High-bandwidth Digital Content Protection System

Mapping HDCP to DisplayPort

Revision 2.3

22 January, 2019

Digital Content Protection LLC

Notice

Acknowledgement

STMicroelectronics and Parade Technologies have contributed to the development of this specification.

Intellectual Property

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

Contact Information

Digital Content Protection LLC
C/O Vital Technical Marketing, 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 mapping of High-bandwidth Digital Content Protection (HDCP) system to DisplayPort, Revision 2.30.

For the purpose of this specification, it is assumed that the Audiovisual content is transmitted over a DisplayPort based wired display link. In an HDCP System, two or more HDCP Devices are interconnected through an HDCP-protected Interface. The Audiovisual Content flows from the Upstream Content Control Function into the HDCP System at the most upstream HDCP Transmitter. From there the Audiovisual Content encrypted by the HDCP System, referred to as HDCP Content, 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.

Unless otherwise specified, the term "HDCP Receiver" is also used to refer to the upstream HDCP-protected interface port of an HDCP Repeater. Similarly, the term "HDCP Transmitter" is also used to refer to the downstream HDCP-protected interface port of an HDCP Repeater. HDCP Transmitters must support HDCP Repeaters.

Except when specified otherwise, HDCP 2.3-compliant Devices must interoperate with other devices compliant with HDCP 2.3 and lower that are connected to their HDCP-protected Interface Ports using the same protocol. HDCP Transmitters must support HDCP Receivers.

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 HDCP License Agreement.

Device discovery and association, and link setup and teardown, is outside the scope of this specification.

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 a connected HDCP Receiver is an Authorized Device by successfully completing the following stages of the authentication protocol – Authentication and Key Exchange (AKE) and Locality check. If the authentication protocol

successfully results in establishing authentication, then the other device is considered by the HDCP Transmitter to be an Authorized Device.

Content Stream. *Content Stream* consists of Audiovisual Content received from an Upstream Content Control Function that is to be encrypted and Audiovisual Content received from an Upstream Content Control Function that is encrypted by the HDCP System.

Device Key Set. An HDCP Receiver has a Device Key Set, which consists of its corresponding Device Secret Keys along with the associated Public Key Certificate.

Device Secret Keys. For an HDCP Transmitter, Device Secret Key consists of the secret Global Constant. For an HDCP Receiver, Device Secret Keys consists of the secret Global Constant and the RSA private key. The Device Secret Keys are to be protected from exposure outside of the HDCP Device.

downstream. The term, *downstream*, is used as an adjective to refer to being towards the sink of the HDCP Content. 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*.

Global Constant. A 128-bit random, secret constant provided only to HDCP adopters and used during HDCP Content encryption or decryption

HDCP 1.x. *HDCP 1.x* refers to, specifically, the variant of HDCP described by Revision 1.00 (referred to as HDCP 1.0), Revision 1.10 (referred to as HDCP 1.1), Revision 1.20 (referred to as HDCP 1.2) and Revision 1.30 (referred to as HDCP 1.3) along with their associated errata, if applicable.

HDCP 1.x-compliant Device. An HDCP Device that is designed in adherence to HDCP 1.x, defined above, is referred to as an *HDCP 1.x-compliant Device*.

HDCP 2. *HDCP 2* refers to, specifically, the variant of HDCP mapping for all HDCP protected interfaces described by Revision 2.00 and higher versions along with their associated errata, if applicable.

HDCP 2.0. *HDCP 2.0* refers to, specifically, the variant of HDCP mapping for all HDCP protected interfaces described by Revision 2.00 of the corresponding specifications along with their associated errata, if applicable.

HDCP 2.0-compliant Device. An HDCP Device that is designed in adherence to HDCP 2.0 is referred to as an *HDCP 2.0-compliant Device*.

HDCP 2.1. *HDCP 2.1* refers to, specifically, the variant of HDCP mapping described by Revision 2.10 of this specification along with its associated errata, if applicable.

HDCP 2.1-compliant Device. An HDCP Device that is designed in adherence to HDCP 2.1 is referred to as an *HDCP 2.1-compliant Device*.**HDCP 2.2**. *HDCP 2.2* refers to, specifically, the variant of HDCP mapping described by Revision 2.20 of this specification along with its associated errata, if applicable.

HDCP 2.3. *HDCP 2.3* refers to, specifically, the variant of HDCP mapping described by Revision 2.30 of this specification along with its associated errata, if applicable.

HDCP 2.3-compliant Device. An HDCP Device that is designed in adherence to HDCP 2.3 is referred to as an *HDCP 2.3-compliant Device*.

HDCP Cipher. The HDCP encryption module consisting of a 128-bit AES module that is operated in a Counter (CTR) mode is referred to as *HDCP Cipher*.

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

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 Session. An *HDCP Session* is established between an HDCP Transmitter and HDCP Receiver with the transmission or reception of the authentication initiation message, AKE_Init. The established HDCP Session remains valid until it is aborted by the HDCP Transmitter or a new HDCP Session is established, which invalidates the HDCP Session that was previously established, by the transmission or reception of a new AKE_Init message.

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 Audiovisual Content from one or more Upstream Content Control Functions. 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*.

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

Master Key. A 128-bit random, secret cryptographic key negotiated between the HDCP Transmitter and the HDCP Receiver during Authentication and Key Exchange and used to pair the HDCP Transmitter with the HDCP Receiver.

Public Key Certificate. Each HDCP Receiver is issued a Public Key Certificate signed by DCP LLC, and contains the Receiver ID and RSA public key corresponding to the HDCP Receiver.

Receiver Connected Indication. An indication to the HDCP Transmitter that an active receiver has been connected to it. The format of the indication or the method used by the HDCP Transmitter to connect to or disconnect from a receiver is outside the scope of this specification.

Receiver Disconnected Indication. An indication to the HDCP Transmitter that the receiver has been disconnected from it. The format of the indication or the method used by the HDCP Transmitter to connect to or disconnect from a receiver is outside the scope of this specification.

Receiver ID. A 40-bit value that uniquely identifies the HDCP Receiver. It has the same format as an HDCP 1.x KSV i.e. it contains 20 ones and 20 zeroes.

Session Key. A 128-bit random, secret cryptographic key negotiated between the HDCP Transmitter and the HDCP Receiver during Session Key exchange and used during HDCP Content encryption or decryption.

Upstream Content Control Function. The HDCP Transmitter most upstream in the HDCP System receives Audiovisual Content to be protected 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.

upstream. The term, *upstream*, is used as an adjective to refer to being towards the source of the HDCP Content. 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.*

1.3 Overview

1. HDCP is designed to protect the transmission of Audiovisual Content between an HDCP Transmitter and an HDCP Receiver. The HDCP Transmitter may support simultaneous connections to HDCP Receivers through one or more of its HDCP-protected interface ports. The system also allows for HDCP Repeaters that support downstream HDCP-protected Interface Ports. The HDCP System allows up to four levels of HDCP Repeaters and as many as 32 total HDCP Devices, including HDCP Repeaters, to be connected to an HDCP-protected Interface port.

Figure 1-1 illustrates an example connection topology for HDCP Devices.

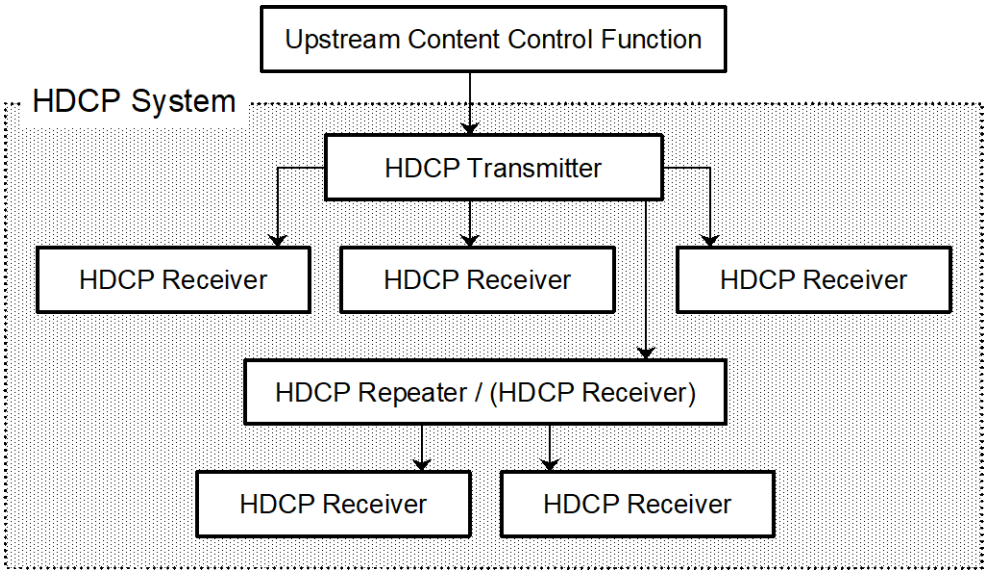


Figure 1-1. Sample Connection Topology of an HDCP System

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. The authentication protocol is implemented between the HDCP Transmitter and its corresponding downstream HDCP Receiver. 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 an 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 structure that is used in the encryption of HDCP Content.

1.4 Terminology

Throughout this specification, names that appear in italic refer to values that are exchanged during the HDCP cryptographic protocol. 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.

This specification uses the big-endian notation to represent bit strings so that the most significant bit in the representation is stored in the left-most bit position. 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

[1]. Digital Content Protection (DCP) LLC, High-bandwidth Digital Content Protection System, Revision 1.4, July 8, 2009.

[2]. Digital Content Protection (DCP) LLC, HDCP Specification 1.3 – Amendment for DisplayPort, Revision 1.0, December 19, 2006.

[3]. National Institute of Standards and Technology (NIST), *Advanced Encryption Standard (AES)*, FIPS Publication 197, November 26, 2001.

[4]. RSA Laboratories, *RSA Cryptography Standard*, PKCS #1 v2.1, June 14, 2002.

[5]. National Institute of Standards and Technology (NIST), *Secure Hash Standard (SHS)*, FIPS Publication 180-2, August 1, 2002.

[6]. Internet Engineering Task Force (IETF), *HMAC: Keyed-Hashing for Message Authentication*, Request for Comments (RFC) 2104, February 1997.

[7]. National Institute of Standards and Technology (NIST), Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90, March 2007

[8]. VESA® DisplayPort® Standard, Version 1, Revision 2a, May 23, 2012

2 Authentication Protocol

2.1 Overview

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. It is comprised of the following stages

- Authentication and Key Exchange (AKE) – The HDCP Receiver's public key certificate is verified by the HDCP Transmitter. A Master Key k_m is exchanged.
- Locality Check – The HDCP Transmitter enforces locality on the content by requiring that the Round Trip Time (RTT) between a pair of messages is not more than 16 ms.
- Session Key Exchange (SKE) – The HDCP Transmitter exchanges Session Key k_s with the HDCP Receiver.
- Authentication with Repeaters – The step is performed by the HDCP Transmitter only with HDCP Repeaters. In this step, the repeater assembles downstream topology information and forwards it to the upstream HDCP Transmitter.

Successful completion of AKE and locality check stages affirms to the HDCP Transmitter that the HDCP Receiver is authorized to receive HDCP Content. At the end of the authentication protocol, a communication path is established between the HDCP Transmitter and HDCP Receiver that only Authorized Devices can access.

All HDCP Devices contain a 128-bit secret Global Constant denoted by lc_{128}. All HDCP Devices share the same Global Constant. lc_{128} is provided only to HDCP adopters.

The HDCP Transmitter contains the 3072-bit RSA public key of DCP LLC denoted by $kpub_{dcp}$.

The HDCP Receiver is issued 1024-bit RSA public and private keys. The public key is stored in a Public Key Certificate issued by DCP LLC, denoted by $cert_{rx}$. Table 2-1 gives the fields contained in the certificate. All values are stored in big-endian format.

Name	Size (bits)	Bit position	Function
Receiver ID	40	4175:4136	Unique receiver identifier. It has the same format as an HDCP 1.x KSV i.e. it contains 20 ones and 20 zeroes
Receiver Public Key	1048	4135:3088	Unique RSA public key of HDCP Receiver denoted by $kpub_{rx}$. The first 1024 bits is the big-endian representation of the modulus n and the trailing 24 bits is the big-endian representation of the public exponent e
Reserved2	4	3087:3084	Reserved for future definition. Must be 0x0 or 0x1.
Reserved1	12	3083:3072	Reserved for future definition. Must be 0x000
DCP LLC Signature	3072	3071:0	A cryptographic signature calculated over all preceding fields of the certificate. RSASSA-PKCS1-v1_5 is the signature scheme used as defined by PKCS #1 V2.1: RSA Cryptography Standard. SHA-256 is the underlying hash function

Table 2-1. Public Key Certificate of HDCP Receiver

The secret RSA private key is denoted by $kpriv_{rx}$. The computation time of RSA private key operation can be reduced by using the Chinese Remainder Theorem (CRT) technique. Therefore, it is recommended that HDCP Receivers use the CRT technique for private key computations.

2.2 Authentication and Key Exchange

Authentication and Key Exchange (AKE) is the first step in the authentication protocol. Figure 2-1 and Figure 2-2 illustrates the AKE. The HDCP Transmitter (*Device A*) can initiate authentication at any time, even before a previous authentication exchange has completed. The HDCP Transmitter initiates a new HDCP Session by sending the authentication initiation message, AKE_Init. Message formats are defined in Section 4.2.

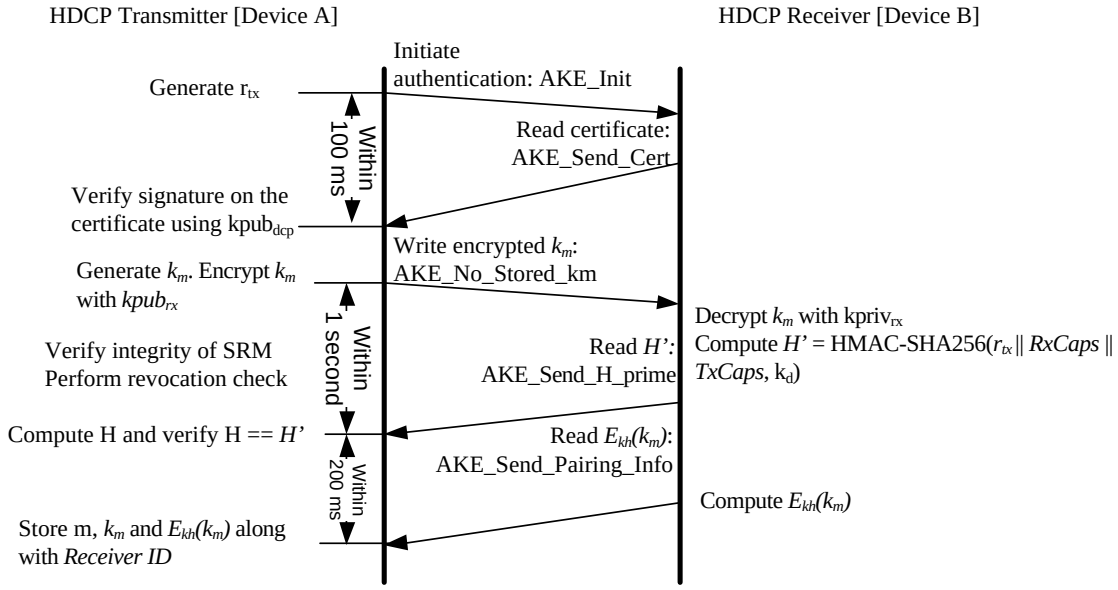


Figure 2-1. Authentication and Key Exchange (Without Stored k_m)

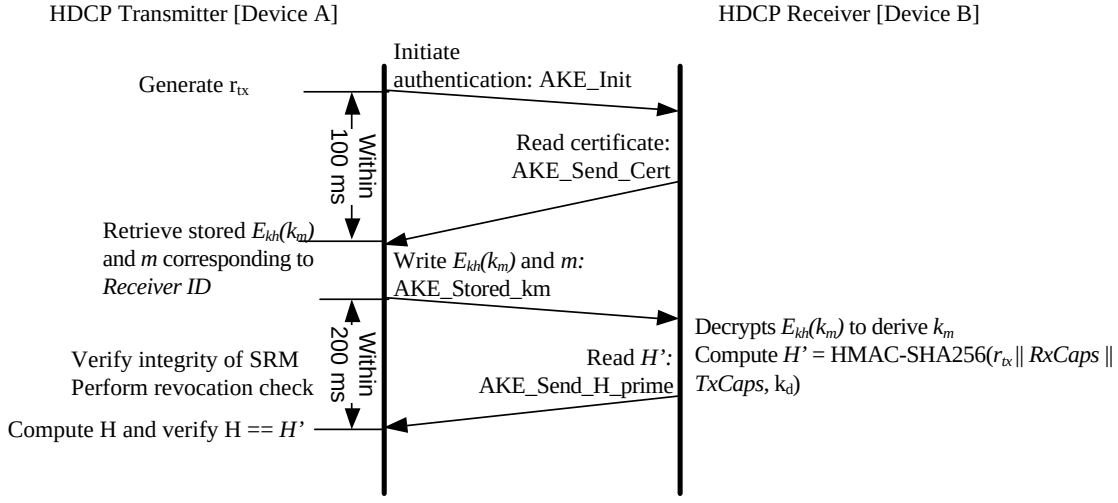


Figure 2-2. Authentication and Key Exchange (With Stored k_m)

The HDCP Transmitter

- Initiates authentication by sending the initiation message, AKE_Init, containing a 64-bit pseudo-random value (r_{tx}) and *TxCaps* parameters.

- Reads AKE_Send_Cert from the receiver containing $cert_{rx}$, a 64-bit pseudo-random value (r_{rx}) and *RxCaps*. REPEATER bit in *RxCaps* indicates whether the connected receiver is an HDCP Repeater. If REPEATER is set to one, it indicates the receiver is an HDCP Repeater. If REPEATER is zero, the receiver is not an HDCP Repeater. The AKE_Send_Cert message must be available for the transmitter to start the read within 100ms from the time the transmitter finishes writing the AKE_Init message parameters to the HDCP Receiver. The transmitter may attempt to read AKE_Send_Cert message sooner than 100ms and the receiver may respond with AUX_DEFERs until the message is ready to be read. The transmitter aborts the authentication protocol if (a) the AKE_Send_Cert message is not available for the transmitter to start the read after 100 ms or (b) the transmitter has not received the entire AKE_Send_Cert message within 110ms since the initiation of the AKE_Send_Cert message read.

- Extracts *Receiver ID* from $cert_{rx}$

    - If the HDCP Transmitter does not have a 128-bit Master Key k_m stored corresponding to the *Receiver ID* (See Section 2.2.1)

        - Verifies the signature on the certificate using $kpub_{dcp}$. Failure of signature verification constitutes an authentication failure and the HDCP Transmitter aborts the authentication protocol.

        - Generates a pseudo-random 128-bit Master Key k_m. Encrypts k_m with $kpub_{rx}$ ($E_{kpub}(km)$) and sends AKE_No_Stored_km message to the receiver containing the 1024-bit $E_{kpub}(km)$. RSAES-OAEP encryption scheme must be used as defined by PKCS #1 V2.1: RSA Cryptography Standard. SHA-256 is the underlying hash function. The mask generation function used is MGF1 which uses SHA-256 as its underlying hash function.

        - Verifies integrity of the System Renewability Message (SRM). It does this by checking the signature of the SRM using $kpub_{dcp}$. Failure of this integrity check constitutes an authentication failure and causes the HDCP Transmitter to abort authentication protocol.

          The top-level HDCP Transmitter checks to see if the *Receiver ID* of the connected device is found in the revocation list. If the *Receiver ID* of the connected HDCP Device is found in the revocation list, authentication fails and the authentication protocol is aborted. SRM integrity check and revocation check are performed only by the top-level HDCP Transmitter.

        - Performs key derivation as explained in Section 2.7 to generate 256-bit k_d. $k_d = dkey_0 \,||\, dkey_1$, where $dkey_0$ and $dkey_1$ are derived keys generated when ctr = 0 and ctr = 1 respectively. $dkey_0$ and $dkey_1$ are in big-endian order.

        - Computes 256-bit H = HMAC-SHA256($r_{tx} \,||\, RxCaps \,||\, TxCaps$, k_d) where HMAC-SHA256 is computed over $r_{tx} \,||\, RxCaps \,||\, TxCaps$ and the key used for HMAC is k_d.

- Reads the H'_AVAILABLE status bit in the *RxStatus* register as soon as it receives the CP_IRQ interrupt. If the H'_AVAILABLE status bit is set, reads the AKE_Send_H_prime message from the receiver containing the 256-bit *H'*. The CP_IRQ interrupt must be generated and the AKE_Send_H_prime message must be available for the transmitter to start the read within one second from the time the transmitter finishes writing the AKE_No_Stored_km message parameters to the HDCP Receiver. The transmitter may attempt to read AKE_Send_H_prime message sooner than one second and the receiver may respond with AUX_DEFERs until the message is ready to be read. The transmitter aborts the authentication protocol if (a) the AKE_Send_H_prime message is not available for the transmitter to start the read after one second or (b) the transmitter has not received the entire AKE_Send_H_prime message within 7ms since the initiation of the AKE_Send_H_prime message read or (c) there is a mismatch between H and H'.

o If the HDCP Transmitter has a 128-bit Master Key k_m stored corresponding to the *Receiver ID* (See Section 2.2.1)

- Sends AKE_Stored_km message to the receiver with the 128-bit $E_{kh}(k_m)$ and the 128-bit *m* corresponding to the *Receiver ID* of the HDCP Receiver

- Verifies integrity of the System Renewability Message (SRM). It does this by checking the signature of the SRM using $kpub_{dcp}$. Failure of this integrity check constitutes an authentication failure and causes the HDCP Transmitter to abort the authentication protocol.

  The top-level HDCP Transmitter checks to see if the *Receiver ID* of the connected device is found in the revocation list. If the *Receiver ID* of the connected HDCP Device is found in the revocation list, authentication fails and the authentication protocol is aborted.

- Performs key derivation as explained in Section 2.7 to generate 256-bit k_d. $k_d = dkey_0 \| dkey_1$, where $dkey_0$ and $dkey_1$ are derived keys generated when ctr = 0 and ctr = 1 respectively. $dkey_0$ and $dkey_1$ are in big-endian order.

- Computes 256-bit H = HMAC-SHA256(r_{tx}|| *RxCaps* || *TxCaps*, k_d) where HMAC-SHA256 is computed over r_{tx} || *RxCaps* || *TxCaps* and the key used for HMAC is k_d.

