initiate re-authentication by setting the NEW_DEV_CONN in the *Bstatus* register high then generating an Interrupt to the upstream HDCP-protected Interface Port.

Transition C2:C3. This transition is made during the vertical blank interval preceding encrypted frames. The Link Integrity Check requires periodic updates to the *Ri'* value.

State C3: Update R_i'. During the vertical blank interval preceding each encrypted frame, the HDCP Repeater determines whether or not to update the response value R_i' with the HDCP Cipher output value available during the frame key calculation. The R_i' value is updated when (i mod 128 == 0). The updated R_i' value must be available through the HDCP-protected Interface Port no more than 128 pixel clocks from the time that encryption enable is indicated for the next frame by the Encryption Enable (EE) bit. When the Enhanced Link Integrity Check option is used, the HDCP Repeater makes an updated P_j' available if (j mod 16 == 0) within 128 pixel clocks after it receives the first pixel of the frame.

Transition C3:C2. Once R_i' has been updated, the upstream side returns to the authenticated state.

State C4: Wait for Downstream. The upstream side state machine waits for all downstream HDCP-protected Interface Ports of the HDCP Repeater to enter either the unconnected (State P0), inactive (State F0), or the authenticated state (State F4).

Transition C4:C0. The watchdog timer expires before all downstream HDCP-protected Interface Ports enter the authenticated or unconnected state.

Transition C4:C5. All downstream HDCP-protected Interface Ports with attached HDCP Receivers have reached the state of authenticated or unconnected.

State C5: Assemble KSV List. The upstream side assembles the list of all attached downstream topology HDCP Devices as the downstream HDCP-protected Interface Ports reach terminal states of the authentication protocol. An HDCP-protected Interface Port that advances to State P0, the unconnected state, or F0, the inactive state, does not add to the list. A downstream HDCP protected Interface Port that arrives in State F4 that has an HDCP Receiver that is not an HDCP Repeater attached, adds the *Bksv* of the attached HDCP Receiver to the list. Downstream HDCP protected Interface Ports that arrive in State F4 that have an HDCP Repeater attached will cause the KSV list read from the attached HDCP Repeater, plus the *Bksv* of the attached HDCP Repeater itself, to be added to the list.

When the KSV list for all downstream HDCP Receivers has been assembled, the upstream side computes the upstream *V'*. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or MAX_CASCADE_EXCEEDED status from a downstream HDCP Repeater, it is required to assert its corresponding upstream status bit.

Transition C5:C0. If any downstream HDCP-protected Interface Port should transition to the unauthenticated state, the upstream connection transitions to the unauthenticated state. This transition is also made when the KSV list integrity check for a downstream HDCP Repeater fails.

Transition C5:C2. The KSV list and *V'*, as well as DEVICE_COUNT and DEPTH, are ready for reading by the upstream HDCP Transmitter.

2.6 HDCP Port

HDCP Transmitter and the HDCP Receiver communicate HDCP register values over the Control Channel. The HDCP Receiver and HDCP Repeaters must support these HDCP registers. Table 2-2 specifies these HDCP registers. Multi-byte values are stored in little-endian format.

Offset (hex)	Name	Size in Bytes	Rd/Wr	Function
0x80	*Bksv*	5	Rd	HDCP Receiver KSV
0x85	*R_i′ / R_o′*	2	Rd	Link verification response
0x87	*P_j′*	1	Rd	Enhanced link verification response
0x88	*An*	8	Rd / Wr	Session random number
0x90	*Aksv*	5	Rd / Wr	HDCP Transmitter KSV
0x95	*Bctrl*	1	Rd / Wr	[7:1]: Reserved [0]: EE
0x96	*Bstatus*	1	Rd / Wr	[7:2]: Reserved [1]: NEW_DEV_CONN [0]: READY
0x97	*Bcaps*	1	Rd / Wr	[7:1]: Reserved [0]: REPEATER (Receiver is a Repeater)
0x98	*Rsvd*	8	Rd	All bytes read as 0x00
0xA0	*V′.H0*	4	Rd	H0 part of SHA-1 hash value
0xA4	*V′.H1*	4	Rd	H1 part of SHA-1 hash value
0xA8	*V′.H2*	4	Rd	H2 part of SHA-1 hash value
0xAC	*V′.H3*	4	Rd	H3 part of SHA-1 hash value
0xB0	*V′.H4*	4	Rd	H4 part of SHA-1 hash value
0xB4	*Binfo*	2	Rd / Wr	[15:12]: Reserved [11]: MAX_CASCADE_EXCEEDED [10:8]: DEPTH [7]: MAX_DEVS_EXCEEDED [6:0]: DEVICE_COUNT
0xB7	Rsvd	3	Rd	All bytes read as 0x00
0xBA	KSV LIST	70	Rd / Wr	List of Key Selection Vectors of all downstream repeaters and receivers.