- Reads the H'_AVAILABLE status bit in the *RxStatus* register as soon as it receives the CP_IRQ interrupt. If the H'_AVAILABLE status bit is set, reads the AKE_Send_H_prime message from the receiver containing the 256-bit *H'*. The CP_IRQ interrupt must be generated and the AKE_Send_H_prime message must be available for the transmitter to start the read within 200ms from the time the transmitter finishes writing the AKE_Stored_km message parameters to the HDCP Receiver. The transmitter may attempt to read AKE_Send_H_prime message sooner than 200ms and the receiver may respond with AUX_DEFERs until the message is ready to be read. The transmitter aborts the authentication protocol if (a) the AKE_Send_H_prime message is not available for the transmitter to

start the read after 200ms or (b) the transmitter has not received the entire AKE_Send_H_prime message within 7ms since the initiation of the AKE_Send_H_prime message read or (c) there is a mismatch between H and H'.

The HDCP Receiver

- Makes available the AKE_Send_Cert message for the transmitter to read in response to AKE_Init. The AKE_Send_Cert message must be available for the transmitter to start the read within 100ms from the time the transmitter finishes writing the AKE_Init message parameters to the HDCP Receiver.

- If AKE_No_Stored_km is received, the HDCP Receiver

  - Decrypts k_m with $kpriv_{rx}$ using RSAES-OAEP decryption scheme.

  - Performs key derivation as explained in Section 2.7 to generate 256-bit k_d. k_d = $dkey_0$ || $dkey_1$, where $dkey_0$ and $dkey_1$ are derived keys generated when ctr = 0 and ctr = 1 respectively. $dkey_0$ and $dkey_1$ are in big-endian order.

  - Computes *H'* = HMAC-SHA256(r_{tx}|| *RxCaps* || *TxCaps*, k_d). Asserts H'_AVAILABLE bit and generates CP_IRQ as soon as it makes AKE_Send_H_prime message available for the transmitter to read. The AKE_Send_H_prime message must be available for the transmitter to start the read within one second from the time the transmitter finishes writing the AKE_No_Stored_km message parameters to the HDCP Receiver.

- If AKE_Stored_km is received, the HDCP Receiver

  - Computes 128-bit k_h = SHA-256($kpriv_{rx}$)[127:0]

  - Decrypts $E_{kh}(k_m)$ using AES with the received *m* as input and k_h as key in to the AES module as illustrated in Figure 2-3 to derive k_m.

  - Performs key derivation as explained in Section 2.7 to generate 256-bit k_d. k_d = $dkey_0$ || $dkey_1$, where $dkey_0$ and $dkey_1$ are derived keys generated when ctr = 0 and ctr = 1 respectively. $dkey_0$ and $dkey_1$ are in big-endian order.

  - Computes *H'* = HMAC-SHA256(r_{tx} || *RxCaps* || *TxCaps*, k_d). Asserts H'_AVAILABLE bit and generates CP_IRQ as soon as it makes AKE_Send_H_prime message available for the transmitter to read. The AKE_Send_H_prime message must be available for the transmitter to start the read within 200ms from the time the transmitter finishes writing the AKE_Stored_km message parameters to the HDCP Receiver.

On a decryption failure of k_m with $kpriv_{rx}$, the HDCP Receiver does not send *H'* and simply lets the timeout occur on the HDCP Transmitter.

2.2.1 Pairing

To speed up the AKE process, pairing must be implemented between the HDCP Transmitter and HDCP Receiver in parallel with AKE. When AKE_No_Stored_km message is received from the transmitter, it is an indication to the receiver that the transmitter does not have k_m stored corresponding to the receiver. In this case, after computing *H'*, the HDCP Receiver

- Computes 128-bit k_h = SHA-256($kpriv_{rx}$)[127:0].

- Generates 128-bit $E_{kh}(k_m)$ by encrypting k_m with k_h using AES as illustrated in Figure 2-3.

- Asserts PAIRING_AVAILABLE and generates CP_IRQ when AKE_Send_Pairing_Info message containing the 128-bit $E_{kh}(k_m)$ is available for the transmitter to read. This message must be available for the transmitter to start the read within 200ms from the time the transmitter finishes writing the AKE_Send_H_prime message parameters to the HDCP Receiver.

The transmitter reads the PAIRING_AVAILABLE status bit in the *RxStatus* register as soon as it receives the CP_IRQ interrupt. If the PAIRING_AVAILABLE status bit is set, the transmitter reads the AKE_Send_Pairing_Info message. The transmitter may attempt to read AKE_Send_Pairing_Info message sooner than 200ms and the receiver may respond with AUX_DEFERs until the message is ready to be read. Authentication fails and the transmitter aborts the authentication protocol if (a) the AKE_Send_Pairing_Info message is not available for the transmitter to start the read after 200ms or (b) the transmitter has not received the entire AKE_Send_Pairing_Info message within 5ms since the initiation of the AKE_Send_Pairing_Info message read. On reading AKE_Send_Pairing_Info message, the HDCP Transmitter may persistently store m (which is r_{tx} concatenated with r_{rx} ($r_{tx} \| r_{rx}$)), k_m and $E_{kh}(k_m)$ along with *Receiver ID*

Note: The HDCP Transmitter may store in its non-volatile storage m, k_m and $E_{kh}(k_m)$ along with corresponding *Receiver ID*s of all HDCP Receivers with which pairing was implemented by the HDCP Transmitter.

Figure 2-3 illustrates the encryption of k_m with k_h.

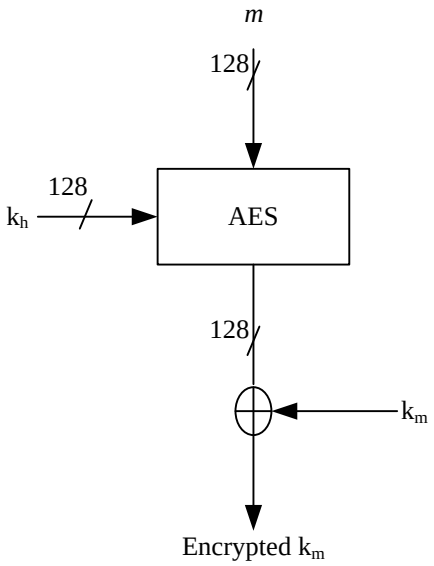


Figure 2-3. $E_{kh}(k_m)$ Computation

128-bit m is constructed by concatenating r_{tx} and r_{rx} ($r_{tx} \| r_{rx}$). Both values are in big-endian order.

2.3 Locality Check

Locality check is performed after AKE and pairing. The HDCP Transmitter initiates locality check by sending a 64-bit pseudo-random nonce r_n to the downstream receiver.

The HDCP Transmitter

- Initiates locality check by writing the LC_Init message containing a 64-bit pseudo-random nonce r_n to the HDCP Receiver.
- Sets its watchdog timer to 16ms. The LC_Send_L_prime message must be received by the transmitter within 16ms from the time the transmitter finishes writing the LC_Init message parameters to the HDCP Receiver i.e. 16ms from the time the last byte of rn has been written to the time the last byte of LC_Send_L_prime message has been received. If the LC_Send_L_prime message is not received by the transmitter within 16ms, locality check fails and the transmitter aborts the authentication protocol.
- Computes L = HMAC-SHA256(r_n , k_d XOR r_{rx}) where HMAC-SHA256 is computed over r_n and the key used for HMAC is k_d XOR r_{rx}, where r_{rx} is XORed with the least-significant 64-bits of k_d.
- On reading LC_Send_L_prime message from the receiver, compares L and *L'*. Locality check fails if L is not equal to *L'*.

An HDCP Repeater initiates locality check on all its downstream HDCP-protected interface ports by sending unique r_n values to the connected HDCP Devices.

Figure 2-4 illustrate locality check between the HDCP Transmitter and HDCP Receiver.

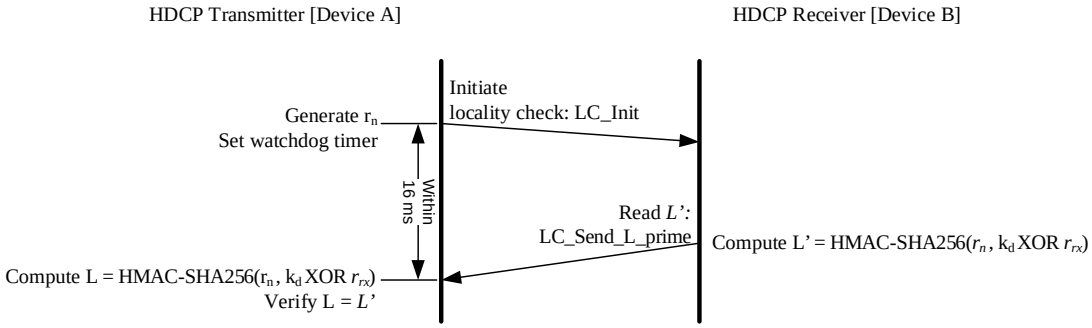


Figure 2-4. Locality Check between HDCP Transmitter and HDCP Receiver

The HDCP Receiver

- Computes a 256-bit value *L'* = HMAC-SHA256(r_n , k_d XOR r_{rx}).
- Makes LC_Send_L_prime message containing 256-bit *L'* available for the transmitter to read. The LC_Send_L_prime message must be received by the transmitter within 16ms from the time the transmitter finishes writing the LC_Init message parameters to the HDCP Receiver.

In the case of a locality check failure due to expiration of the watchdog timer or due to mismatch of L and *L'* at the HDCP Transmitter, locality check may be reattempted by the HDCP Transmitter for a maximum of 1023 additional attempts (for a maximum allowed 1024 total trials) with the transmission of an LC_Init message containing a new r_n. Failure of locality check on the first attempt and subsequent zero or more reattempts results in an authentication failure and the authentication protocol is aborted.

2.4 Session Key Exchange

Successful completion of AKE and locality check stages affirms to HDCP Transmitter that the HDCP Receiver is authorized to receive HDCP Content. Session Key Exchange (SKE) is initiated by the HDCP Transmitter after a successful locality check. The HDCP Transmitter sends encrypted Session Key to the HDCP Receiver at least 200 ms before enabling HDCP Encryption and beginning the transmission of HDCP Content. If the attached HDCP Receiver is not an HDCP Repeater, the HDCP Transmitter also writes the Type value corresponding to the Content Stream to be transmitted to the HDCP Receiver at least 200 ms before enabling HDCP Encryption.

HDCP Encryption may be enabled 200 ms after the transmission of the encrypted Session Key and Type value to the HDCP Receiver and at no time prior. Type value is written only to HDCP Receivers that are not HDCP Repeaters. Content encrypted with the Session Key ks starts to flow between the HDCP Transmitter and HDCP Receiver. HDCP Encryption must be enabled only after successful completion of AKE, locality check and SKE stages.

During SKE, the HDCP Transmitter

- Generates a pseudo-random 128-bit Session Key k_s and 64-bit pseudo-random number r_{iv}.
- Performs key derivation as explained in Section 2.7 to generate 128-bit $dkey_2$ where $dkey_2$ is the derived key when ctr =2.
- Computes 128-bit $E_{dkey}(k_s) = k_s$ XOR ($dkey_2$ XOR r_{rx}), where r_{rx} is XORed with the least-significant 64-bits of $dkey_2$.
- Writes SKE_Send_Eks message containing $E_{dkey}(k_s)$ and r_{iv} to the HDCP Receiver.

On receiving SKE_Send_Eks message, the HDCP Receiver

- Performs key derivation as explained in Section 2.7 to generate 128-bit $dkey_2$ where $dkey_2$ is the derived key when ctr =2.
- Computes $k_s = E_{dkey}(k_s)$ XOR ($dkey_2$ XOR r_{rx})

2.5 Authentication with Repeaters

The HDCP Transmitter executes authentication with repeaters after Session Key exchange and only when REPEATER bit is set, indicating that the connected HDCP Receiver is an HDCP Repeater. Authentication with repeaters stage is used for the upstream propagation of topology information and the downstream propagation of Content Stream management information as explained in Section 2.5.1 and Section 2.5.2 respectively. Authentication with repeaters may be implemented by the HDCP Transmitter in parallel with the flow of encrypted content and Link Integrity Check. The Link Integrity Check process is explained in Section 2.6.

2.5.1 Upstream Propagation of Topology Information

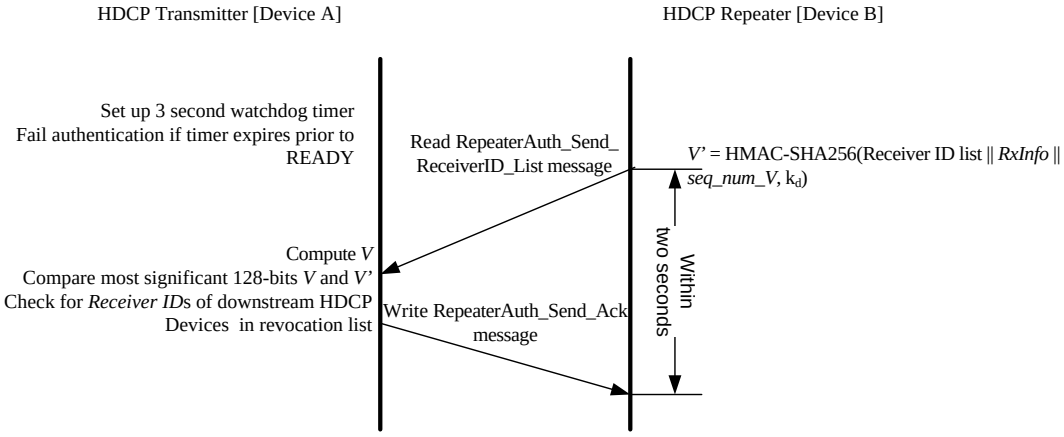


Figure 2-5. Upstream Propagation of Topology Information

Figure 2-5 illustrates the upstream propagation of topology information. This stage assembles a list of all downstream *Receiver ID*s connected to the HDCP Repeater through a permitted connection tree, enabling revocation support upstream. This stage is implemented after successful completion of Session Key Exchange. This stage is used to assemble the latest topology information at the beginning of the HDCP Session immediately following an SKE or on subsequent changes to the topology due to connect or disconnect of an HDCP Receiver or HDCP Repeater.

HDCP Repeaters assemble the list of all connected downstream HDCP Receivers as the downstream HDCP-protected Interface Ports of the HDCP Repeater successfully complete the authentication protocol with connected HDCP Receivers. The list is represented by a contiguous set of bytes, with each *Receiver ID* occupying five bytes stored in big-endian order. The total length of the Receiver ID list is five bytes times the total number of connected and active downstream HDCP Devices, including downstream HDCP Repeaters, with which the HDCP Repeater has successfully completed the authentication protocol. This total number is represented in the RepeaterAuth_Send_ReceiverID list message by the DEVICE_COUNT value. An HDCP-protected Interface Port with no active device connected adds nothing to the list. Also, the *Receiver ID* of the HDCP Repeater itself at any level is not included in its own Receiver ID list. An HDCP-protected Interface Port connected to an HDCP Receiver that is not an HDCP Repeater adds the *Receiver ID* of the connected HDCP Receiver to the list. HDCP-protected Interface Ports that have an HDCP Repeater connected add the Receiver ID list received from the connected downstream HDCP Repeater, plus the *Receiver ID* of the connected downstream HDCP Repeater itself.

When the HDCP Repeater has assembled the complete list of *Receiver ID*s of connected and active HDCP Devices with which the HDCP Repeater has successfully completed the authentication protocol, it computes the 256-bit verification value *V'*.

An HDCP Repeater and an HDCP Transmitter compute respective *V'* and *V* values as given below. HMAC-SHA256 is computed over the concatenation of Receiver ID list, *RxInfo* and *seq_num_V* received as part of the RepeaterAuth_Send_ReceiverID_List message. The key used for HMAC is k_d.

V' (or *V*) = HMAC-SHA256(Receiver ID list || *RxInfo* || *seq_num_V*, k_d)

Receiver ID list is formed by appending downstream *Receiver ID*s in big-endian order. When the Receiver ID list, *V'*, DEPTH, DEVICE_COUNT, HDCP2_LEGACY_DEVICE_DOWNSTREAM and HDCP1_DEVICE_DOWNSTREAM are

available, the HDCP Repeater asserts its READY status indicator in the *RxStatus* register and asserts the CP_IRQ interrupt.

The HDCP Transmitter checks the READY bit when a CP_IRQ interrupt is received. If READY is set, the HDCP Transmitter reads the RepeaterAuth_Send_ReceiverID_List message. The HDCP Repeater makes available the most significant 128-bits of *V'* for the transmitter to read as part of the RepeaterAuth_Send_ReceiverID_List message.

The HDCP Repeater initializes *seq_num_V* to 0 at the beginning of the HDCP Session i.e. after AKE_Init is received. It is incremented by one after the transmission of every RepeaterAuth_Send_ReceiverID_List message. *seq_num_V* must never be reused during an HDCP Session for the computation of *V* (or *V'*). If *seq_num_V* rolls over, the HDCP Transmitter must detect the roll-over in the RepeaterAuth_Send_ReceiverID_List read from the HDCP Repeater and the transmitter must disable HDCP Encryption if encryption is enabled, restart authentication by the transmission of a new AKE_Init message.

When the HDCP Repeater receives HDCP2_LEGACY_DEVICE_DOWNSTREAM or HDCP1_DEVICE_DOWNSTREAM bits that are set from a downstream HDCP Repeater, it must propagate this information to the upstream HDCP Transmitter by setting the corresponding bits in the RepeaterAuth_Send_ReceiverID_List message.

If HDCP2_LEGACY_DEVICE_DOWNSTREAM or HDCP1_DEVICE_DOWNSTREAM bit is set, the Upstream Content Control Function may instruct the most upstream HDCP Transmitter to abort the transmission of certain HDCP encrypted Type 1 Content Streams. The most upstream HDCP Transmitter must be prepared to process the request and immediately cease the transmission of specific Content Streams as instructed by the Upstream Content Control Function.

Whenever the HDCP Transmitter reads the RepeaterAuth_Send_ReceiverID_List message, it verifies the integrity of the Receiver ID list by computing *V* and comparing the most significant 128-bits of *V* and *V'*. If the values do not match, authentication fails, the authentication protocol is aborted and HDCP Encryption is disabled.

On successful verification of Receiver ID list and topology information, i.e. if the values match, none of the reported *Receiver ID*s are in the current revocation list (in the case of the most upstream HDCP Transmitter), the HDCP Transmitter does not detect a roll-over of *seq_num_V*, the downstream topology does not exceed specified maximums (explained below), the HDCP Transmitter (including downstream port of HDCP Repeater) writes the least significant 128-bits of *V* to the HDCP Repeater as part of the RepeaterAuth_Send_Ack message. Every RepeaterAuth_Send_ReceiverID_List message from the repeater to the transmitter must be followed by a RepeaterAuth_Send_Ack message from the transmitter to repeater on successful verification of Receiver ID list and topology information by the transmitter.

The RepeaterAuth_Send_Ack message must be received by the HDCP Repeater within two seconds from the time the READY status was set and the CP_IRQ interrupt was asserted by the HDCP Repeater and the downstream topology does not exceed specified maximums. A match between the least significant 128-bits of *V* and *V'* indicates successful upstream transmission of topology information. If a mismatch occurs or the RepeaterAuth_Send_Ack message is not received by the repeater within two seconds, the HDCP Repeater must transition in to an unauthenticated state, set the REAUTH_REQ status bit in the *RxStatus* register and assert CP_IRQ interrupt (See Section 2.10.3).

If the upstream HDCP Transmitter receives a CP_IRQ interrupt with REAUTH_REQ status bit set, it may initiate re-authentication with the HDCP Repeater with the transmission of a new AKE_Init message.

After transmitting the SKE_Send_Eks message, the HDCP Transmitter, having determined that REPEATER received earlier in the protocol is set, sets a three second watchdog timer. If the asserted READY status is not received by the HDCP Transmitter within a maximum-permitted time of three seconds after transmitting SKE_Send_Eks message, authentication of the HDCP Repeater fails. With this failure, the HDCP Transmitter disables HDCP Encryption and aborts the authentication protocol with the HDCP Repeater.

When an HDCP Receiver (including HDCP Repeater) is connected to the HDCP Repeater or when a connected, active HDCP Receiver with which the HDCP Repeater has successfully completed the authentication protocol is disconnected from the HDCP Repeater, the HDCP Repeater assert the READY status bit and the CP_IRQ interrupt and must make the RepeaterAuth_Send_ReceiverID_List message available for the upstream HDCP Transmitter to read. The RepeaterAuth_Send_ReceiverID_List message must include the Receiver IDs of all connected and active downstream HDCP Receivers with which the HDCP Repeater has successfully completed the authentication protocol. This enables upstream propagation of the most recent topology information after changes to the topology without interrupting the transmission of HDCP Content.

Refer to Table 2-2 for the HDCP Repeater upstream and downstream propagation time.

The 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 connected downstream HDCP Receivers and HDCP Repeaters. The value is calculated as the sum of the number of directly connected downstream HDCP Receivers and HDCP Repeaters plus the sum of the DEVICE_COUNT received from all connected 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 connected downstream HDCP Repeater).

In Figure 2-6, R1 has three downstream HDCP Receivers connected to it. It reports a DEPTH of one and a DEVICE_COUNT of three.

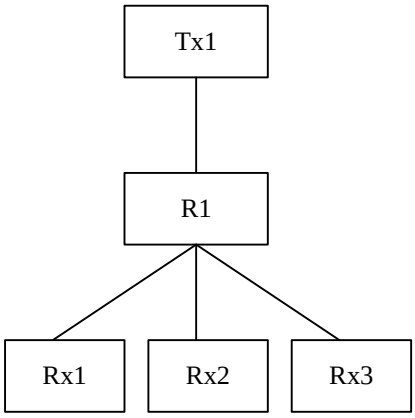


Figure 2-6. DEPTH and DEVICE_COUNT for HDCP Repeater

In Figure 2-7, R1 reports a DEPTH of two and a DEVICE_COUNT of four.

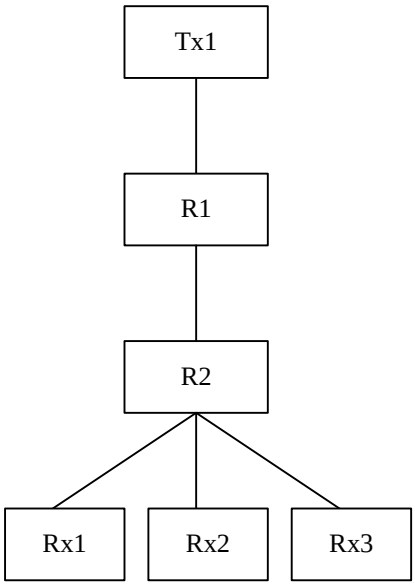


Figure 2-7. DEPTH and DEVICE_COUNT for HDCP Repeater

HDCP Repeaters must be capable of supporting DEVICE_COUNT values of up to 31 and DEPTH values of up to 4. If the computed DEVICE_COUNT for an HDCP Repeater exceeds 31, the error is referred to as MAX_DEVS_EXCEEDED error. The repeater sets MAX_DEVS_EXCEEDED bit to one in the RepeaterAuth_Send_ReceiverID_List message. If the computed DEPTH for an HDCP Repeater exceeds four, the error is referred to as MAX_CASCADE_EXCEEDED error. The repeater sets MAX_CASCADE_EXCEEDED bit to one in the RepeaterAuth_Send_ReceiverID_List message. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED error from a downstream HDCP Repeater, it must propagate the error to the upstream HDCP Transmitter, assert the READY bit and assert the CP_IRQ interrupt. If a transmitter receives these errors, it must not read the most significant 128-bits of *V'*, Receiver ID list and *seq_num_V* from the HDCP Repeater.

Authentication fails if the topology maximums are exceeded. HDCP Encryption is disabled and the authentication protocol is aborted. The top-level HDCP Transmitter, having already performed SRM integrity check during AKE, proceeds to see if the *Receiver ID* of any downstream device from the Receiver ID list is found in the current revocation list, and, if present, authentication fails, HDCP Encryption is disabled and authentication protocol is aborted.

In some instances, certain Upstream Content Control Functions may implement functionality to perform revocation checking of the downstream HDCP Receivers (including HDCP Repeaters). In such instances, and if requested by such Upstream Content Control Function, the top-level HDCP Transmitter must pass the downstream *Receiver ID*s, including the *Receiver ID* of the attached downstream HDCP Receiver or HDCP Repeater and any *Receiver ID*s received as part of the Receiver ID list, to such Upstream Content Control Function. If the top-level HDCP Transmitter receives an indication from the Upstream Content Control Function that a downstream device has been found to be revoked, the top-level HDCP Transmitter must fail authentication, disable HDCP Encryption and abort the authentication protocol.

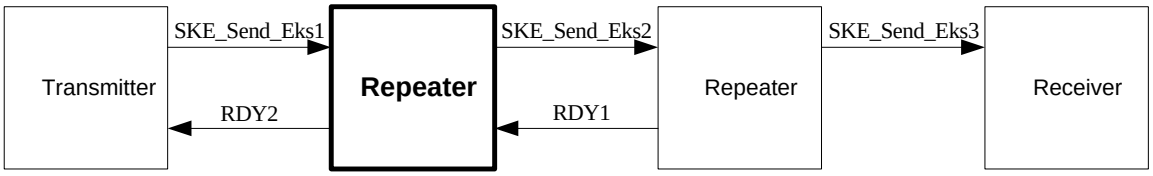


Figure 2-8. HDCP Repeater Protcol Timing Requirements

From	To	Max Delay	Conditions and Comments
SKE_Send_Eks1 Session Key received from Upstream HDCP Transmitter	SKE_Send_Eks2 k_s generated by HDCP Repeater transmitted downstream	110 ms	Downstream propagation time.
SKE_Send_Eks3 k_s transmitted to all downstream HDCP-protected Interface Ports	RDY1 Upstream READY asserted	220 ms	Upstream propagation time when no downstream HDCP Repeaters are attached (no downstream Receiver ID lists to process).
RDY1 Downstream *Receiver ID*s and topology information received	RDY2 Upstream READY asserted	220 ms	Upstream propagation time when one or more HDCP Repeaters are attached. From latest downstream READY (downstream Receiver ID lists must be processed).
SKE_Send_Eks1 Upstream HDCP Transmitter transmits k_s	RDY2 Upstream transmitter receives CP_IRQ with READY asserted	1.32 seconds	For the Maximum of four repeater levels, 4 * (110 ms + 220 ms).

Table 2-2. HDCP Repeater Protocol Timing Requirements

Table 2-2 specifies HDCP Repeater timing requirements that bound the worst-case propagation time for the Receiver ID list. A maximum delay of three seconds, to receive the RepeaterAuth_Send_ReceiverID_List message by the upstream transmitter, has been provided after transmission of the SKE_Send_Eks message to account for authentication delays due to the presence of downstream receivers that have not been paired with the upstream HDCP Repeater. Note that because each HDCP Repeater does not know the number of downstream HDCP Repeaters, it must use the same three-second timeout used by the upstream HDCP Transmitter for receiving the RepeaterAuth_Send_ReceiverID_List message.

2.5.2 Downstream Propagation of Content Stream Management Information

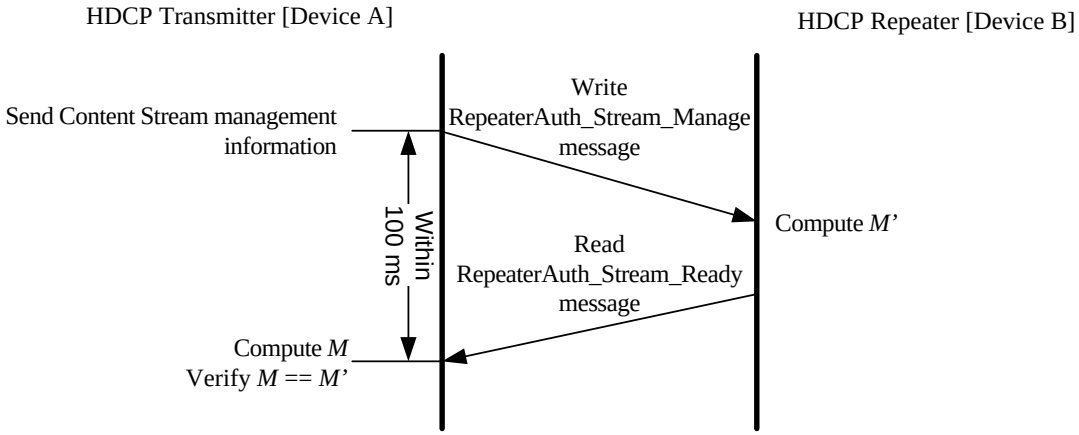


Figure 2-9. Downstream Propagation of Content Stream Management Information

The HDCP Transmitter may transmit multiple Content Streams to an HDCP Receiver during an HDCP Session. The HDCP Transmitter may use the same Session Key, k_s, negotiated during the HDCP Session for HDCP Encryption of the Content Streams.

The HDCP Transmitter propagates Content Stream management information, which includes Type values assigned to Content Streams, using the RepeaterAuth_Stream_Manage message to the attached HDCP Repeater. The HDCP Transmitter executes this step after successful completion of Session Key Exchange and before beginning the transmission of a Content Stream after HDCP Encryption to the HDCP Repeater. The RepeaterAuth_Stream_Manage message from an HDCP Transmitter to the attached HDCP Repeater identifies restrictions, as specified by the Upstream Content Control Function, on the transmission of Content Streams to specific devices.

Type values are assigned to all Content Streams by the most upstream HDCP Transmitter based on instructions received from the Upstream Content Control Function. The exact mechanism used by the Upstream Content Control Function to instruct the HDCP Transmitter is outside the scope of this specification. Type 0 Content Streams (see Section 4.2.12) may be transmitted by the HDCP Repeater to all HDCP Devices. Type 1 Content Streams (see Section 4.2.12) must not be transmitted by the HDCP Repeater through its HDCP-protected Interface Ports connected to HDCP 1.x-compliant Devices, HDCP 2.0-compliant Devices, and HDCP 2.1-compliant Devices.

The HDCP Transmitter must write the RepeaterAuth_Stream_Manage message specifying Type values assigned to Content Streams, to the attached HDCP Repeater at least 110ms before the transmission of the corresponding Content Streams after HDCP Encryption. The HDCP Transmitter must only send the RepeaterAuth_Stream_Manage message corresponding to encrypted Content Streams it will transmit to the HDCP Repeater. The HDCP Transmitter initializes *seq_num_M* to 0 at the beginning of the HDCP Session i.e. after AKE_Init is sent. It is incremented by one after the transmission of every RepeaterAuth_Stream_Manage message.

On receiving the RepeaterAuth_Stream_Manage message, the HDCP Repeater computes *M'* as given below. HMAC-SHA256 is computed over the concatenation of *StreamID_Type* (see Section 4.2.12) and *seq_num_M* values received as part of the RepeaterAuth_Stream_Manage message. All values are in big-endian order. The key used for HMAC is SHA256(k_d). *seq_num_M* must never be reused during an HDCP Session for the computation of *M'* (or *M*). If *seq_num_M* rolls over, the HDCP Transmitter must disable HDCP Encryption if encryption is enabled, restart authentication by the transmission of a new AKE_Init message.

*M' (*or *M)* = HMAC-SHA256(*StreamID_Type* || *seq_num_M*, SHA256(k_d)).

M' must be made available by the HDCP Repeater for the HDCP Transmitter to read as part of the RepeaterAuth_Stream_Ready message.

The RepeaterAuth_Stream_Ready message must be available for the transmitter to start the read within 100 ms from the time the transmitter finishes writing the RepeaterAuth_Stream_Manage message parameters to the HDCP Receiver. Every RepeaterAuth_Stream_Manage message from the transmitter to the repeater must be followed by a RepeaterAuth_Stream_Ready message from the repeater to the transmitter.

When the RepeaterAuth_Stream_Ready message is read, the HDCP Transmitter verifies the integrity of the message by computing *M* and comparing this value to *M'*. If *M* is equal to *M'*, the HDCP Transmitter may transmit the Content Streams identified in the corresponding RepeaterAuth_Stream_Manage message. The transmitter may attempt to read RepeaterAuth_Stream_Ready message sooner than 100ms and the receiver may respond with AUX_DEFERs until the message is ready to be read. The HDCP Transmitter must not transmit the Content Streams identified in the corresponding RepeaterAuth_Stream_Manage message if (a) the RepeaterAuth_Stream_Ready message is not available for the transmitter to start the read after 100ms or (b) the transmitter has not received the entire RepeaterAuth_Stream_Ready message within 7ms since the initiation of the RepeaterAuth_Stream_Ready message read or (c) if *M* is not equal to *M'*. Type value is assigned to each Content Stream through the successful transmission/reception of a single RepeaterAuth_Stream_Manage message. The Content Stream shall be associated with such Type value throughout the HDCP Session.

An HDCP Repeater connected to an HDCP 2.0-compliant Transmitter or an HDCP 1.x-compliant Transmitter will not receive the RepeaterAuth_Stream_Manage message from the transmitter. In this case, the HDCP Repeater must assign a Type value of 0x00 to all Content Streams received from the HDCP Transmitter.

The HDCP Repeater must in turn propagate the received Content Stream management information using the RepeaterAuth_Stream_Manage message further downstream.

2.6 Link Integrity Check

After successful completion of SKE, HDCP Encryption is enabled and encrypted content starts to flow between the HDCP Transmitter and the HDCP Receiver. Once encryption is enabled, a periodic Link Integrity Check is performed to maintain cipher synchronization between the HDCP Transmitter and the HDCP Receiver.

2.6.1 Link Integrity Check in MST Mode

To perform link integrity check in the MST mode, two MTPH timeslots immediately following SR are used to transmit a known 16-bit pattern, 0x531F, from the transmitter to the receiver. This pattern is referred to as LINK_VERIFICATION_PATTERN and is transmitted least significant byte first. The LINK_VERIFICATION_PATTERN is duplicated per lane on 2-lane and 4-lane Main Links. The transmitter sets the two MTPH timeslots following a given SR symbol to the corresponding byte of the pattern, encrypts the MTPHs with the Type input to the HDCP Cipher set to 0x00 (Refer to Section 3.2) and sends the MTPHs to the receiver. The receiver decrypts the MTPHs and compares the decrypted byte values to the corresponding byte in the LINK_VERIFICATION_PATTERN. If the received pattern, which is transmitted least significant byte first, matches the LINK_VERIFICATION_PATTERN at the receiver, it indicates that the ciphers are in sync. An error is determined to have occurred if the decrypted byte does not match the corresponding byte in the LINK_VERIFICATION_PATTERN. No error correction techniques (e.g., majority voting) should be applied to the MTPH timeslots used to transmit the LINK_VERIFICATION_PATTERN. HDCP Encryption is only applied to the MTPH timeslots

used to transmit LINK_VERIFICATION_PATTERN; other MTPH timeslots must not be encrypted.

A link integrity failure is determined to have occurred if pattern mismatches at the receiver are detected for two successive link frame periods. Two successive link frame periods are checked to enable recovery from simple transient synchronization errors (e.g., random bit error bursts). If a failure is detected within two successive link frames then the receiver has experienced a non-recoverable loss of cipher synchronization.

On detecting an unrecoverable loss of cipher synchronization, the HDCP Receiver must assert the LINK_INTEGRITY_FAILURE bit in the *RxStatus* register and generate a CP_IRQ interrupt. On receiving a CP_IRQ interrupt, the HDCP Transmitter is required to read the *RxStatus* register to determine the cause of the interrupt. The HDCP Transmitter must disable HDCP Encryption at the SR boundary as soon as feasible after receiving the CP_IRQ interrupt from the HDCP Receiver if the LINK_INTEGRITY_FAILURE bit is set and must initiate re-authentication with the transmission of a new AKE_Init message.

2.6.2 Link Integrity Check in SST Mode

To perform link integrity check in SST mode, Bit 5 of VB-ID is used to transmit a known 16-bit pattern, 0x531F, from the transmitter to the receiver one bit at a time. This pattern is referred to as LINK_VERIFICATION_PATTERN. The VB-ID is transmitted on all lanes after every BS/SR/CPBS/CPSR symbol, as described in the DisplayPort Specification. It is transmitted once per lane for 4-lane Main Link, twice per lane for 2-lane Main Link, and four times for 1-lane Main Link, resulting in a total of four VB-ID's following each CPBS/CPSR. The LINK_VERIFICATION_PATTERN is continuously and repeatedly transmitted least significant bit first. All four VB-ID's following each CPBS/CPSR shall carry the same Bit 5 value. After every CPSR symbol the pattern transmission is restarted, inserting the LSB of the pattern in all VB-IDs associated with the CPSR symbol. The transmitter sets Bit 5 of the VB-ID symbol associated with a given CPBS/CPSR symbol to the corresponding pattern bit value, encrypts the VB-ID with the Type input to the HDCP Cipher set to the Type value corresponding to the Content Stream to be encrypted (Refer to Section 3.2) and sends the VB-ID to the receiver. The receiver decrypts the VB-ID and compares Bit 5 of VB-ID to the corresponding bit value in the LINK_VERIFICATION_PATTERN. If the received pattern, which is transmitted one bit at a time, matches the LINK_VERIFICATION_PATTERN at the receiver, it indicates that the ciphers are in sync. An error is determined to have occurred if the bit pattern in any of the VB-ID symbols is found to not match the expected bit of the LINK_VERIFICATION_PATTERN. No error correction techniques (e.g., majority voting) should be applied to Bit 5 of the VB-ID symbols associated with a given CPBS/CPSR symbol.

A link integrity failure is determined to have occurred if three consecutive pattern mismatches at the receiver (in 16 * 3 = 48 VB-ID transmissions) are detected within two successive frame periods. Two successive frame periods are checked to enable recovery from simple transient synchronization errors (e.g., random bit error bursts). If a failure is detected within two successive frames then the receiver has experienced a non-recoverable loss of cipher synchronization. The state machine shown in Figure 2-10 illustrates the expected HDCP Receiver link integrity check behavior.

On detecting an unrecoverable loss of cipher synchronization (e.g., transition from "Check 2nd Frame" to "Disable Pending" in Figure 2-10), the HDCP Receiver must assert the LINK_INTEGRITY_FAILURE bit in the *RxStatus* register and generate a CP_IRQ interrupt. On receiving a CP_IRQ interrupt, the HDCP Transmitter is required to read the *RxSstatus* register to determine the cause of the interrupt. The HDCP Transmitter must disable HDCP Encryption at the CPSR/SR transmission boundary as soon as feasible after receiving the CP_IRQ interrupt from the HDCP Receiver if the LINK_INTEGRITY_FAILURE bit is set and must initiate re-authentication with the transmission of a new AKE_Init message.

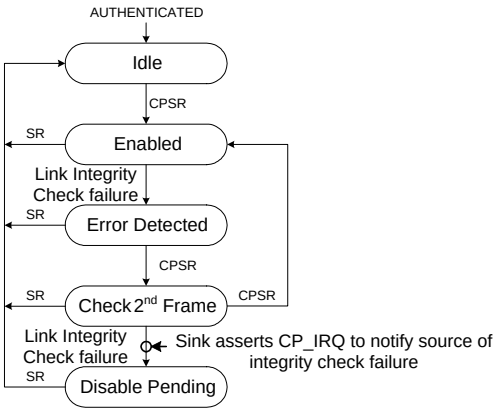


Figure 2-10. HDCP Receiver Link Integrity Check State Machine

2.7 Key Derivation

Key derivation is illustrated in Figure 2-11.

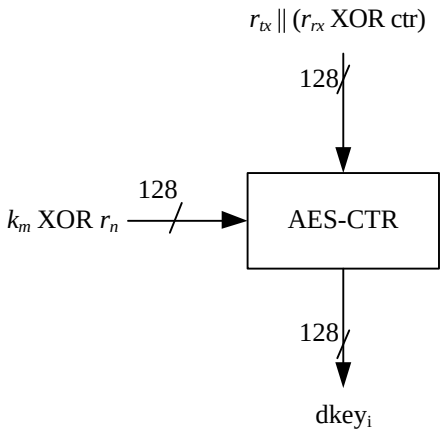


Figure 2-11. Key Derivation

r_{tx} is concatenated with r_{rx} XOR ctr (r_{tx} || (r_{rx} XOR ctr)). All values are in big-endian order. ctr is a 64-bit counter and is initialized to 0 at the beginning of the HDCP Session i.e. after AKE_Init is sent or received. It is incremented by one after every derived key computation. dkey$_i$ is the 128-bit derived key when ctr = i. ctr must never be reused during an HDCP Session.

r_n is initialized to 0 during AKE i.e. during the generation of dkey$_0$ and dkey$_1$. It is set to a pseudo-random value during locality check as explained in Section 2.3. The pseudo-random r_n is XORed with the least-significant 64-bits of k_m during generation of dkey$_2$.

2.8 HDCP Transmitter State Diagram

As explained in Section 1.3, the HDCP Transmitter may support simultaneous connections to HDCP Receivers through one or more of its HDCP-protected interface ports. The HDCP Transmitter state diagram is implemented independently on each HDCP-protected interface port.

The HDCP Transmitter Link State Diagram and HDCP Transmitter Authentication Protocol State Diagram (Figure 2-12 and Figure 2-13) 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 2.10.2.

Transmitter's decision to begin authentication is dependent on events such as detection of an HDCP Receiver, availability of premium content or other implementation dependent details in the transmitter. In the event of authentication failure, an HDCP Receiver must be prepared to process subsequent authentication attempts. The HDCP Transmitter may cease to attempt authentication for transmitter-specific reasons, which include receiving a Receiver Disconnected Indication or after a certain number of authentication re-attempts by the transmitter.

The transmitter must not initiate authentication unless it determines that the receiver is HDCP-capable.

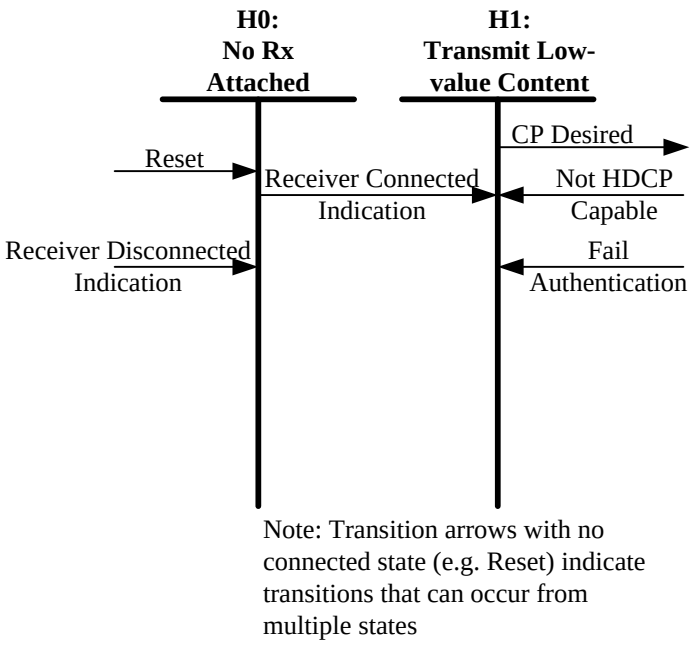


Figure 2-12. HDCP Transmitter Link State Diagram

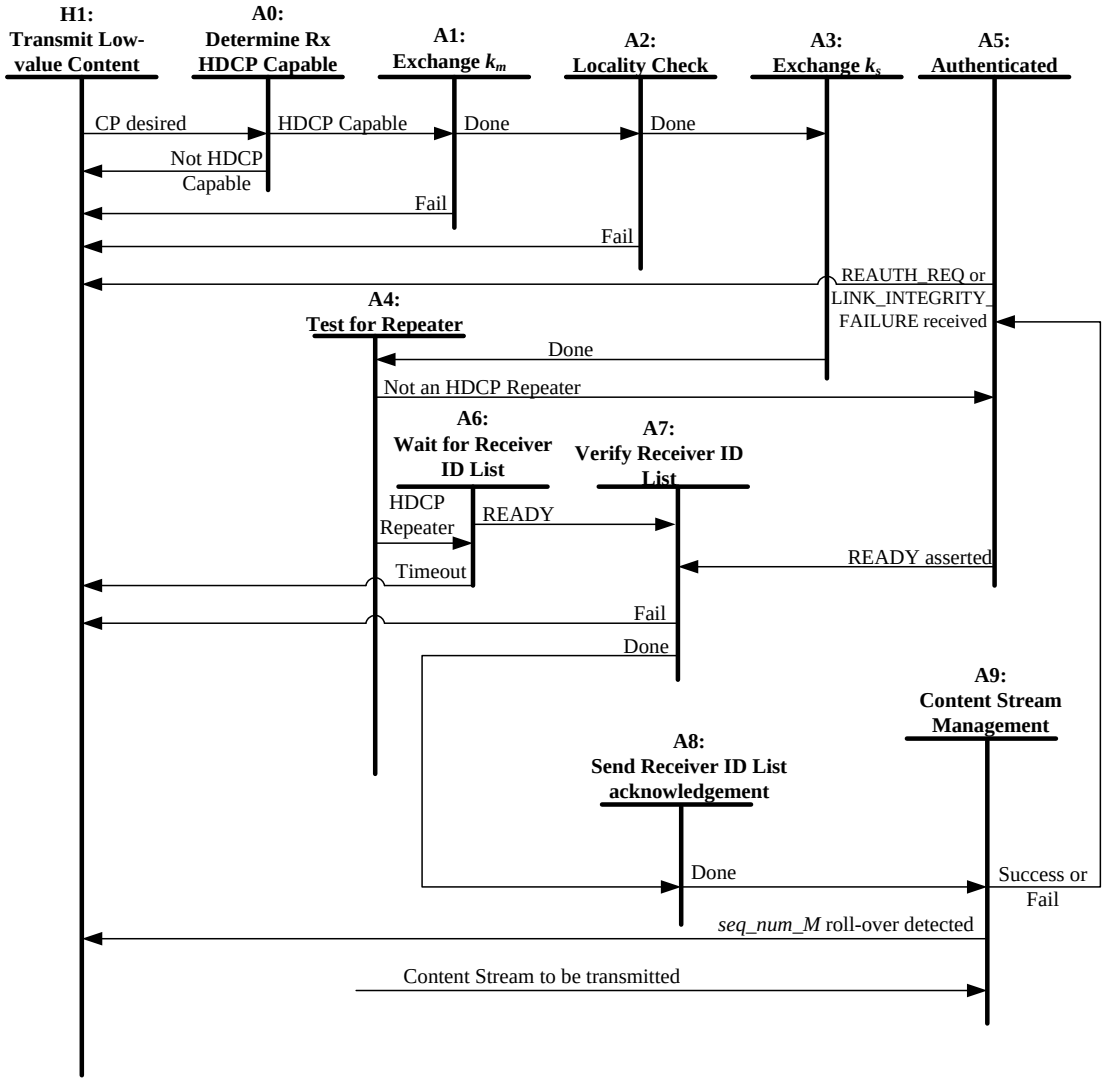


Figure 2-13. HDCP Transmitter Authentication Protocol State Diagram

Transition Any State:H0. Reset conditions at the HDCP Transmitter or disconnect of the connected HDCP capable receiver cause the HDCP Transmitter to enter the No Receiver Attached state.