Table 2-2. HDCP Receiver Registers

3 Data Encryption

HDCP encryption is applied in the HDCP-GMSL Transmitter at the input of the PHY layer, before GMSL encoding and serialization. HDCP decryption is applied in the HDCP-GMSL Receiver at the output of the PHY layer, after GMSL decoding and deserialization. Using standard HDCP methods, HDCP-GMSL encryption consists of a bitwise exclusive-OR (XOR) of 25-bit or 19-bit parallel data starting with LSB of the 32-bit block of pseudo random bits provided by the HDCP cipher. The HDCP cipher output function is the same as shown in Table 4-7 of HDCP v1.3 Amendment for DisplayPort rev1.0.
Figure 3-1 is a diagram of HDCP-GMSL encryption and decryption, which is the same as Figure 3-1 of HDCP v1.3 except that the TMDS link is replaced by the GMSL link.
Encryption and decryption operations use the pixel clock as the main clock signal.

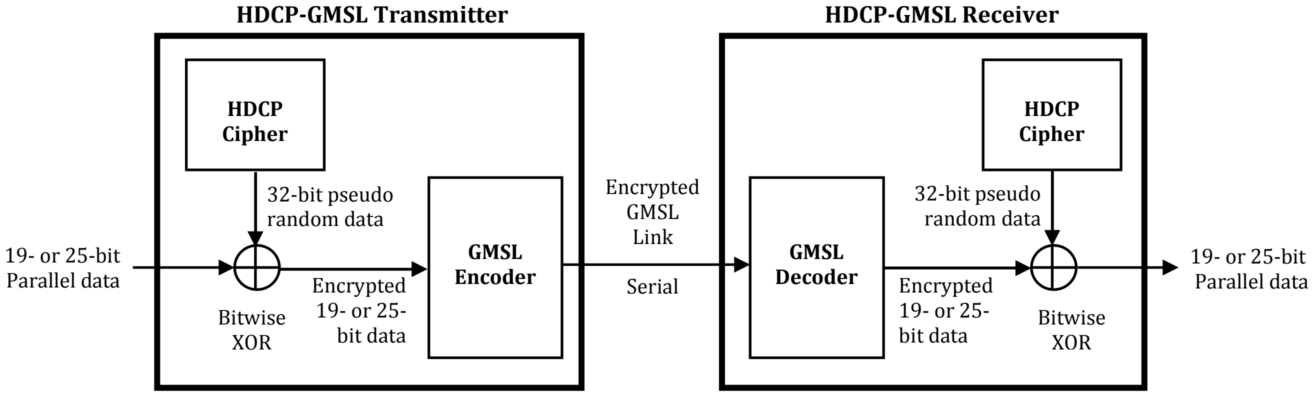

HDCP-GMSL Transmitter
HDCP Cipher
32-bit pseudo random data
19- or 25-bit Parallel data
Bitwise XOR
Encrypted 19- or 25-bit data
GMSL Encoder
Encrypted GMSL Link
Serial
HDCP-GMSL Receiver
GMSL Decoder
HDCP Cipher
32-bit pseudo random data
Encrypted 19- or 25-bit data
Bitwise XOR
19- or 25-bit Parallel data

Figure 3-1. HDCP-GMSL Encryption and Decryption

4 HDCP Cipher

HDCP-GMSL uses the cipher in HDCP v1.3 Amendment for DisplayPort rev1.0.