Transition H0:H1. The detection of a sink device (through Receiver Connected Indication) indicates to the transmitter that a sink device is connected and ready to display the received content. When the receiver is no longer active, the transmitter is notified through Receiver Disconnected Indication.

State H1: Transmit Low-value Content. In this state the transmitter should begin sending an unencrypted signal with HDCP Encryption disabled. The transmitted signal can be a low value content or informative on-screen display. This will ensure that a valid video signal is displayed to the user before and during authentication.

Transition H1:A0. If content protection is desired by the Upstream Content Control Function, then the HDCP Transmitter should immediately attempt to determine whether the receiver is HDCP capable.

State A0: Determine Rx HDCP Capable. The transmitter determines that the receiver is HDCP capable by reading the HDCP_CAPABLE bit in the receiver's *RxCaps* register. If this bit is set to 1, it indicates that the receiver is HDCP capable. If state A0 is reached when content protection is desired by the Upstream Content Control Function, authentication must be started immediately by the transmitter if the receiver is HDCP capable. A valid video screen is displayed to the user with encryption disabled during this time.

Transition A0:H1. If the receiver is not HDCP capable, the transmitter continues to transmit low value content or informative on-screen display.

Transition A0:A1. If the receiver is HDCP capable, the transmitter initiates the authentication protocol.

State A1: Exchange k_m. In this state, the HDCP Transmitter initiates authentication by writing AKE_Init message to the HDCP Receiver. It reads AKE_Send_Cert from the receiver within the time period specified in Section 2.2.

If the HDCP Transmitter does not have k_m stored corresponding to the *Receiver ID*, it generates $E_{kpub}(km)$ and sends $E_{kpub}(km)$ as part of the AKE_No_Stored_km message to the receiver after verification of signature on $cert_{rx}$. It performs integrity check on the SRM and checks to see whether the *Receiver ID* of the connected HDCP Device is in the revocation list. It computes H, receives AKE_Send_H_prime message from the receiver containing *H'* within the time period specified in Section 2.2 and compares *H'* against H.

If the HDCP Transmitter has k_m stored corresponding to the *Receiver ID*, it writes AKE_Stored_km message containing $E_{kh}(k_m)$ and *m* to the receiver, performs integrity check on the SRM and checks to see whether the *Receiver ID* of the connected HDCP Device is in the revocation list. It computes H, reads AKE_Send_H_prime message from the receiver containing *H'* within the time period specified in Section 2.2 and compares *H'* against H.

If the HDCP Transmitter does not have a k_m stored corresponding to the *Receiver ID*, it implements pairing with the HDCP Receiver as explained in Section 2.2.1.

Transition A1:H1. This transition occurs on failure of signature verification on $cert_{rx}$, failure of SRM integrity check, if *Receiver ID* of the connected HDCP Device is in the revocation list or if there is a mismatch between H and *H'*. This transition also occurs if AKE_Send_H_prime message is not received within the time period specified in Section 2.2.

Transition A1:A2. The HDCP Transmitter implements locality check after successful completion of AKE and pairing.

State A2: Locality Check. In this state, the HDCP Transmitter implements the locality check as explained in Section 2.3 with the HDCP Receiver.

Transition A2:H1. This transition occurs on one or more consecutive locality check failures. Locality check fails when the watchdog timer at the HDCP Transmitter expires or on a mismatch between L and *L'*.

Transition A2:A3. The HDCP Transmitter implements SKE after successful completion of locality check.

State A3: Exchange k_s. The HDCP Transmitter sends encrypted Session Key, $E_{dkey}(k_s)$, and r_{iv} to the HDCP Receiver as part of the SKE_Send_Eks message. It may enable HDCP Encryption 200 ms after sending encrypted Session Key. HDCP Encryption must be enabled only after successful completion of AKE, locality check and SKE stages.

Transition A3:A4. This transition occurs after completion of SKE.

State A4: Test for Repeater. The HDCP Transmitter evaluates the REPEATER value that was received in State A1.

Transition A4:A5. REPEATER bit is not set (the HDCP Receiver is not an HDCP Repeater).

State A5: Authenticated. At this time, and at no prior time, the HDCP Transmitter has completed the authentication protocol.

A periodic Link Integrity Checkis performed to maintain cipher synchronization between the HDCP Transmitter and the HDCP Receiver.

Transition A4:A6. REPEATER bit is set (the HDCP Receiver is an HDCP Repeater).

State A6: Wait for Receiver ID List. The HDCP Transmitter sets up a three-second watchdog timer after sending SKE_Send_Eks.

Transition A6:H1. The watchdog timer expires before READY has been asserted by the repeater.

Transition A6:A7. READY bit is asserted.

State A7: Verify Receiver ID List. If a transition in to this state occurs from State A6, the watchdog timer is cleared. The transmitter reads the RepeaterAuth_Send_ReceiverID_List message. If both MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED bits are not set, the transmitter computes *V* and compares the most significant 128-bits of *V* and *V'*. The *Receiver ID*s from the Receiver ID list are compared against the current revocation list.

Transition A7:H1. This transition is made if a mismatch occurs between the most significant 128-bits of *V* and *V'*. This transition is also made if any of the *Receiver ID*s in the Receiver ID list are found in the current revocation list or if the HDCP Transmitter detects a roll-over of *seq_num_V*. A MAX_CASCADE_EXCEEDED or MAX_DEVS_EXCEEDED error also causes this transition.

Transition A7:A8. This transition occurs on successful verification of the most significant 128-bits of *V* and *V'*, none of the reported *Receiver ID*s are in the current revocation list, the HDCP Transmitter does not detect a roll-over of *seq_num_V* and the downstream topology does not exceed specified maximums.

State A8: Send Receiver ID list acknowledgement. , The HDCP Transmitter sends the least significant 128-bits of *V* to the HDCP Repeater as part of the RepeaterAuth_Send_Ack message.

The RepeaterAuth_Send_Ack message must be received by the HDCP Repeater within two seconds from the time the READY status was set and the CP_IRQ interrupt was asserted by the HDCP Repeater..

Transition A8:A9. This transition occurs after the RepeaterAuth_Send_Ack message has been written to the repeater.

Transition A5:H1. This transition occurs if a CP_IRQ interrupt was received with the REAUTH_REQ or LINK_INTEGRITY_FAILURE status bit set.

Transition A5:A7. This transition occurs whenever a CP_IRQ interrupt is received from the connected HDCP Repeater with READY bit set.

State A9: Content Stream Management. This stage is implemented if Content Stream is to be transmitted. The HDCP Transmitter sends the RepeaterAuth_Stream_Manage message specifying Type values assigned to Content Streams, to the attached HDCP Repeater at least 110ms before the transmission of the corresponding Content Streams after HDCP Encryption. It must receive the RepeaterAuth_Stream_Ready message from the HDCP Repeater within the time period specified in Section 2.5.2, and verifies M'. This step fails if the RepeaterAuth_Stream_Ready message is not available to read within the time period specified in Section 2.5.2, or if *M* is not equal to *M'*.

This stage may be implemented in parallel with the upstream propagation of topology information (State A4, State A6, State A7 and State A8) and with the flow of encrypted content and Link Integrity Check (State A5). This state may be implemented asynchronously from the rest of the state diagram. A transition in to this state may occur from State A4, State A5, State A6, State A7 or State A8 if Content Stream is to be transmitted. Also, the transition from State A9 must return to the appropriate state to allow for undisrupted operation.

Transition A9:A5. This transition occurs on success or failure of the Content Stream management stage.

Transition A9:H1. This transition occurs if *seq_num_M* rolls over.

Note: Since Link Integrity Check (State A5) may be implemented in parallel with the upstream propagation of topology information (State A4, State A6, State A7 and State A8) and Content Stream management (State A9) stages, the link integrity checkprocess (i.e. State A5) may be implemented asynchronously from the rest of the state diagram. 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.

The HDCP Transmitter may support simultaneous connections to HDCP Receivers through one or more of its HDCP-protected interface ports. It may share the same Session Key and r_{iv} across all its HDCP-protected interface ports, as explained in Section 3.5. However, the HDCP Transmitter must ensure that each connected HDCP Receiver receives distinct k_m and r_{tx} values.

2.9 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 2-14. For HDCP Repeaters, the upstream (HDCP Receiver) side is covered in Section 2.10.3.

The HDCP Receiver must be ready to re-authenticate with the HDCP Transmitter at any point in time. In particular, the only indication to the HDCP Receiver of a re-authentication attempt by the HDCP Transmitter is the reception of the AKE_Init message from the HDCP Transmitter.

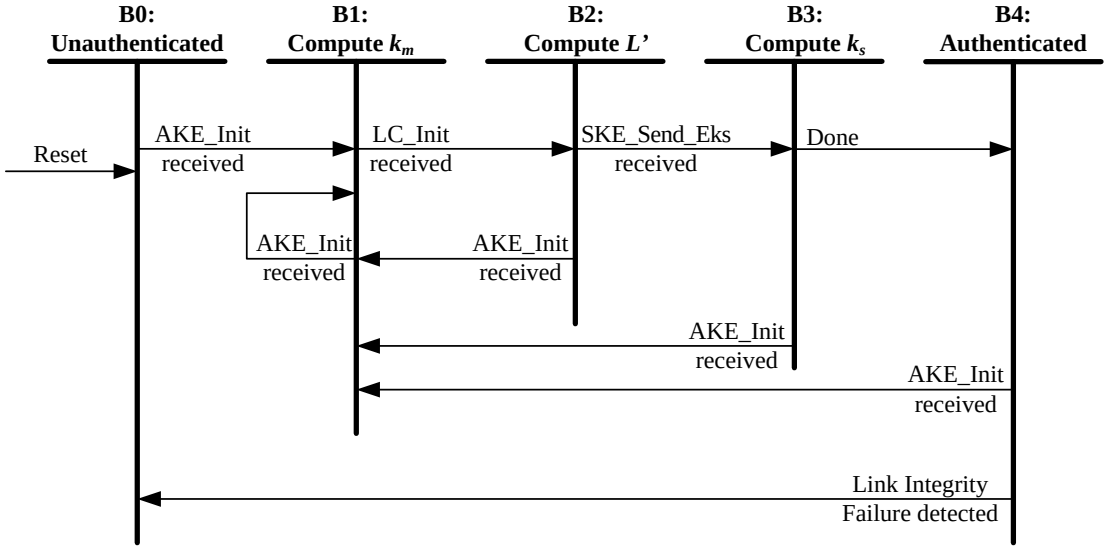


Figure 2-14. HDCP Receiver Authentication Protocol State Diagram

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 AKE_Init from the HDCP Transmitter to trigger the authentication protocol.

Transition B0:B1. AKE_Init message is received from the HDCP Transmitter.

State B1: Compute k_m. In this state, the HDCP Receiver makes the AKE_Send_Cert message available for reading by the transmitter in response to AKE_Init. If AKE_No_Stored_km is received, the receiver decrypts k_m with $kpriv_{rx,}$ calculates *H'*. It makes AKE_Send_H_prime message available for reading immediately after computation of *H'* to ensure that the message is received by the transmitter within the time period specified in Section 2.2.

If AKE_Stored_km is received, the HDCP Receiver decrypts $E_{kh}(k_m)$ to derive k_m and calculates *H'*. It makes AKE_Send_H_prime message available for reading immediately after computation of *H'* to ensure that the message is received by the transmitter within the time period specified in Section 2.2.

If AKE_No_Stored_km is received, this is an indication to the HDCP Receiver that the HDCP Transmitter does not contain a k_m stored corresponding to its *Receiver ID*. It implements pairing with the HDCP Transmitter as explained in Section 2.2.1.

Transition B1: B1. Should the HDCP Transmitter write an AKE_Init while the HDCP Receiver is in State B1, the HDCP Receiver abandons intermediate results and restarts computation of k_m.

Transition B1: B2. The transition occurs when r_n is received as part of LC_Init message from the transmitter.

State B2: Compute *L'*. The HDCP Receiver computes *L'* required during locality check and makes the LC_Send_L_prime message available for reading by the transmitter.

Transition B2: B1. Should the HDCP Transmitter write an AKE_Init while the HDCP Receiver is in State B2, the HDCP Receiver abandons intermediate results and restarts computation of k_m.

Transition B2: B3. The transition occurs when SKE_Send_Eks message is received from the transmitter.

State B3: Compute k_s. The HDCP Receiver decrypts $E_{dkey}(k_s)$ to derive k_s.

Transition B3: B1. Should the HDCP Transmitter write an AKE_Init while the HDCP Receiver is in State B3, the HDCP Receiver abandons intermediate results and restarts computation of k_m.

Transition B3: B4. Successful computation of k_s transitions the receiver into the authenticated state.

State B4: Authenticated. The HDCP Receiver has completed the authentication protocol. Periodically, it performs a link integrity check.

Transition B4: B1. Should the HDCP Transmitter write an AKE_Init while the HDCP Receiver is in State B4, the HDCP Receiver abandons intermediate results and restarts computation of k_m.

Transition B4: B0. This transition occurs when a link integrity failure is detected by the HDCP Receiver as explained in Section 2.6. The HDCP Receiver asserts the LINK_INTEGRITY_FAILURE bit in the *RxStatus* register and generates a CP_IRQ interrupt.

2.10 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. The state diagram for each downstream connection (Figure 2-15 and Figure 2-16) is substantially the same as that for the host HDCP Transmitter (Section 2.8), with the exception that the HDCP Repeater is not required to check for downstream Receiver IDs in a revocation list.

When the upstream HDCP-protected interface port of the HDCP Repeater is in an unauthenticated state, it signals the detection of an active downstream HDCP Receiver to the upstream HDCP Transmitter in one of the following ways. The HDCP Repeater must generate the CONNECTION_STATUS_NOTIFY message to indicate plug of an active HDCP Receiver when the most upstream HDCP Transmitter is capable of operating in the MST mode (MST-capable). It must pulse IRQ_HPD when the most upstream HDCP Transmitter is capable of operating only in the SST mode (SST-capable only) and has enabled IRQ_HPD for upstream notification. Hot plug, CONNECTION_STATUS_NOTIFY and IRQ_HPD are collectively referred to as Receiver Connected Indication in this specification.

Hot Unplug is referred to as Receiver Disconnected Indication in this specification.

Whenever authentication is initiated by the upstream HDCP Transmitter by sending AKE_Init, the HDCP Repeater immediately initiates authentication on all its downstream HDCP-protected interface ports if its downstream ports are in an unauthenticated state.

The HDCP Repeater may cache the latest Receiver ID list and topology information received on its downstream ports. Whenever authentication is attempted by the upstream transmitter by writing the AKE_Init message, the HDCP Repeater may propagate the cached Receiver ID list upstream without initiating a re-authentication on all its downstream ports.

The HDCP Repeater must generate unique k_m values for HDCP Devices connected to each of its downstream HDCP-protected Interface Ports.

The HDCP Repeater may transmit the same session key, $k_{s,}$ to all its authenticated and active downstream HDCP-protected Interface Ports before beginning the transmission of HDCP Content to any of its downstream ports. After beginning the transmission of HDCP Content by the HDCP Repeater to any of its downstream ports, subsequent connection of a new HDCP Receiver to its downstream port must result in (a) a unique session key, k_s, exchanged with that HDCP Receiver or (b) a new authentication attempt with all its downstream HDCP-protected Interface ports and subsequent exchange of the same session key, $k_{s,}$ to all its authenticated and active downstream HDCP-protected Interface Ports.

If an HDCP Repeater has no active downstream HDCP Devices, it must authenticate as an HDCP Receiver with REPEATER bit set to zero if it wishes to receive HDCP Content, but must not pass HDCP Content to downstream devices.

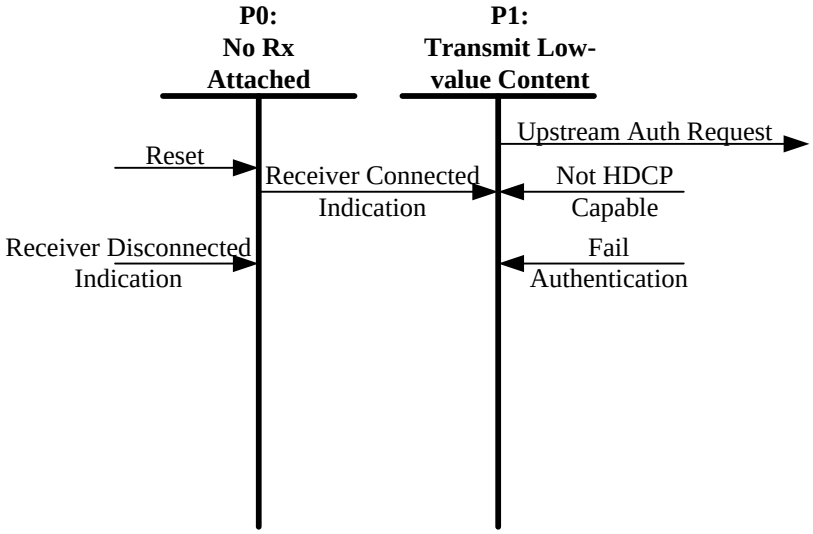
When the upstream HDCP-protected Interface Port of the HDCP Repeater transitions in to an unauthenticated state from an authenticated state (See Transition C5:C0 and Transition C6:C0 in Section 2.10.3), the HDCP Repeater must set the REAUTH_REQ status bit in the *RxStatus* register and assert CP_IRQ interrupt. If the upstream HDCP Transmitter receives a CP_IRQ interrupt with REAUTH_REQ status bit set, it may initiate re-authentication with the HDCP Repeater with the transmission of a new AKE_Init message.

2.10.1 Propagation of Topology Errors

MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED: HDCP Repeaters must be capable of supporting DEVICE_COUNT values of up to 31 and DEPTH values of up to 4. If the computed DEVICE_COUNT for an HDCP Repeater exceeds 31, the error is referred to as MAX_DEVS_EXCEEDED error. The repeater sets MAX_DEVS_EXCEEDED bit to one in the RepeaterAuth_Send_ReceiverID_List message. If the computed DEPTH for an HDCP Repeater exceeds four, the error is referred to as MAX_CASCADE_EXCEEDED error. The repeater sets MAX_CASCADE_EXCEEDED bit to one in the RepeaterAuth_Send_ReceiverID_List message. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED error from a downstream HDCP Repeater, it must propagate the error to the upstream HDCP Transmitter and must not transmit *V'* and Receiver ID list.

2.10.2 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.



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

Figure 2-15. HDCP Repeater Downstream Link State Diagram

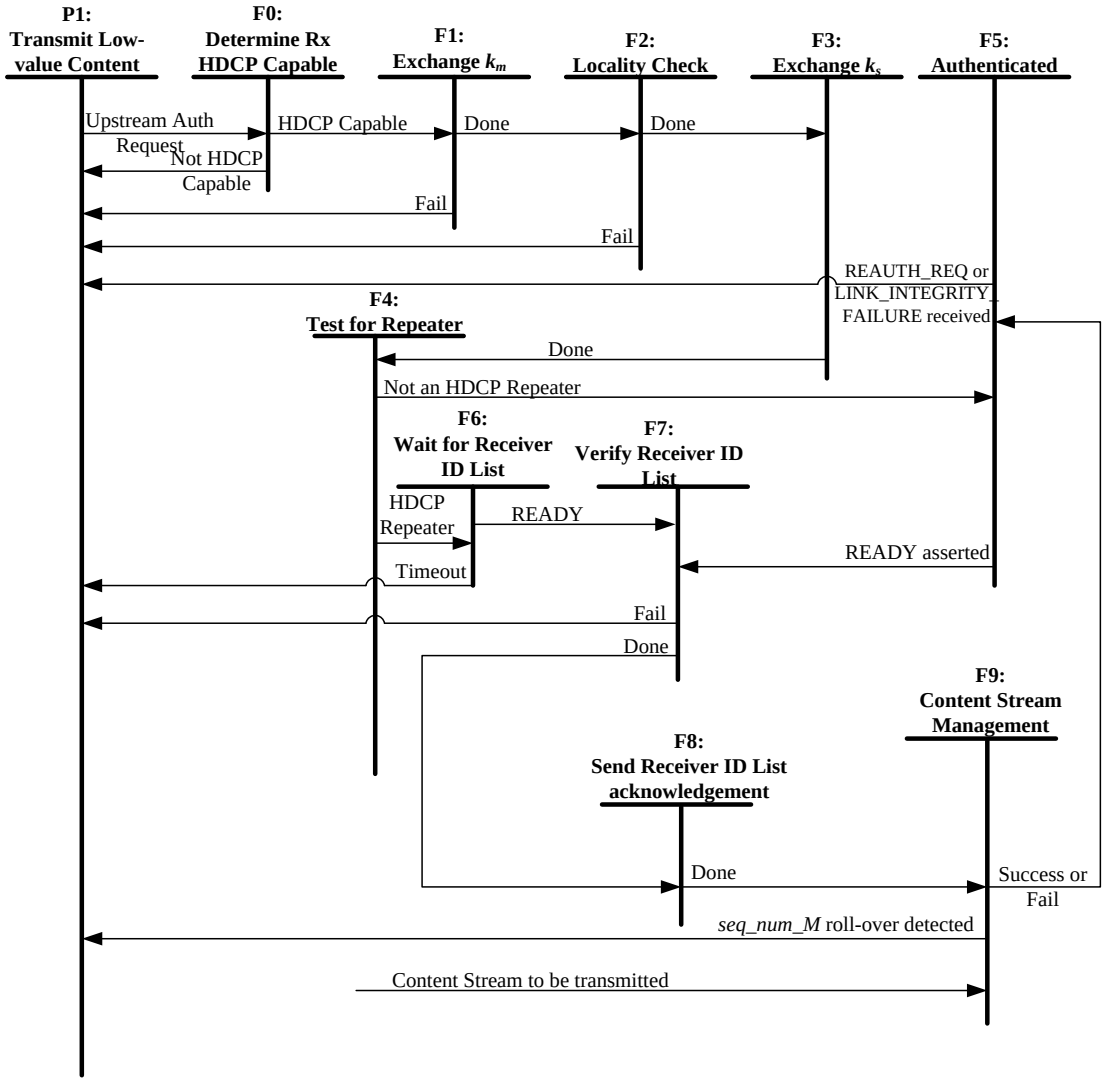


Figure 2-16. HDCP Repeater Downstream Authentication Protocol State Diagram

Transition Any State:P0. Reset conditions at the HDCP Repeater or disconnect of the connected HDCP capable receiver cause the HDCP Repeater to enter the No Receiver Attached state for this port.

Transition P0:P1. The detection of a sink device (through Receiver Connected Indication) indicates that the receiver is available and active (ready to display received content). When the receiver is no longer active, the downstream (HDCP Transmitter) side is notified through Receiver Disconnected Indication.

State P1: Transmit low-value content. 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 downstream side should immediately attempt to determine whether the receiver is HDCP capable.

State F0: Determine Rx HDCP Capable. The downstream side determines that the receiver is HDCP capable by reading the HDCP_CAPABLE bit in the receiver's *RxCaps* register. If this bit is set to 1, it indicates that the receiver is HDCP capable. If state F0 is reached upon an Upstream Authentication Request, authentication must be started immediately by the downstream side if the receiver is HDCP capable. A valid video screen is displayed to the user with encryption disabled during this time.

Note: The downstream side may initiate authentication with the attached HDCP Receiver before an Upstream Authentication Request is received.

Transition F0:P1. If the receiver is not HDCP capable, the downstream side continues to transmit low value content or informative on-screen display received from the upstream HDCP Transmitter.

Transition F0:F1. If the receiver is HDCP capable, the downstream side initiates the authentication protocol.

State F1: Exchange k_m. In this state, the downstream side initiates authentication by writing AKE_Init message to the HDCP Receiver. It reads AKE_Send_Cert from the receiver within the time period specified in Section 2.2.

If the downstream side does not have k_m stored corresponding to the *Receiver ID*, it generates $E_{kpub}(km)$ and sends $E_{kpub}(km)$ as part of the AKE_No_Stored_km message to the receiver after verification of signature on $cert_{rx}$. It computes H, receives AKE_Send_H_prime message from the receiver containing *H'* within the time period specified in Section 2.2 and compares *H'* against H.

If the downstream side has k_m stored corresponding to the *Receiver ID*, it sends AKE_Stored_km message containing $E_{kh}(k_m)$ and *m* to the receiver. It computes H, receives AKE_Send_H_prime message from the receiver containing *H'* within the time period specified in Section 2.2 and compares *H'* against H.

If the downstream side does not have a k_m stored corresponding to the *Receiver ID*, it implements pairing with the HDCP Receiver as explained in Section 2.2.1.

Transition F1:P1. This transition occurs on failure of signature verification on $cert_{rx}$ or if there is a mismatch between H and *H'*. This transition also occurs if AKE_Send_H_prime message is not received within the time period specified in Section 2.2.

Transition F1:F2. The downstream side implements locality check after successful completion of AKE and pairing.

State F2: Locality Check. In this state, the downstream side implements the locality check as explained in Section 2.3 with the HDCP Receiver..

Transition F2:P1. This transition occurs on one or more consecutive locality check failures. Locality check fails when the watchdog timer at the downstream side expires or on a mismatch between L and L'.

Transition F2:F3. The downstream side implements SKE after successful completion of locality check.

State F3: Exchange k_s. The downstream side sends encrypted Session Key, $E_{dkey}(k_s)$, and r_{iv} to the HDCP Receiver as part of the SKE_Send_Eks message. It may enable HDCP Encryption 200 ms after sending encrypted Session Key. HDCP Encryption must be enabled only after successful completion of AKE, locality check and SKE stages.

Transition F3:F4. This transition occurs after completion of SKE.

State F4: Test for Repeater. The downstream side evaluates the REPEATER value that was received in State F1.

Transition F4:F5. REPEATER bit is not set (the HDCP Receiver is not an HDCP Repeater).

State F5: Authenticated. At this time, and at no prior time, the downstream side has completed the authentication protocol.

A periodic Link Integrity Checkis performed to maintain cipher synchronization between the downstream side and the HDCP Receiver.

Transition F4:F6. REPEATER bit is set (the HDCP Receiver is an HDCP Repeater).

State F6: Wait for Receiver ID List. The downstream side sets up a three-second watchdog timer after sending SKE_Send_Eks.

Transition F6:P1. The watchdog timer expires before READY has been asserted by the repeater.

Transition F6:F7. READY bit is asserted.

State F7: Verify Receiver ID List. If a transition in to this state occurs from State F6, the watchdog timer is cleared. The downstream side reads the RepeaterAuth_Send_ReceiverID_List message. If both MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED bits are not set, the downstream side computes *V* and compares the most significant 128-bits of *V* and *V'*. The *Receiver ID*s from this port are added to the Receiver ID list for this HDCP Repeater. The upstream HDCP Transmitter must be informed if topology maximums are exceeded.

Transition F7:P1. This transition is made if a mismatch occurs between the most significant 128-bits of *V* and *V'*. This transition is also made if the downstream side detects a roll-over of *seq_num_V*. A MAX_CASCADE_EXCEEDED or MAX_DEVS_EXCEEDED error also causes this transition.

Transition F7:F8. This transition occurs on successful verification of the most significant 128-bits of *V* and *V'*, the downstream side does not detect a roll-over of *seq_num_V* and the downstream topology does not exceed specified maximums.

State F8: Send Receiver ID list acknowledgement. , The downstream side sends the least significant 128-bits of *V* to the attached HDCP Repeater as part of the RepeaterAuth_Send_Ack message.

The RepeaterAuth_Send_Ack message must be received by the attached HDCP Repeater within two seconds from the time the READY status was set and the CP_IRQ interrupt was asserted by the attached HDCP Repeater..

Transition F8:F9. This transition occurs after the RepeaterAuth_Send_Ack message has been written to the repeater.

Transition F5:P1. This transition occurs if a CP_IRQ interrupt was received with the REAUTH_REQ or LINK_INTEGRITY_FAILURE status bit set.

Transition F5:F7. This transition occurs whenever a CP_IRQ interrupt is received from the connected HDCP Repeater with READY bit set.

State F9: Content Stream Management. This stage is implemented if Content Stream is to be transmitted. The downstream side propagates the Content Stream management information, received from the upstream transmitter, using the RepeaterAuth_Stream_Manage message to the attached HDCP Repeater at least 110ms before the transmission of the corresponding Content Streams after HDCP Encryption. If the upstream transmitter is HDCP 2.0-compliant or HDCP 1.x-compliant, the downstream side will not receive the RepeaterAuth_Stream_Manage message from the upstream transmitter and assigns a Type value of 0x00 to all Content Streams received from the upstream transmitter and propagates the Content Stream management information using the RepeaterAuth_Stream_Manage message.

The downstream side must receive the RepeaterAuth_Stream_Ready message from the HDCP Repeater within the time period specified in Section 2.5.2, and verifies *M'*. This step fails if the RepeaterAuth_Stream_Ready message is not available to read within the time period specified in Section 2.5.2, or if *M* is not equal to *M'*.

This stage may be implemented in parallel with the upstream propagation of topology information (State F4, State F6, State F7 and State F8) and with the flow of encrypted content and Link Integrity Check (State F5). This state may be implemented asynchronously from the rest of the state diagram. A transition in to this state may occur from State F4, State F5, State F6, State F7 or State F8 if Content Stream is to be transmitted and the Content Stream management information is received from the upstream HDCP Transmitter. Also, the transition from State F9 must return to the appropriate state to allow for undisrupted operation.

Transition F9:F5. This transition occurs on success or failure of the Content Stream management stage.

Transition F9:P1. This transition occurs if *seq_num_M* rolls over.

Note: Since Link Integrity Check may be implemented in parallel with the upstream propagation of topology information (State F4, State F6, State F7 and State F8) and Content Stream management (State F9) stages, the link integrity checkprocess (i.e. State F5) may be implemented asynchronously from the rest of the state diagram. 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.

2.10.3 HDCP Repeater Upstream State Diagram

The HDCP Repeater upstream state diagram, illustrated in Figure 2-17, makes reference to states of the HDCP Repeater downstream state diagram. 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.

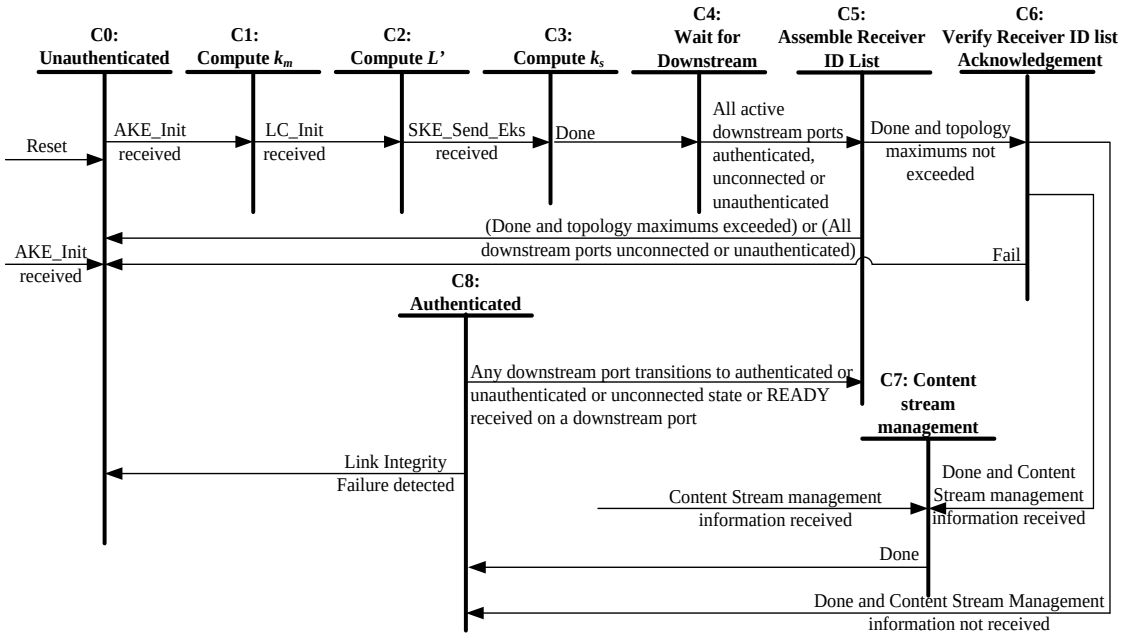


Figure 2-17. HDCP Repeater Upstream Authentication Protocol State Diagram

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 AKE_Init is received from the connected HDCP Transmitter, with a transition through the unauthenticated state.

State C0: Unauthenticated. The device is idle, awaiting the reception of AKE_Init from the HDCP Transmitter to trigger the authentication protocol.

If a transition in to this state occurred from State C5, when State C5 is implemented in parallel with State C8, or from State C6, the HDCP Repeater must set the REAUTH_REQ status bit in the *RxStatus* register and assert CP_IRQ interrupt.

If a transition in to this state occurred from State C8, the HDCP Repeater must set the LINK_INTEGRITY_FAILURE bit in the *RxStatus* register and assert CP_IRQ interrupt.

Transition C0:C1. AKE_Init message is received from the HDCP Transmitter.

State C1: Compute k_m. In this state, the upstream (HDCP Receiver) side makes available the AKE_Send_Cert message for the transmitter to read in response to AKE_Init. If AKE_No_Stored_km is received, it decrypts k_m with $kpriv_{rx,}$ calculates *H'*. It makes available the AKE_Send_H_prime message within the time period specified in Section 2.2.

If AKE_Stored_km is received, the upstream side decrypts $E_{kh}(k_m)$ to derive k_m and calculates *H'*. It makes available the AKE_Send_H_prime message within the time period specified in Section 2.2.

If AKE_No_Stored_km is received, this is an indication to the upstream side that the HDCP Transmitter does not contain a k_m stored corresponding to its *Receiver ID*. It implements pairing with the HDCP Transmitter as explained in Section 2.2.1.

Transition C1:C2. The transition occurs when r_n is received as part of LC_Init message from the transmitter.

State C2: Compute *L'*. The upstream side computes *L'* required during locality check and sends LC_Send_L_prime message.

Transition C2: C3. The transition occurs when SKE_Send_Eks message is received from the transmitter.

State C3: Compute k_s. The upstream side decrypts $E_{dkey}(k_s)$ to derive k_s.

Transition C3: C4. Successful computation of k_s causes this transition.

State C4: Wait for Downstream. The upstream state machine waits for all downstream HDCP-protected Interface Ports of the HDCP Repeater to enter the unconnected (State P0), unauthenticated (State P1), or the authenticated state (State F5).

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

State C5: Assemble Receiver ID List. The upstream side assembles the list of all connected 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 P1, the unauthenticated state, does not add to the list. A downstream HDCP-protected Interface Port that arrives in State F5 that has an HDCP Receiver that is not an HDCP Repeater connected, adds the *Receiver ID* of the connected HDCP Receiver to the list. Downstream HDCP-protected Interface Ports that arrive in State F5 that have an HDCP Repeater connected will cause the Receiver ID list read from the connected HDCP Repeater, plus the *Receiver ID* of the connected HDCP Repeater itself, to be added to the list.

Note: The upstream side may add the Receiver ID list read from the HDCP Repeater connected to the downstream HDCP-protected Interface port, plus the *Receiver ID* of the connected HDCP Repeater itself to the list after the downstream port has transitioned in to State F8.

When the Receiver ID list for all downstream HDCP Receivers has been assembled, the upstream side computes DEPTH, DEVICE_COUNT and the upstream *V'* and asserts its READY status indicator in the *RxStatus* register and asserts the CP_IRQ interrupt. In the case of a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED error, it asserts the corresponding bits to the upstream transmitter. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or MAX_CASCADE_EXCEEDED error from a downstream HDCP Repeater, it is required to inform the upstream HDCP Transmitter.

If any downstream port connected to an HDCP Repeater detects the HDCP2_LEGACY_DEVICE_DOWNSTREAM or HDCP1_DEVICE_DOWNSTREAM bits read from the repeater to be set to one, the upstream side sets the corresponding bits to one in the *RxInfo* register which is read by the upstream HDCP Transmitter as part of the RepeaterAuth_Send_ReceiverID_List message.

Transition C5:C0. This transition occurs if RepeaterAuth_Send_ReceiverID_List message has been read by the upstream HDCP Transmitter and topology maximums are exceeded i.e. on a MAX_DEVS_EXCEEDED or MAX_CASCADE_EXCEEDED error. This transition may also occur if all downstream HDCP-protected Interface Ports have reached the state of unconnected or unauthenticated .

Transition C5:C6. RepeaterAuth_Send_ReceiverID_List message has been read by the upstream HDCP Transmitter and topology maximums are not exceeded.

State C6. Verify Receiver ID list acknowledgement. In this state, the upstream side receives the RepeaterAuth_Send_Ack message from the upstream transmitter and compares the least significant 128-bits of *V* and *V'*. A match between the least significant 128-bits of *V* and *V'* indicates successful upstream transmission of topology information. The last byte of the RepeaterAuth_Send_Ack message must be written to the upstream side by the transmitter within two seconds from the time the READY status was set and the CP_IRQ interrupt was asserted by the upstream side..

Transition C6:C0. This transition occurs if the RepeaterAuth_Send_Ack message is not received by the upstream side within two seconds or on a mismatch between the least significant 128-bits of *V* and *V'*. If this transition occurs, the upstream side must set the REAUTH_REQ status bit in the *RxStatus* register and assert CP_IRQ interrupt.

Transition C6:C7. This transition occurs if the RepeaterAuth_Send_Ack message is received by the upstream side within two seconds, on a successful match between the least significant 128-bits of *V* and *V'* and if Content Stream management information is received from the upstream transmitter.

Transition C6:C8. This transition occurs if the RepeaterAuth_Send_Ack message is received by the upstream side within two seconds, on a successful match between the least significant 128-bits of *V* and *V'* and if Content Stream management information is not received from the upstream transmitter.

State C7: Content Stream Management. On receiving the RepeaterAuth_Stream_Manage message, the upstream side computes *M'* and makes it available for the the upstream Transmitter to read as part of the RepeaterAuth_Stream_Ready message.

This stage may be implemented in parallel with the upstream propagation of topology information (State C4, State C5 and State C6) and with the flow of encrypted content and link integrity check (State C8). This state may be implemented asynchronously from the rest of the state diagram. A transition in to this state may occur from State C4, State C5, State C6 or State C8 if Content Stream management information is received from the upstream transmitter. Also, the transition from State C7 may return to the appropriate state to allow for undisrupted operation.

The upstream side must be prepared to implement this stage in parallel with the upstream propagation of topology information and with the flow of encrypted content and link integrity checkif these stages are implemented in parallel by the upstream transmitter.

Transition C7:C8. This transition occurs after RepeaterAuth_Stream_Ready message has been read by the upstream transmitter.

State C8: Authenticated. The upstream side has completed the authentication protocol. Periodically, it performs a link integrity check.

Transition C8:C0. This transition occurs when a link integrity failure is detected by the upstream side as explained in Section 2.6.

Transition C8:C5. This transition occurs on detection of any changes to the topology.

This transition occurs when a downstream port that was previously in the unauthenticated (State P1) or unconnected (State P0) state transitions in to the authenticated (State F5) state. For example, the transition may occur when a new HDCP Receiver is connected to a downstream port, that previously had no receivers connected, and the downstream port completes the authentication protocol with the HDCP Receiver.

This transition may also occur when a downstream port that was previously in an authenticated state transitions in to an unauthenticated or unconnected state. For example, the transition may occur when an active, authenticated HDCP Receiver attached to the downstream port is disconnected. Note that the upstream side need not transition from State C8 to State C5 when a previously authenticated downstream port transitions in to an unauthenticated or unconnected state.

Reception of a CP_IRQ interrupt with the READY status bit asserted on a downstream port from the connected downstream HDCP Repeater also causes this transition.

Note: Since Link Integrity Check may be implemented in parallel with the upstream propagation of topology information (State C4, State C5 and State C6) and Content Stream management (State C7), the link integrity checkprocess (i.e. State C8) may be implemented asynchronously from the rest of the state diagram. The transition into State C8 may occur from any state for which encryption is currently enabled. Also, the transition from state C8 may return to the appropriate state to allow for undisrupted operation.

The upstream side must be prepared to implement the link integrity checkprocess in parallel with the upstream propagation of topology information and Content Stream management if these stages are implemented in parallel by the upstream transmitter.

2.11 Converters

2.11.1 HDCP 2 – HDCP 1.x Converters

HDCP 2 – HDCP 1.x converters are HDCP Repeaters with an HDCP 2 compliant interface port on the upstream (HDCP Receiver) side and one or more HDCP 1.x compliant interface ports on the downstream (HDCP Transmitter) side.

The HDCP 1.x compliant downstream side implements the state diagram explained in the corresponding HDCP 1.x specification (See Section 1.5).

The HDCP 2 compliant upstream side implements the state diagram as explained in Section 2.10.3 with these modifications.

- **State C5: Assemble Receiver ID List**. The upstream side assembles the list of all connected 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 the unconnected state or the unauthenticated state does not add to the list. A downstream HDCP-protected Interface Port that arrives in an authenticated state that has an HDCP Receiver that is not an HDCP Repeater connected, adds the *Bksv* of the connected HDCP Receiver to the Receiver ID list. Downstream HDCP-protected Interface Ports that arrive in an authenticated state that have an HDCP Repeater connected will cause the KSV list read from the connected HDCP Repeater, plus the *Bksv* of the connected HDCP Repeater itself, to be added to the list. KSVs are used in place of *Receiver ID*s and are added to the Receiver ID list in big-endian order

  When the Receiver ID list (comprising KSVs of connected downstream HDCP 1.x Receivers, where the KSVs are added to the list in big-endian order) for all downstream HDCP Receivers has been assembled, the upstream side computes DEPTH, DEVICE_COUNT and the upstream *V'* and asserts its READY status indicator in the *RxStatus* register and asserts the CP_IRQ interrupt. In the case of a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED error, it asserts the corresponding bits to the upstream transmitter. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or MAX_CASCADE_EXCEEDED error from a downstream HDCP Repeater, it is required to inform the upstream HDCP Transmitter.

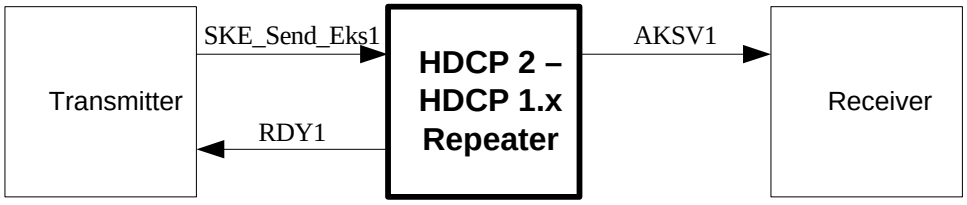


Figure 2-18. HDCP 2 – HDCP 1.x Repeater Protocol Timing with Receiver Attached

From	To	Max Delay	Conditions and Comments
SKE_Send_Eks1 Session Key received from Upstream HDCP Transmitter	AKSV1 HDCP Repeater's *Aksv* transmitted downstream	110 ms	Downstream propagation time.
AKSV1 HDCP Repeater's *Aksv* transmitted downstream	RDY1 Upstream READY asserted	220 ms	Upstream propagation time when no downstream HDCP Repeaters are attached (no downstream KSV lists to process).

Table 2-3. HDCP 2 – HDCP 1.x Repeater Protocol Timing with Receiver Attached

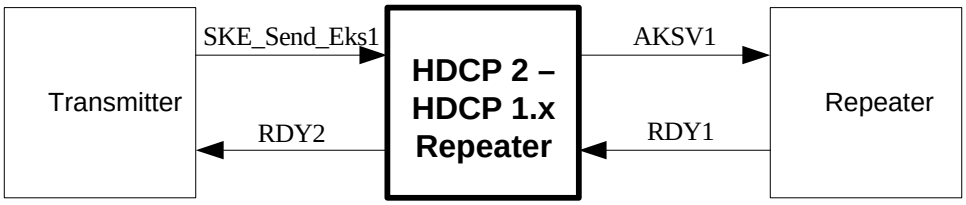


Figure 2-19. HDCP 2 – HDCP 1.x Repeater Protocol Timing with Repeater Attached

From	To	Max Delay	Conditions and Comments
SKE_Send_Eks1 Session Key received from Upstream HDCP Transmitter	AKSV1 HDCP Repeater's *Aksv* transmitted downstream	110 ms	Downstream propagation time.
RDY1 Downstream *Receiver ID*s and topology information received	RDY2 Upstream READY asserted	220 ms	Upstream propagation time when one or more HDCP 1.x-compliant Repeaters are attached. From latest downstream READY. (downstream KSV lists must be processed).