5 Encryption Status Signaling

After authentication, the HDCP-GMSL Transmitter sets the Encryption Enable (EE) bit high in the HDCP-GMSL Transmitter and Receiver, using the Control Channel, to enable encryption and to indicate that encryption is active.

Figure 5-1 shows a timing diagram for the HDCP-GMSL Transmitter and Receiver. The timing and function for CMO and CM1 in Figure 5-1 are the same as in Figure 5-1 of HDCP v1.3 Amendment for DisplayPort rev1.0, and refer to the HDCP cipher structure in Figure 4-1. ENC_EN is a signal on the HDCP-GMSL Transmitter and Receiver. VSYNC and HSYNC are signals from the video system seen in common by the HDCP-GMSL Transmitter and Receiver. The logic level in blanking is shown low for VSYNC and HSYNC. Events corresponding to numbers in the diagram are:

(1): A frame key is generated after first part of authentication to initialize BM1.
(2): ENC_EN goes high synchronously with the first VSYNC falling edge after EE is set high in both the HDCP-GMSL Transmitter and Receiver. The falling edge of VSYNC causes a frame re-key calculation. EE must be set high in the HDCP-GMSL Transmitter and Receiver at least one pixel clock cycle before the falling edge of VSYNC and within the same VSYNC cycle.
(3) and (5): line re-key is performed on the falling edge of HSYNC in horizontal blanking.
(4): VSYNC falling edge causes a frame re-key calculation.

For encryption disable, after the Upstream Content Control Function halts the output of content protected data, using the same timing shown in Figure 5-1, the HDCP-GMSL Transmitter sets EE low in the HDCP-GMSL transmitter and receiver and ENC_EN transitions high-to-low at point (2). When encryption is disabled the frame key and line key are not generated.

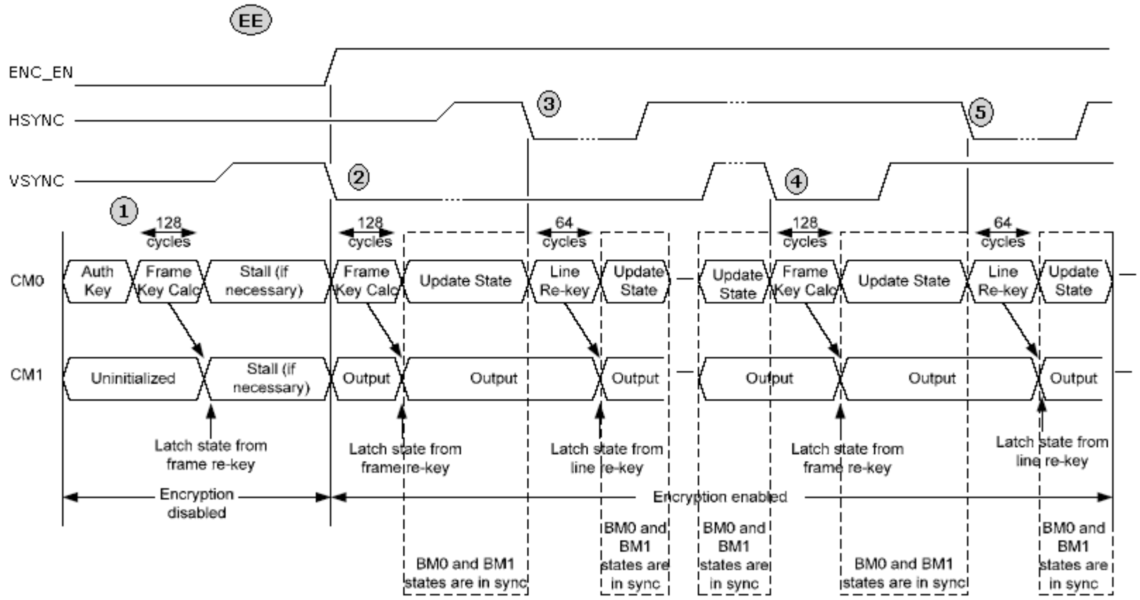


Figure 5-1. Encryption Status Signaling

6 Renewability

Renewability for HDCP-GMSL is the same as described in section 6 of the High-Bandwidth Digital Content Protection System, Revision 1.3, December 21, 2006, Digital Content Protection LLC