Table 2-4. HDCP 2 – HDCP 1.x Repeater Protocol Timing with Repeater Attached

2.11.2 HDCP 1.x – HDCP 2 Converters

HDCP 1.x – HDCP 2 converters are HDCP Repeaters with an HDCP 1.x compliant interface port on the upstream (HDCP Receiver) side and one or more HDCP 2 compliant interface ports on the downstream (HDCP Transmitter) side.

The HDCP 1.x compliant upstream side implements the state diagram explained in the corresponding HDCP 1.x specification (See Section 1.5).

The HDCP 2 compliant downstream side implements the state diagram as explained in Section 2.10.2 with these modifications.

- **State F7: Verify Receiver ID List**. If a transition in to this state occurs from State F6, the watchdog timer is cleared. The downstream side reads the RepeaterAuth_Send_ReceiverID_List message. If both MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED bits are not set, the downstream side computes *V* and compares the most significant 128-bits of *V* and *V'*. The *Receiver ID*s from this port are used in place of KSVs and are added to the KSV list for this HDCP Repeater. KSV list is constructed by appending *Receiver ID*s in little-endian order. The upstream HDCP Transmitter must be informed if topology maximums are exceeded.

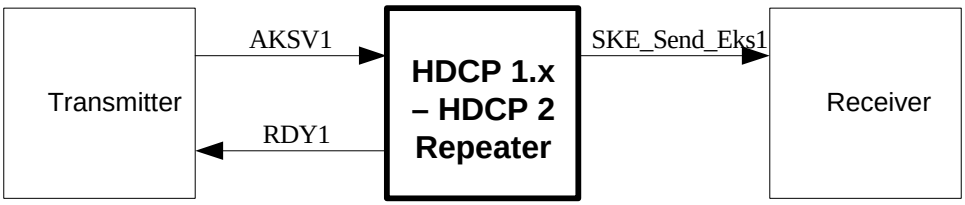


Figure 2-20. HDCP 1.x – HDCP 2 Repeater Protocol Timing with Receiver Attached

From	To	Max Delay	Conditions and Comments
AKSV1 Upstream HDCP Transmitter *Aksv* received	SKE_Send_Eks1 k_s generated by HDCP Repeater transmitted downstream	410 ms	Downstream propagation time.
SKE_Send_Eks1 k_s generated by HDCP Repeater transmitted downstream	RDY1 Upstream READY asserted	520 ms	Upstream propagation time when no downstream HDCP Repeaters are attached (no downstream Receiver ID lists to process).

Table 2-5. HDCP 1.x – HDCP 2 Repeater Protocol Timing with Repeater Attached

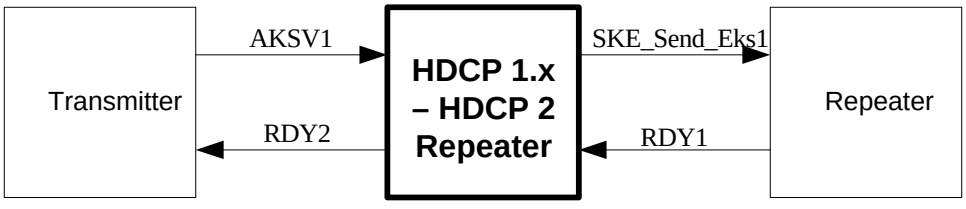


Figure 2-21. HDCP 1.x – HDCP 2 Repeater Protocol Timing with Repeater Attached

From	To	Max Delay	Conditions and Comments
AKSV1 Upstream HDCP Transmitter *Aksv* received	SKE_Send_Eks1 k_s generated by HDCP Repeater transmitted downstream	410 ms	Downstream propagation time.
RDY1 READY asserted by downstream repeater	RDY2 Upstream READY asserted	520 ms	Upstream propagation time when one or more HDCP Repeaters are attached. From latest downstream READY (downstream Receiver ID lists must be processed).

Table 2-6. HDCP 1.x – HDCP 2 Repeater Protocol Timing with Repeater Attached

2.12 Session Key Validity

When HDCP Encryption is disabled, the transmitter and receiver ceases to perform HDCP Encryption and stops incrementing the *inputCtr*.

If HDCP Encryption was disabled, from its enabled state, due to the detection of Receiver Connected Indication, Receiver Disconnected Indication or authentication failures, the HDCP Transmitter expires the Session Key. The HDCP Transmitter initiates re-authentication with the transmission of a new AKE_Init message. In all other cases, where HDCP Encryption was disabled, from its enabled state, while the link was still active and authenticated (for e.g., HDCP Encryption may be briefly disabled during transmission of low value content), the HDCP Transmitter need not expire the Session Key . The HDCP Transmitter may maintain the encryption parameters used during the HDCP Session i.e. *inputCtr* value after the last HDCP Encryption operation (after which HDCP Encryption was disabled), k_s and r_{iv}. When encryption is re-enabled, HDCP Encryption may be applied seamlessly, without requiring re-authentication, by using the same stored encryption parameters.

If HDCP Encryption was disabled, from its enabled state, the HDCP Receiver must maintain the *inputCtr* value after the last HDCP Encryption operation (after which HDCP Encryption was disabled), k_s and r_{iv} used during the HDCP Session. If encryption was re-enabled, without intervening re-authentication requests from the transmitter, the HDCP Receiver must use the same stored *inputCtr*, k_s and r_{iv}.

2.13 Random Number Generation

Random number generation is required both in the HDCP Transmitter logic and in the HDCP Receiver logic. Counter mode based deterministic random bit generator using AES-128 block cipher specified in NIST SP 800-90 is the recommended random number generator. The minimum entropy requirement for random values that are not used as secret key material (i.e. r_{tx}, r_{rx} , r_{iv}, r_n) is 40 random bits out of 64-bits. This means that a reasonable level of variability or entropy is established if out of 1,000,000 random (r_{tx}, r_{rx}, r_{iv} or r_n) values collected after the first authentication attempt (i.e. after power-up cycles on the HDCP Transmitter or HDCP Receiver logic), the probability of there being any duplicates in this list of 1,000,000 random values is less than 50%.

For randomly generated secret key material (k_m, k_s) the minimum entropy requirement is 128-bits of entropy (i.e. the probability of there being any duplicates in the list of 2^64 secret values (k_m or k_s) collected after power-up and first authentication attempt on the HDCP Transmitter logic is less than 50%).

A list of possible entropy sources that may be used for generation of random values used as secret key material include

- a true Random Number Generator or analog noise source, even if a poor (biased) one
- a pseudo-random number generator (PRNG), seeded by a true RNG with the required entropy, where the state is stored in non-volatile memory after each use. The state must be kept secret. Flash memory or even disk is usable for this purpose as long as it is secure from tampering.

A list of possible entropy sources that may be used for generation of random values not used as secret key material include

- timers, network statistics, error correction information, radio/cable television signals, disk seek times, etc.
- a reliable (not manipulatable by the user) calendar and time-of-day clock. For example, some broadcast content sources may give reliable date and time information.

2.14 CP_IRQ Interrupt Processing

HDCP Transmitters must process CP_IRQ interrupts when they are received from HDCP Receivers or HDCP Repeaters. HDCP Receivers and HDCP Repeaters must generate CP_IRQ interrupts as specified in this specification.

The HDCP Transmitter uses the following steps when processing HPD interrupts.

1. If CP_IRQ is not set, process the interrupt as specified in the DisplayPort specification and exit
2. Read *RxStatus* register
3. If LINK_INTEGRITY_FAILURE, abort HDCP session
4. If REAUTH_REQ, abort HDCP Session
5. If H'_AVAILABLE, read AKE_Send_H_prime message
6. If PAIRING_AVAILABLE, read AKE_Send_Pairing_Info message
7. If READY, continue with Authentication with Repeaters stage as specified in Section 2.5.1
8. Else ignore interrupt and continue HDCP Session without aborting

2.15 HDCP Port

HDCP Transmitters and HDCP Receivers communicate HDCP register values over the AUX channel. HDCP Receivers and HDCP Repeaters must support these HDCP registers. Within the DPCD address space, addresses from 0x69000 are reserved for HDCP2. Multi-byte values are stored in big-endian format.

Offset (hex)	Name	Size in Bytes	Rd/ Wr	Function
0x69000	r_{tx}	8	Wr	Pseudo-random number written as part of AKE_Init message. This multi-byte value must be written by the HDCP Transmitter before *TxCaps* is written.
0x69008	*TxCaps*	3	Wr	This multi-byte value is written as part of AKE_Init message. Refer to Table 2-11 for definitions.

0x6900B	$cert_{rx}$	522	Rd	HDCP Receiver Public Key Certificate read by the HDCP Transmitter as part of AKE_Send_Cert message.
0x69215	r_{rx}	8	Rd	Pseudo-random value read as part of AKE_Send_Cert message.
0x6921D	*RxCaps*	3	Rd	This multi-byte value is read as part of AKE_Send_Cert message. Refer to Table 2-10 for definitions.
0x69220	$E_{kpub}\_k_m$	128	Wr	Encrypted k_m written as part of AKE_No_Stored_km message.
0x692A0	$E_{kh}\_k_m$	16	Wr	Encrypted k_m written as part of AKE_Stored_km message. This multi-byte value must be written by the HDCP Transmitter before *m* is written.
0x692B0	*m*	16	Wr	Initialization Vector used for encryption of k_m during the pairing process and written as part of the AKE_Stored_km message.
0x692C0	*H´*	32	Rd	HMAC computed during AKE and read as part of AKE_Send_H_prime message
0x692E0	$E_{kh}\_k_m$	16	Rd	Encrypted k_m read as part of AKE_Send_Pairing_Info message.
0x692F0	r_n	8	Wr	Pseudo-random number written as part of LC_Init message.
0x692F8	*L´*	32	Rd	HMAC computed during Locality Check and read as part of LC_Send_L_prime message.
0x69318	$E_{dkey}\_k_s$	16	Wr	Encrypted session key written as part of SKE_Send_Eks message. This multi-byte value must be written by the HDCP Transmitter before r_{iv} is written.This value must be written to the HDCP Receiver at least 200 ms before enabling HDCP Encryption.
0x69328	r_{iv}	8	Wr	Pseudo-random number written as part of SKE_Send_Eks message. This value must be written to the HDCP Receiver at least 200 ms before enabling HDCP Encryption.
0x69330	*RxInfo*	2	Rd	Refer to Table 2-9 for definitions.
0x69332	*seq_num_V*	3	Rd	Sequence number used for the computation of *V*/*V*' and read as part of RepeaterAuth_Send_ReceiverID_List message.
0x69335	*V´*	16	Rd	Most-significant 128-bits of HMAC computed during upstream propagation of topology information and read as part of RepeaterAuth_Send_ReceiverID_List message.
0x69345	Receiver ID List	155	Rd	List of Receiver ID read as part of RepeaterAuth_Send_ReceiverID_List message. All bytes read as 0x00 for HDCP Receivers that are not HDCP Repeaters (REPEATER = 0).
0x693E0	*V*	16	Wr	Least-significant 128-bits of HMAC computed during upstream propagation of topology information and written as part of RepeaterAuth_Send_Ack message. The last byte of this value must be written less than two seconds from the time the READY status was set and the CP_IRQ interrupt was asserted by the HDCP Repeater.
0x693F0	*seq_num_M*	3	Wr	Sequence number used for the computation of *M*/*M*' and written as part of RepeaterAuth_Stream_Manage message. This multi-byte value is written before *k* is written.

0x693F3	*k*	2	Wr	This value indicates the number of Content Streams that are being transmitted by the HDCP Transmitter to the attached HDCP Repeater during the HDCP Session and is written as part of RepeaterAuth_Stream_Manage message. This multi-byte va;ue is written before *StreamID_Type* is written.
0x693F5	*StreamID_Type*	126	Wr	Concatenation of Stream Identifiers and Type values of Content Streams that are being transmitted by the HDCP Transmitter to the attached HDCP Repeater during the HDCP Session and is written as part of RepeaterAuth_Stream_Manage message.
0x69473	*M'*	32	Rd	HMAC computed during downstream propagation of Content Stream management information and read as part of RepeaterAuth_Stream_Ready message.
0x69493	*RxStatus*	1	Rd	Refer to Table 2-8 for definitions.
0x69494	Type	1	Wr	Type value assigned to the Content Stream to be transmitted to the HDCP Receiver. This value is written by the HDCP Transmitter only to HDCP Receivers and not to HDCP Repeaters.
0x69495	Rsvd	131	Rd	All bytes read as 0x00.
0x69518	dbg	64	Rd/ Wr	Implementation-specific debug registers. Confidential values must not be exposed through these registers.

Table 2-7. HDCP Addresses in DPCD

Name	Bit Field	Rd/ Wr	Description
Rsvd	7:5	Rd	Reserved. Read as zero.
LINK_INTEGRITY_FAILURE	4	Rd	When set to one, indicates that loss of cipher synchronization was detected at the HDCP Receiver during a link integrity check. This value must be reset by the HDCP Receiver on every new authentication request by the HDCP Transmitter as indicated by a write of the AKE_Init message.
REAUTH_REQ	3	Rd	When set to one, indicates that the upstream side of the HDCP Repeater has transitioned in to an unauthenticated state from State C5, when State C5 is implemented in parallel with State C8, or from State C6 (See Section 2.10.3). The HDCP Transmitter may initiate re-authentication with the HDCP Repeater. This value must be reset by the HDCP Receiver on every new authentication request by the HDCP Transmitter as indicated by a write of the AKE_Init message.
PAIRING_AVAILABLE	2	Rd	When set to one, indicates that $E_{kh}(k_m)$ is available for reading at the HDCP Receiver. This value must be reset by the HDCP Receiver as soon as $E_{kh}(k_m)$ is read by the HDCP Transmitter.
H'_AVAILABLE	1	Rd	When set to one, indicates that *H'* is available for reading at the HDCP Receiver. This value must be reset by the HDCP Receiver as soon as *H'* is read by the HDCP Transmitter.
READY	0	Rd	When set to one, this HDCP Repeater has built the list of downstream Receiver IDs and computed the verification value *V'*. READY must be reset by the HDCP Repeater as soon as *Rxinfo* has been read by the HDCP Transmitter. This value is always zero during the computation of *V'*. READY bit must be set less than three seconds from the time the transmitter finishes writing the SKE_Send_Eks message parameters i.e. from the time the last byte of r_{iv} has been written.

Table 2-8. *RxStatus* Register Bit Field Definitions

Name	Bit Field	Rd/ Wr	Description
Rsvd	15:12	Rd	Reserved. Read as zero.
DEPTH	11:9	Rd	Repeater cascade depth. This value gives the number of attached levels through the connection topology.
DEVICE_COUNT	8:4	Rd	Total number of attached downstream devices. Always zero for HDCP Receivers. This count does not include the HDCP Repeater itself, but only devices downstream from the HDCP Repeater.
MAX_DEVS_EXCEEDED	3	Rd	Topology error indicator. When set to one, more than 31 downstream devices are attached.
MAX_CASCADE_EXCEEDED.	2	Rd	Topology error indicator. When set to one, more than four levels of repeaters have been cascaded together.
HDCP2_LEGACY_DEVICE_DOWNSTREAM	1	Rd	When set to one, indicates presence of an HDCP2.0-compliant Device or HDCP 2.1-compliant Device in the topology.
HDCP1_DEVICE_DOWNSTREAM	0	Rd	When set to one, indicates presence of an HDCP 1.x-compliant Device in the topology.

Table 2-9. *RxInfo* Register Bit Field Definitions

Name	Bit Field	Rd/ Wr	Description
VERSION	23:16	Rd	The HDCP Receiver must set VERSION to 0x02.
RECEIVER_CAPABILITY_MASK	15:2	Rd	Reserved. Read as zero.
HDCP_CAPABLE	1	Rd	When set to one, indicates that the receiver is HDCP capable. This bit does not change while the HDCP Receiver is active.
REPEATER	0	Rd	When set to one, this HDCP Receiver supports downstream connections as permitted by the Digital Content Protection LLC license. This bit does not change while the HDCP Receiver is active.

Table 2-10. *RxCaps* Register Bit Field Definitions

Name	Bit Field	Rd/ Wr	Description
VERSION	23:16	Wr	The HDCP Transmitter must set VERSION to 0x02.
TRANSMITTER_CAPABILITY_MASK	15:0	Wr	Reserved. Read as zero.

Table 2-11. *TxCaps* Register Bit Field Definitions

3 HDCP Encryption

3.1 Data Encryption

HDCP Encryption is applied in the DisplayPort transmitter at the input of the PHY layer before inter-lane skewing is applied, and in the DisplayPort receiver at the output of the data scrambler after inter-lane de-skewing has been applied (Figure 3-1 and Figure 3-2). HDCP Encryption consists of a bit-wise exclusive-OR (XOR) of the 32-bit HDCP Content with a 32-bit block of pseudo-random bits produced by the HDCP Cipher. The HDCP Cipher produces a new 128-bit block of pseudo-random bits, referred to as key stream, for every input HDCP Cipher clock pulse.

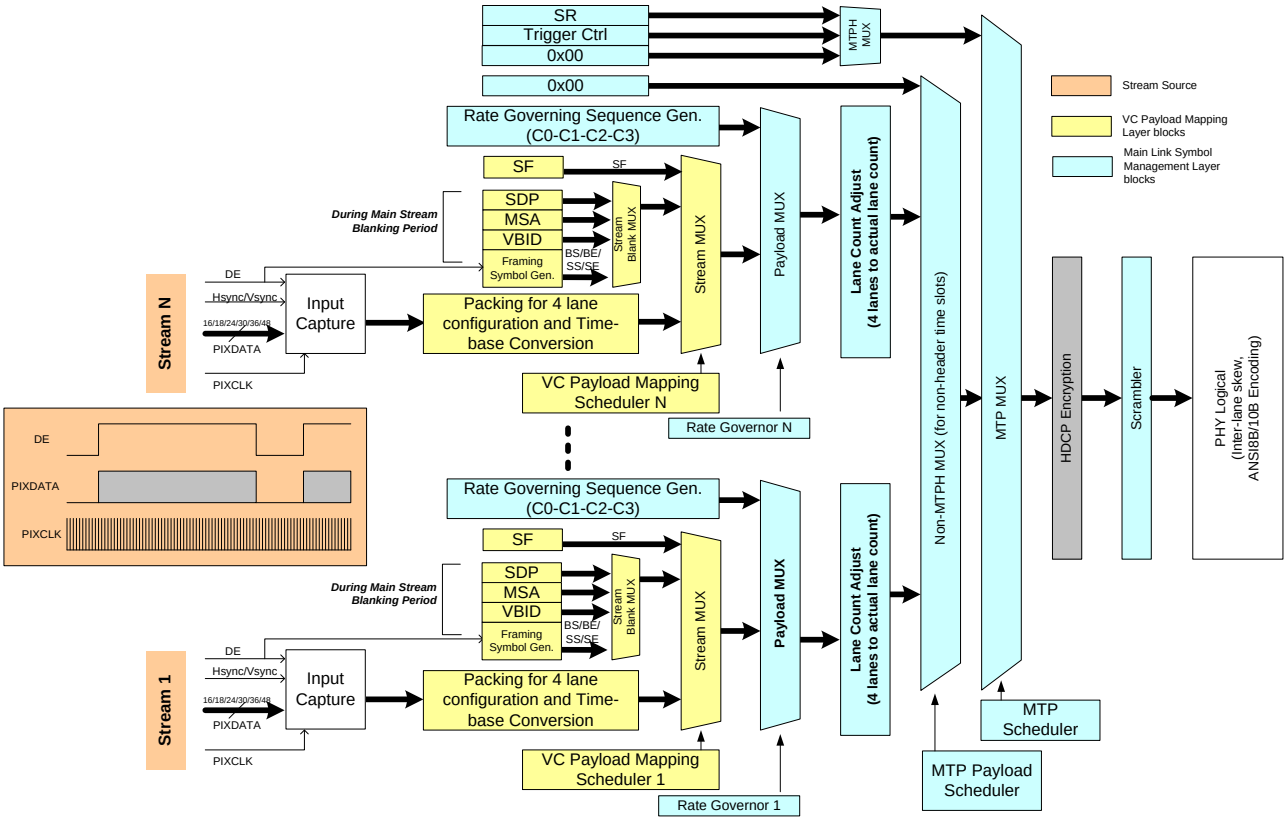


Figure 3-1. HDCP Encryption in the DisplayPort Transmitter

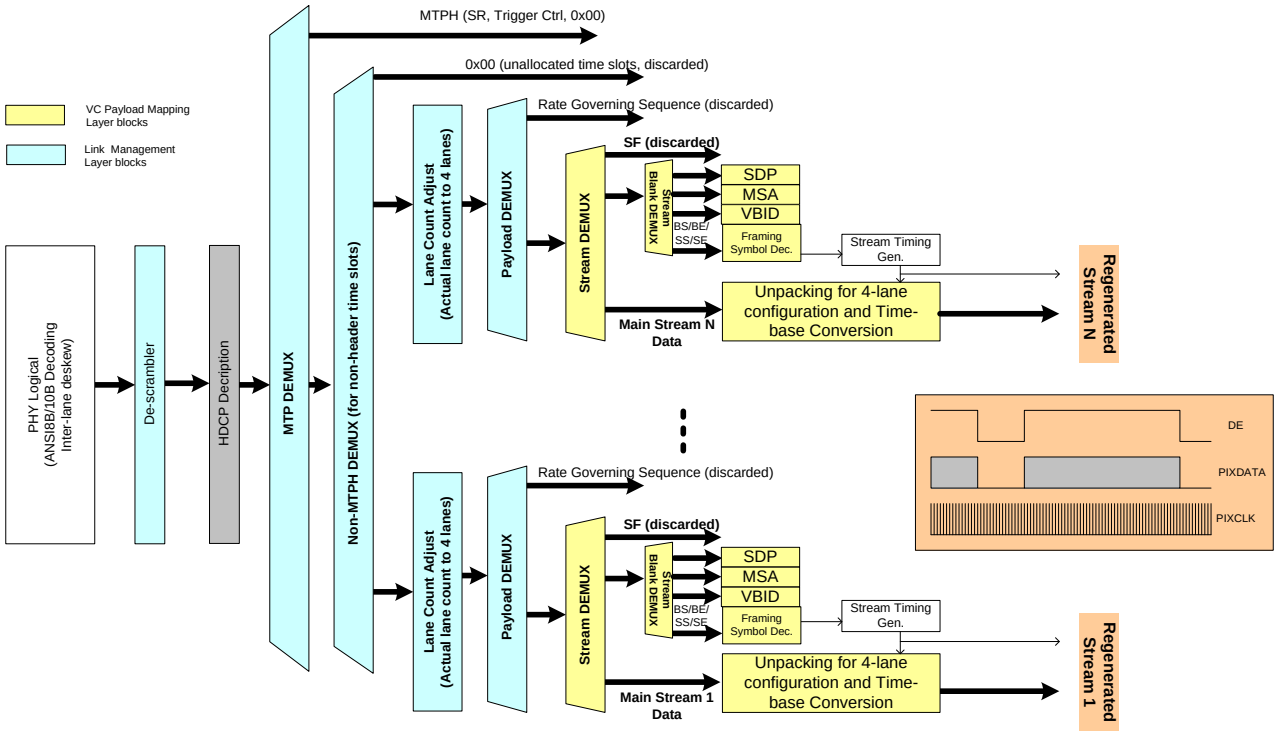


Figure 3-2. HDCP Decryption in the DisplayPort Receiver

When HDCP Encryption is applied to a timeslot in the MST mode or when HDCP Encryption is enabled in the SST mode, all data symbols (including video data, secondary data and dummy symbols) must be encrypted and K-codes must not be encrypted. Section 3.3 and Section 3.4 explains in detail the encryption signaling protocol that is used to enable/disable HDCP Encryption.

The HDCP Cipher is clocked at the following rates.

- For 4-lane Main link configurations, the HDCP Cipher is clocked for every four link symbol clocks (LS_CLK/4)
- For 2-lane Main link configurations, the HDCP Cipher is clocked at LS_CLK/8
- For 1-lane Main link configurations, the HDCP Cipher is clocked LS_CLK/16

Unless otherwise specified, all references to clock in this specification denote the HDCP Cipher clock. The clock is applied regardless of K-codes or data symbols.

The mappings of the 128-bit HDCP Cipher key stream to the DisplayPort Lanes are shown in Table 3-1, Table 3-2 and Table 3-3.

DisplayPort Lane	Cipher Key Stream	Symbol
3	127:120	15

	95:88	11
	63:56	7
	31:24	3
2	119:112	14
	87:80	10
	55:48	6
	23:16	2
1	111:104	13
	79:72	9
	47:40	5
	15:8	1
0	103:96	12
	71:64	8
	39:32	4
	7:0	0

Table 3-1. Encryption Stream Mapping for 4-lane Main Link Configuration

DisplayPort Lane	Cipher Key Stream	Symbol
1	127:120	15
	111:104	13
	95:88	11
	79:72	9
	63:56	7
	47:40	5
	31:24	3
	15:8	1
0	119:112	14
	103:96	12

	87:80	10
	71:64	8
	55:48	6
	39:32	4
	23:16	2
	7:0	0

Table 3-2. Encryption Stream Mapping for 2-lane Main Link Configuration

DisplayPort Lane	Cipher Key Stream	Symbol
0	127:120	15
	119:112	14
	111:104	13
	103:96	12
	95:88	11
	87:80	10
	79:72	9
	71:64	8
	63:56	7
	55:48	6
	47:40	5
	39:32	4
	31:24	3
	23:16	2
	15:8	1
	7:0	0

Table 3-3. Encryption Stream Mapping for 1-lane Main Link Configuration

3.2 HDCP Cipher

The HDCP cipher consists of a 128-bit AES module that is operated in a Counter (CTR) mode as illustrated in Figure 3-3.

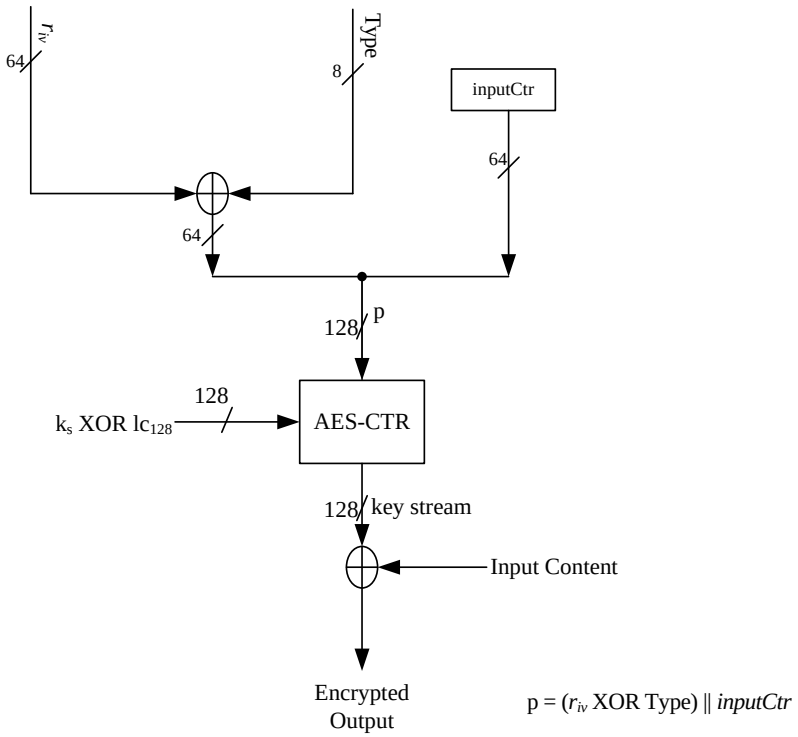


Figure 3-3. HDCP Cipher Structure

k_s is the 128-bit Session Key which is XORed with lc_{128}.

p = (r_{iv} XOR Type) || *inputCtr*. All values are in big-endian order.

The Type value, associated with the Content Stream to be encrypted, is XORed with the least significant 8-bits of r_{iv}. Type value is associated with the STREAM_ID corresponding to a Content Stream as explained in Section 4.2.12. *inputCtr* is a 64-bit counter. It is initialized to zero when HDCP Encryption is enabled for the first time during the HDCP Session i.e. immediately after SKE and must not be reset at any other time. HDCP Encryption of data symbols begins with an *inputCtr* value of zero.

The HDCP Cipher, illustrated in Figure 3-3, may be viewed as two logical HDCP Cipher modules where one module is associated with a Type value of 0x00 and the other module is associated with a Type value of 0x01. Type value is associated with the VC PayloadID corresponding to a Content Stream as explained in Section 4.2.12.

When the HDCP Cipher is clocked, it produces a 128-bit block of key stream and increments the *inputCtr* following generation of the key stream. The key stream is XORed with 32-bit data symbols as shown in Table 3-1, Table 3-2 and Table 3-3. The value of *inputCtr* must never be reused for a given set of encryption parameters i.e. k_s and r_{iv}.

When HDCP Encryption is enabled, the HDCP Cipher continues to be clocked irrespective of whether HDCP Encryption is applied to the HDCP Content i.e. the HDCP Cipher continues to be clocked even when control symbols are detected or when the XOR Enable/Disable Indicator bits indicate HDCP Encryption must not be applied to the corresponding timeslots. The HDCP Cipher is not clocked when HDCP Encryption is disabled.

3.3 Encryption Status Signaling in MST Mode

Encryption Status Signaling is the process by which an HDCP Transmitter signals to a downstream device that encryption is enabled or disabled and indicates whether or not encryption must be applied to the following data symbols.

In the MST mode, an HDCP Encryption Indicator bit is used to indicate whether HDCP Encryption is enabled or disabled. XOR Enable/Disable Indicator bits are used to indicate whether HDCP Encryption must be applied to a specific MTP timeslot. HDCP Encryption is enabled or disabled at link frame boundaries using an HDCP Encryption Indicator bit in the MTP Header.

A multi-stream link may be comprised of different Content Streams where some Content Streams may require HDCP Encryption to be applied and some for which HDCP Encryption is not required. An XOR Enable/Disable indication, corresponding to each of the 63 MTP timeslots, contained in MTP Header bits indicates whether or not HDCP Encryption must be applied to each timeslot. XOR Enable/Disable indication is valid only when HDCP Encryption is enabled as indicated by the HDCP Encryption Indicator bit. When HDCP Encryption is disabled as indicated by the HDCP Encryption Indicator bit, any further XOR Enable/Disable indication must be ignored.

The HDCP Encryption Indicator bit and the bits used for XOR Enable/Disable indication are represented by a 64-bit data structure referred to as HDCP_Encryption_Control, where

HDCP_Encryption_Control[0] = HDCP Encryption Indicator bit. When HDCP Encryption is disabled, the HDCP Encryption Indicator bit is set to 0. When HDCP Encryption is enabled, the HDCP Encryption Indicator bit is set to 1.

HDCP_Encryption_Control[1..63] = XOR Enable/Disable Indicator bits. HDCP_Encryption_Control[1] corresponds to XOR Enable/Disable indication for timeslot 1 and so on. XOR Enable/Disable Indicator bits are set to 0 (XOR Disabled) when HDCP Encryption must not be applied to the corresponding timeslots and they are set to 1 (XOR Enabled) when HDCP Encryption must be applied to the corresponding timeslots.

The 64-bit HDCP_Encryption_Control is contained in the Encryption_Control_Field which is described in the DisplayPort Specification (see References). The HDCP_Encryption_Control occupies eight MTP Headers.

The HDCP_Encryption_Control is transmitted in the Encryption_Control_Field as explained in the DisplayPort specification (see References). The HDCP_Encryption_Control consists of an 8 (scrambled) data code sequence spanning consecutive MTPH's. The data code sequence is identical per-lane, regardless of lane count. The HDCP_Encryption_Control is repeated four consecutive times, resulting in a total sequence length of 32 MTPs. HDCP Receivers apply majority voting to the repeated sequence for error correction, as described in DisplayPort specification. Unless otherwise noted, references in the HDCP specification to receiver use of the HDCP_Encryption_Control refer to the post-error corrected result.

The HDCP_Encryption_Control must be transmitted starting exactly 36 MTPs prior to each link frame boundary SR signal, and immediately prior to any standalone ACT[1] sequence. A single HDCP_Encryption_Control must be transmitted immediately preceding any (optional) back to back ACT/SR sequence.

The HDCP_Encryption_Control starting 36 MTPs before the SR is used to enable or disable HDCP Encryption in addition to indicating XOR Enable/Disable status for timeslots. The HDCP Encryption Indicator bit is valid only at link frame boundaries and HDCP Encryption is

[1] For this section, a *standalone ACT* is defined as an ACT not part of a back-to-back ACT/SR sequence

enabled/disabled only at link frame boundaries. The XOR Enable/Disable status for timeslots specified by the HDCP_Encryption_Control must also apply at the link frame boundary.

The HDCP_Encryption_Control preceding a standalone ACT must not be used to enable or disable HDCP Encryption. It is only used to indicate XOR Enable/Disable status for every timeslot. The HDCP Encryption Indicator bit in the HDCP_Encryption_Control preceding a standalone ACT must be ignored. The XOR Enable/Disable status for timeslots specified by the HDCP_Encryption_Control preceding a standalone ACT must apply starting after the MTP carrying the end of the ACT sequence.

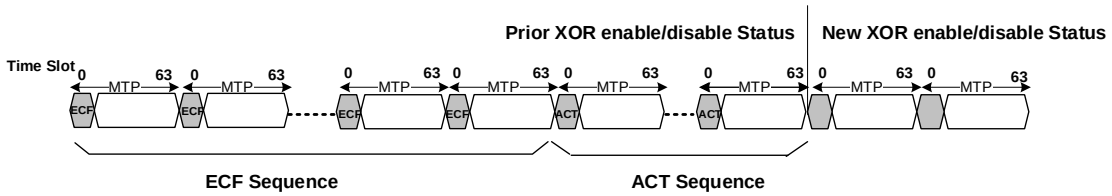


Figure 3-4: HDCP_Encryption_Control preceding a standalone ACT

In cases of the (optional) back-to-back ACT/SR sequence an HDCP Transmitter must transmit a single HDCP_Encryption_Control sequence preceding the ACT/SR pair. The HDCP_Encryption_Control preceding an ACT/SR pair is used to enable/disable HDCP Encryption in addition to indicating XOR Enable/Disable status for timeslots.

HDCP Devices must support and use Enhanced Framing Mode. Refer to the DisplayPort Specification for more details regarding enhanced framing mode.

When HDCP Encryption is enabled, the HDCP Cipher is clocked, as explained in Section 3.1, following the subsequent SR. The resulting key stream bits are used to encrypt the data symbols following SR, as explained in Section 3.2, if the XOR Enable/Disable Indicator bits indicate encryption must be applied to the corresponding timeslots. The HDCP Cipher is not clocked when HDCP Encryption is disabled.

HDCP Encryption is disabled at the link frame boundary by the transmission of the HDCP_Encryption_Control starting 36 MTPs before an SR or immediately preceding a back-to-back ACT/SR sequence with HDCP Encryption Indictor bit set to 0.

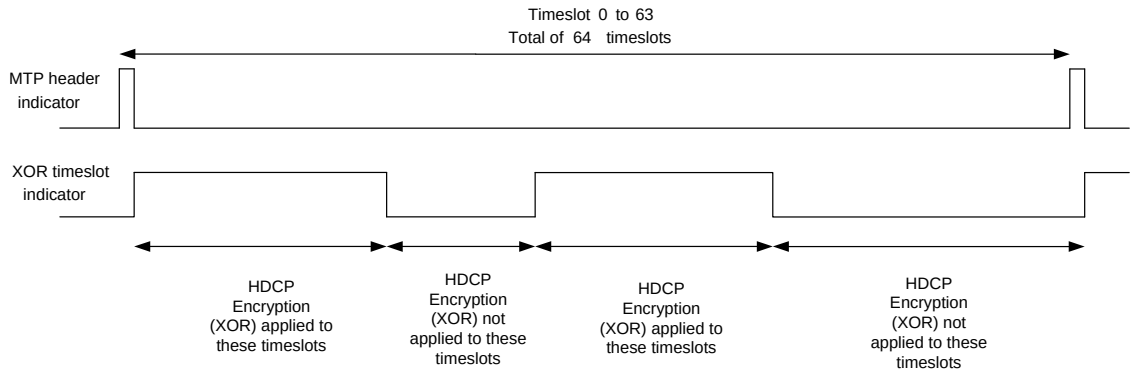


Figure 3-5. HDCP Encryption Applied Based on HDCP_Encryption_Control Bits 1:63

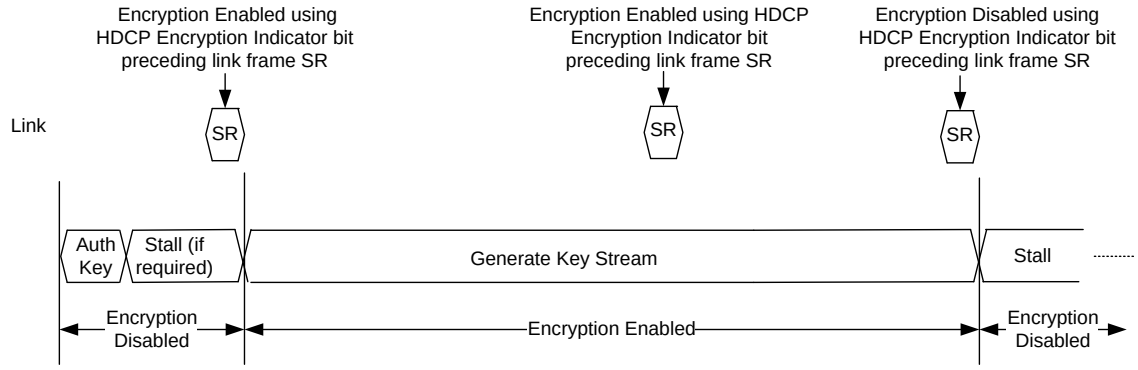


Figure 3-6. Encryption Status Signaling in MST Mode and HDCP Cipher Operations

In Figure 3-7, Figure 3-8 and Figure 3-9, X_i denotes the HDCP Cipher state that is used to generate the 128-bit keystream where i is the value of the *inputCtr* used during keystream generation. For each HDCP Cipher clock, the HDCP Cipher generates the keystream and increments the *inputCtr* following keystream generation.

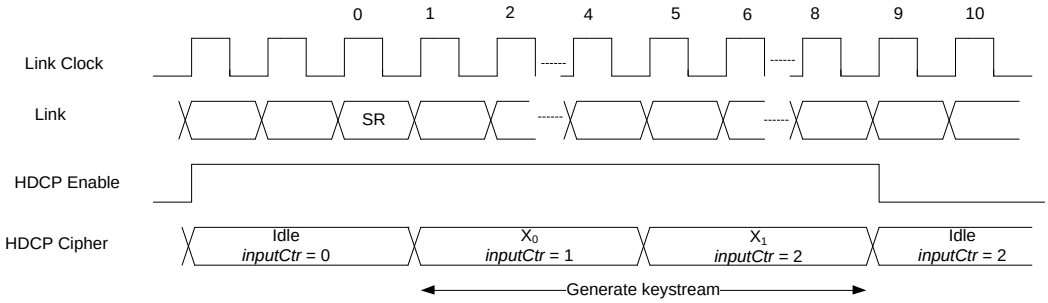


Figure 3-7. 4-Lane HDCP Cipher Operations in MST Mode

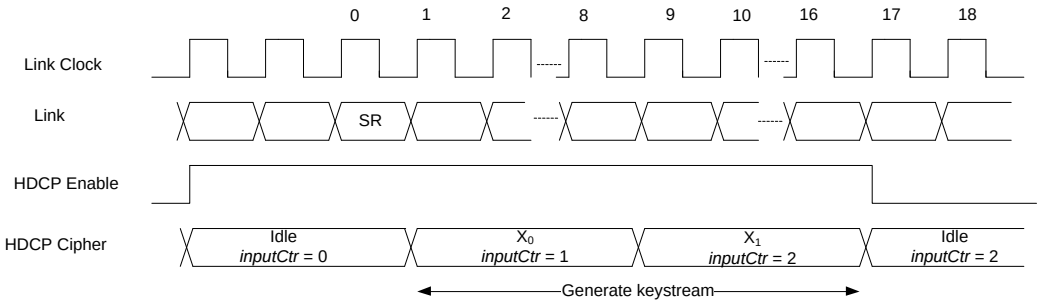


Figure 3-8. 2-Lane HDCP Cipher Operations in MST Mode

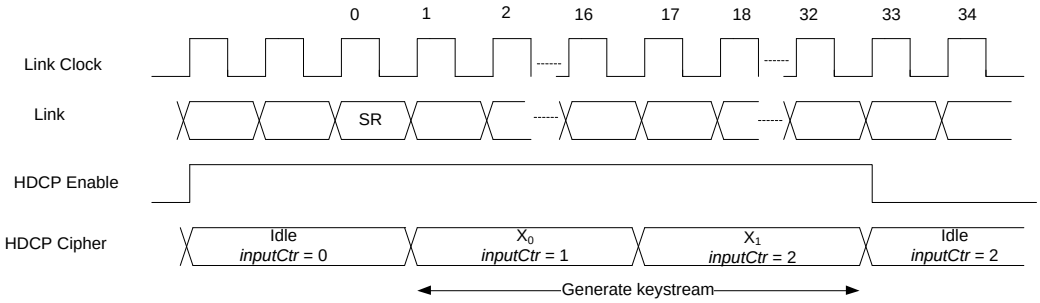


Figure 3-9. 1-Lane HDCP Cipher Operations in MST Mode

3.4 Encryption Status Signaling in SST Mode

The DisplayPort transmitter inserts a blanking start (BS) symbol after each line of video is transmitted including the last line of a frame. For audio only transmissions, a BS symbol is transmitted every 8192 symbols per lane. Every 512th BS symbol is replaced by a scrambler reset (SR) symbol. When encryption is currently disabled, the BS and SR control symbols are transmitted. When encryption is currently enabled, BS and SR control symbols are replaced by CPBS and CPSR control symbols respectively.

In the SST mode, detection of SR indicates that encryption is disabled and detection of CPSR indicates that encryption is enabled.

Following the transmission or detection of CPSR, the HDCP Cipher is clocked as explained in Section 3.1 and the resulting key stream bits are used to encrypt the data symbols following the CPSR as explained in Section 3.2. The HDCP Cipher is not clocked when HDCP Encryption is disabled.

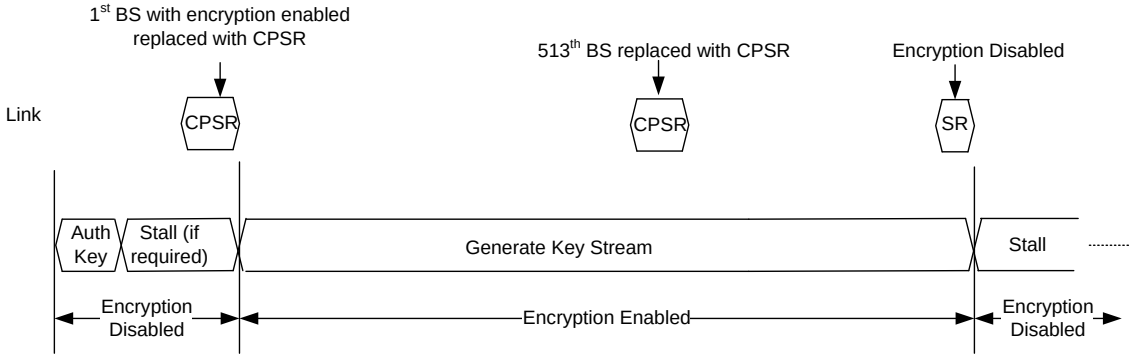


Figure 3-10. Encryption Status Signaling in SST Mode and HDCP Cipher Operations

In Figure 3-11, Figure 3-12 and Figure 3-13, X_i denotes the HDCP Cipher state that is used to generate the 128-bit keystream where i is the value of the *inputCtr* used during keystream generation. For each HDCP Cipher clock, the HDCP Cipher generates the keystream and increments the *inputCtr* following keystream generation.

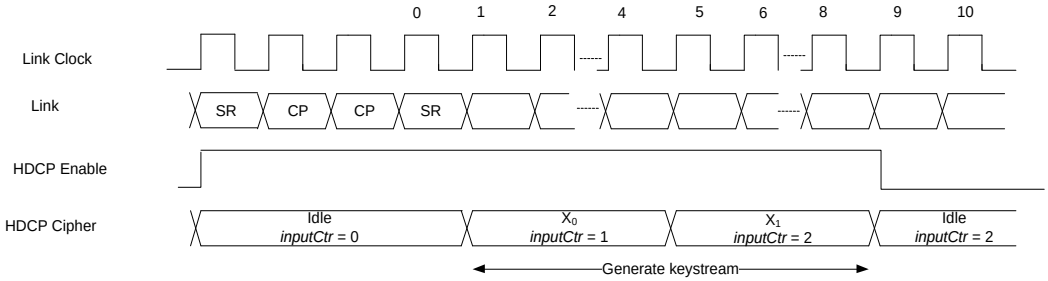


Figure 3-11. 4-Lane HDCP Cipher Operations in SST Mode

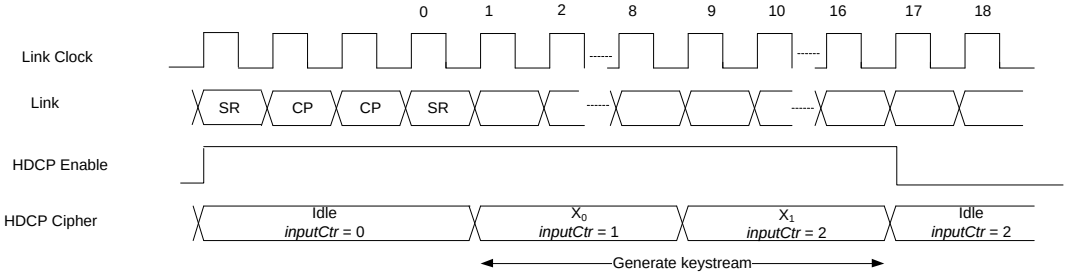


Figure 3-12. 2-Lane HDCP Cipher Operations in SST Mode

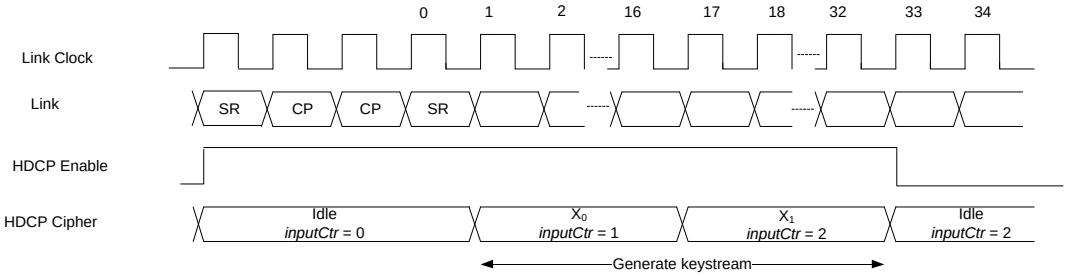


Figure 3-13. 1-Lane HDCP Cipher Operations in SST Mode

3.5 Uniqueness of k_s and r_{iv}

HDCP Receivers and HDCP Repeaters with multiple inputs may share the same Public Key Certificates and Private Keys across all inputs. The HDCP Transmitter (including downstream side of HDCP Repeater) must negotiate distinct k_m with each directly connected downstream HDCP Device. While r_{tx} used during each HDCP Session is required to be fresh, transmitters with multiple downstream HDCP links must ensure that each link receives a distinct r_{tx} value.

As illustrated in Figure 3-14, HDCP Transmitters, including downstream side of HDCP Repeaters, with multiple downstream HDCP links may share the same k_s and r_{iv} across those links only if HDCP Content from the same HDCP Cipher module following all the requirements specified in Section 3.2 is transmitted to those links.

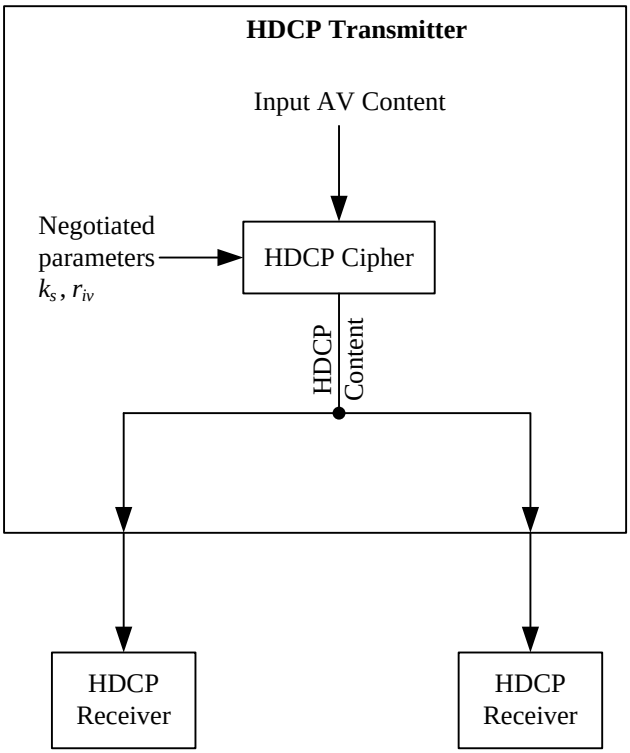


Figure 3-14. k_s and r_{iv} Shared across HDCP Links

4 Authentication Protocol Messages

4.1 Overview

Message parameters are specified in the order that they are read or written by the HDCP Transmitter. For example, in the case of the AKE_Init message, the transmitter first writes the r_{tx} parameter to the receiver followed by *TxCaps*. In the case of the AKE_Send_Cert message, the transmitter first reads the $cert_{rx}$ parameter followed by r_{rx} and finally *RxCaps*.

4.2 Message Format

4.2.1 AKE_Init (Write)

Syntax	No. of Bytes
AKE_Init {	
r_{tx}[63..0]	8
TxCaps	3
}	

Table 4-1. AKE_Init Format

4.2.2 AKE_Send_Cert (Read)

Syntax	No. of Bytes
AKE_Send_Cert {	
$cert_{rx}$[4175..0]	522
r_{rx}[63..0]	8
RxCaps	3
}	

Table 4-2. AKE_Send_Cert Format

4.2.3 AKE_No_Stored_km (Write)

Syntax	No. of Bytes
AKE_No_Stored_km {	
$E_{kpub}\_k_m$[1023..0]	128
}	

Table 4-3. AKE_No_Stored_km Format

4.2.4 AKE_Stored_km (Write)

Syntax	No. of Bytes
AKE_Stored_km{	
$E_{kh}\_k_m$[127..0]	16
m[127..0]	16
}	

Table 4-4. AKE_Stored_km Format

4.2.5 AKE_Send_H_prime (Read)

AKE_Send_H_prime must be available for the transmitter to start the read within one second after writing the AKE_No_Stored_km message i.e. after the last byte of Ekpub_km has been written, or within 200 ms after writing the AKE_Stored_km message i.e. after the last byte of m has been written.

Syntax	No. of Bytes

AK_Send_H_prime{
H´[255..0]
}
	32

Table 4-5. AKE_Send_H_prime Format

4.2.6 AKE_Send_Pairing_Info (Read)

AKE_Send_Pairing_Info must be available for the transmitter to start the read within 200 ms from the time the transmitter finishes writing the AKE_Send_H_prime message parameters to the HDCP Receiver i.e. after the last byte of *H'* has been written.

Syntax	No. of Bytes

AKE_Send_Pairing_Info{
$E_{kh}\_k_m$[127..0]
}
	16

Table 4-6. AKE_Send_Pairing_Info Format

4.2.7 LC_Init (Write)

Syntax	No. of Bytes

LC_Init {
r_n[63..0]
}
	8

Table 4-7. LC_Init Format

4.2.8 LC_Send_L_prime (Read)

The LC_Send_L_prime message must be received by the transmitter within 16ms from the time the transmitter finishes writing the LC_Init message parameters to the HDCP Receiver i.e. 16ms from the time the last byte of r_n has been written to the time the last byte of LC_Send_L_prime message has been received.

Syntax	No. of Bytes

LC_Send_L_prime{
L´[255..0]
}
	32

Table 4-8. LC_Send_L_prime Format

4.2.9 SKE_Send_Eks (Write)

Syntax	No. of Bytes

SKE_Send_Eks{
$E_{dkey}\_k_s$[127..0]
r_{iv}[63..0]
}
	
	16
	8

Table 4-9. SKE_Send_Eks Format

4.2.10 RepeaterAuth_Send_ReceiverID_List (Read)

Receiver ID list is constructed by appending *Receiver ID*s in big-endian order.

Receiver ID list = *Receiver* ID_0 || *Receiver* ID_1 || … || *Receiver* ID_{n-1}, where n is the DEVICE_COUNT.

If the computed DEVICE_COUNT for an HDCP Repeater exceeds 31, the repeater sets the *RxInfo*.MAX_DEVS_EXCEEDED bit to one. If the computed DEPTH for an HDCP Repeater exceeds four, the repeater sets *RxInfo*.MAX_CASCADE_EXCEEDED bit to one. If topology maximums are not exceeded, *RxInfo*.MAX_DEVS_EXCEEDED and *RxInfo*.MAX_CASCADE_EXCEEDED are set to zero.

The HDCP Repeater sets *RxInfo*.HDCP2_LEGACY_DEVICE_DOWNSTREAM bit to one if an HDCP 2.0-compliant Device or HDCP 2.1-compliant Device is attached to any one of its downstream ports, else it sets *RxInfo*.HDCP2_LEGACY_DEVICE_DOWNSTREAM to zero .

The HDCP Repeater sets *RxInfo*.HDCP1_DEVICE_DOWNSTREAM to one if an HDCP 1.x-compliant Device i.e. an HDCP 1.x-compliant Receiver or an HDCP 1.x-compliant Repeater is attached to any one of its downstream port, else it sets *RxInfo*.HDCP1_DEVICE_DOWNSTREAM to zero.

When the HDCP Repeater receives HDCP2_LEGACY_DEVICE_DOWNSTREAM or HDCP1_DEVICE_DOWNSTREAM bits that are set from a downstream HDCP Repeater, it must propagate this information to the upstream HDCP Transmitter by setting the corresponding bits in the RepeaterAuth_Send_ReceiverID_List message.

Syntax	No. of Bytes
RepeaterAuth_Send_ReceiverID_List{	
RxInfo	2
seq_num_V	3
V'[255..128]	16
Receiver ID List	5*DEVICE_COUNT
}	

Table 4-10. RepeaterAuth_Send_ReceiverID_List Format

4.2.11 RepeaterAuth_Send_Ack (Write)

The last byte of the RepeaterAuth_Send_Ack message i.e the last byte of *V'*[127..0] must be written to the repeater by the transmitter within two seconds from the time the READY status was set and the CP_IRQ interrupt was asserted by the HDCP Repeater and the downstream topology does not exceed specified maximums.

Syntax	No. of Bytes
RepeaterAuth_Send_Ack{	
V[127..0]	16
}	

Table 4-11. RepeaterAuth_Send_Ack Format

4.2.12 RepeaterAuth_Stream_Manage (Write)

Content Streams are assigned a Type value by the most upstream HDCP Transmitter based on instructions received from the Upstream Content Control Function.

The VC Payload ID, assigned to a Content Stream as specified in the DisplayPort specification, is followed by its assigned Type value in the RepeaterAuth_Stream_Manage message. All Content Streams transmitted by the HDCP Transmitter to the HDCP Repeater, after HDCP Encryption, are assigned Type values.

Syntax	No. of Bytes
RepeaterAuth_Stream_Manage{	
seq_num_M	3
k	2
StreamID_Type	2**k*
}	

Table 4-12. RepeaterAuth_Stream_Manage Format

StreamID_Type = VC Payload ID_1 || Type || VC Payload ID_2 || Type ||… || VC Payload ID_k || Type

VC Payload ID assigned to the Content Stream is concatenated with its assigned Type value. All values are in big-endian order.

In SST mode, the VC Payload ID is set to 0 (zero).

Parameter *k* is the number of Content Streams that are being transmitted by the HDCP Transmitter to the attached HDCP Repeater during the HDCP Session.

Parameter	No. of Bytes	Description
VC Payload ID	1	VC Payload ID, corresponding to the Content Stream, as defined in the DisplayPort specification. VC Payload ID is set to 0 (zero) in SST mode.
Type	1	0x00: Type 0 Content Streams. May be transmitted by the HDCP Repeater to all HDCP Devices. 0x01: Type 1 Content Streams. Must not be transmitted by the HDCP Repeater to HDCP 1.x-compliant Devices, HDCP 2.0-compliant Devices and HDCP 2.1-compliant Devices. 0x02 – 0xFF : Reserved for future use only. Content Streams with reserved Type values must be treated similar to Type 1 Content Streams i.e. must not be transmitted by the HDCP Repeater to HDCP 1.x-compliant Devices and HDCP 2.0-compliant Repeaters

Table 4-13. VC Payload ID, Type Description

4.2.13 RepeaterAuth_Stream_Ready (Read)

The RepeaterAuth_Stream_Ready message must be available for the transmitter to start the read within 100ms from the time the transmitter finishes writing the RepeaterAuth_Stream_Manage message parameters to the HDCP Receiver i.e. after the last byte of *StreamID_Type* has been written.

Syntax	No. of Bytes
RepeaterAuth_Stream_Ready{	
M'[255..0]	32
}	

Table 4-14. RepeaterAuth_Stream_Ready Format

5 Renewability

It is contemplated that an authorized participant in the authentication protocol may become compromised so as to expose the RSA private keys it possesses for misuse by unauthorized parties. In consideration of this, each HDCP Receiver is issued a unique Receiver ID which is contained in $cert_{rx}$. Through a process defined in the HDCP Adopter's License, the Digital Content Protection LLC may determine that an HDCP Receiver's RSA private key, $kpriv_{rx,}$ has been compromised. If so, it places the corresponding Receiver ID on a revocation list that the HDCP Transmitter checks during authentication.

The HDCP Transmitter is required to manage system renewability messages (SRMs) carrying the Receiver ID revocation list. The validity of an SRM is established by verifying the integrity of its signature with the Digital Content Protection LLC public key, which is specified by the Digital Content Protection LLC.

For interoperability with HDCP 1.x, KSVs of revoked HDCP 1.x devices will be included in the HDCP 2 SRM, in addition to the HDCP 1.x SRM. Similarly, Receiver IDs of revoked HDCP 2 devices will be included in the HDCP 1.x SRM, in addition to the HDCP 2 SRM.

The SRMs are delivered with content and must be checked when available. The Receiver IDs must immediately be checked against the SRM when a new version of the SRM is received. Additionally, devices compliant with HDCP 2.0 and higher must be capable of storing at least 5kB of the SRM in their non-volatile memory. The process by which a device compliant with HDCP 2.0 or higher updates the SRM stored in its non-volatile storage when presented with a newer SRM version is explained in Section 5.2.

5.1 SRM Size and Scalability

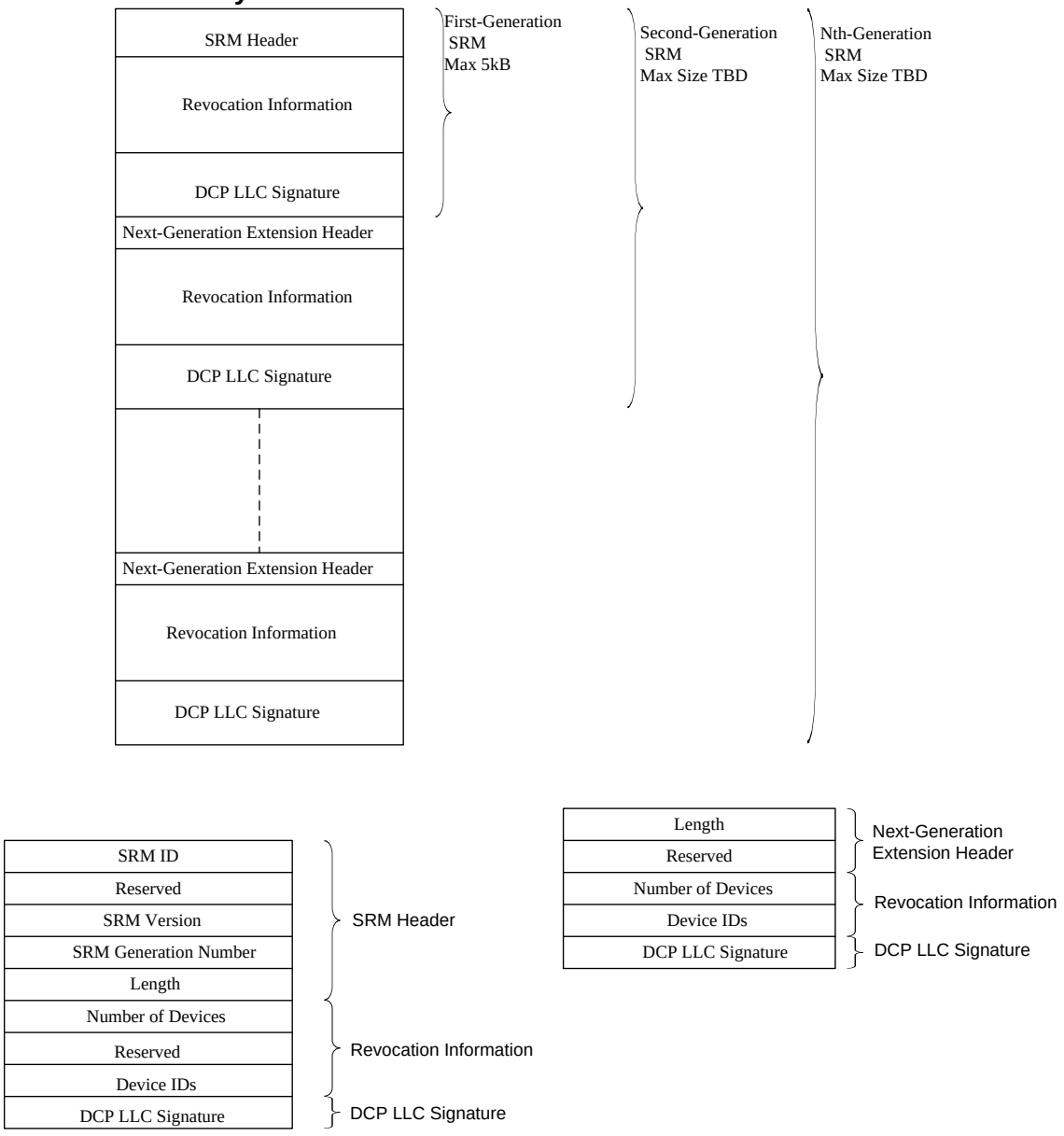


Figure 5-1. SRM Generational Format

As illustrated in Figure 5-1, the size of the First-Generation HDCP SRM will be limited to a maximum of 5kB. The actual size of the First-Generation SRM is 5116 bytes. For scalability of the SRM, the SRM format supports next-generation extensions. By supporting generations of SRMs, an HDCP SRM can, if required in future, grow beyond the 5kB limit to accommodate more Receiver IDs. Next-generation extensions are appended to the current-generation SRM in order to ensure backward compatibility with devices that support only previous-generation SRMs.

Table 5-1 gives the format of the HDCP 2 SRM. All values are stored in big endian format.

Name	Size (bits)	Function
SRM ID	4	A value of 0x9 signifies that the message is for HDCP 2. All other

		values are reserved. SRMs with values other than 0x9 must be ignored.
HDCP2 Indicator	4	A value of 0x1 signifies that the message is for HDCP2.
Reserved	8	Reserved for future definition. Must be 0x00.
SRM Version	16	Sequentially increasing unique SRM numbers. Higher numbered SRMs are more recent.
SRM Generation Number	8	Indicates the generation of the SRM. The generation number starts at 1 and increases sequentially.
Length	24	Length in bytes and includes the combined size of this field (three bytes) and all following fields contained in the first-generation SRM i.e. size of this field, Number of Devices field, Reserved (22 bits) field, Device IDs field and Digital Content Protection LLC signature field (384 bytes) in the first-generation SRM.
Number of Devices	10	Specifies the number (N1) of Receiver IDs / KSVs contained in the first-generation SRM.
Reserved	22	Reserved for future definition. All bits set to 0.
Device IDs	40 * N1 Max size for this field is 37760 (4720 bytes)	40-bit Receiver IDs / KSVs.
DCP LLC Signature	3072	A cryptographic signature calculated over all preceding fields of the SRM. RSASSA-PKCS1-v1_5 is the signature scheme used as defined by PKCS #1 V2.1: RSA Cryptography Standard. SHA-256 is the underlying hash function.

Table 5-1. System Renewability Message Format

Each subsequent next-generation extensions to the first-generation SRM will have the following fields.

Name	**Size (bits)**	**Function**
Length	16	Length in bytes and includes the combined size of this field (two bytes) and all following fields contained in this next-generation extension i.e. size of this field, Number of Devices field, Reserved (6 bits) field, Device IDs field and Digital Content Protection LLC signature field (384 bytes) in this next-generation SRM.
Reserved	6	Reserved for future definition. All bits set to 0.
Number of Devices	10	Specifies the number (N2) of Receiver IDs / KSVs contained in this next generation extension.
Device IDs	40 * N2	40-bit Receiver IDs / KSVs.
DCP LLC Signature	3072	A cryptographic signature calculated over all preceding fields of the SRM. RSASSA-PKCS1-v1_5 is the signature scheme used as defined by PKCS #1 V2.1: RSA Cryptography Standard. SHA-256 is the underlying hash function.

Table 5-2. Next-generation extension format

5.2 Updating SRMs

The stored HDCP SRM must be updated when a newer version of the SRM is delivered with the content. The procedure for updating an SRM is as follows:

1. Verify that the version number of the new SRM is greater than the version number of the SRM currently stored in the device's non-volatile storage

2. If the version number of the new SRM is greater (implying that it is a more recent version), verify the signature on the new SRM

On successful signature verification, replace the current SRM in the device's non-volatile storage with the new SRM. If, for instance, the device supports only second-generation SRMs and the new SRM is a third-generation SRM, the device is not required to store the third-generation extension. Devices compliant with HDCP 2.0 or higher must be capable of storing at least 5kB (actual size is 5116 bytes) of the SRM (First-Generation SRM).

Appendix A. Core Functions and Confidentiality and Integrity of Values

Table A-1 identifies the requirements of confidentiality and integrity for values within the protocol. A *confidential* value must never be revealed. The *integrity* of many values in the system is protected by fail-safe mechanisms of the protocol. Values that are not protected in this manner require active measures beyond the protocol to ensure integrity. Such values are noted in the table as requiring integrity. Core Functions must be implemented in Hardware. The values used by Core Functions, along with the corresponding Core Functions by which they are used, are identified in the table.

Value	Confidentiality Required[±]?	Integrity Required[±]?	Value used by Core Functions?	Core Function
lc_{128}	Yes	Yes	Yes	HDCP Encryption and Decryption
$kpub_{dcp}$	No	Yes	No	N/A
$cert_{rx}$	No	No	No	N/A
$kpub_{rx}$	No	Yes	No	N/A
Receiver ID	No	Yes	No	N/A
$kpriv_{rx}$	Yes	Yes	Yes	Handling of Device Secret Key, during AKE, in plaintext form
r_{tx}	No	Yes*	Yes	
r_{iv}	No	Yes*	Yes	N/A
REPEATER	No	Yes	No	N/A
r_{rx}	No	Yes**	Yes	N/A
k_m	Yes	Yes*	Yes	Handling of Master Key, during AKE (including Pairing) and Key Derivation, in plaintext form
k_d	Yes	Yes*	No	N/A
$dkey_0$,$dkey_1$	Yes	Yes*	No	N/A

[±] According to the robustness rules in the HDCP Adopter's License

* Only within the transmitter
* Only within the transmitter
** Only within the receiver

$dkey_2$	Yes	Yes*	Yes	Handling of information or materials during Key Derivation and SKE, including but not limited to cryptographic keys used to encrypt or decrypt HDCP Core Keys (k_s), from which HDCP Core Keys could reasonably be derived
ctr	No	Yes*	Yes	N/A
H	Yes	Yes	No	N/A
H'	No	No	No	N/A
m	No	No	Yes	N/A
k_h	Yes	Yes	Yes	Handling of information or materials during Pairing, including but not limited to cryptographic keys used to encrypt or decrypt HDCP Core Keys (k_m), from which HDCP Core Keys could reasonably be derived
r_n	No	Yes*	Yes	N/A
L	Yes	Yes	No	N/A
L'	No	No	No	N/A
k_s	Yes	Yes*	Yes	Handling of Session Key, during SKE and HDCP Encryption/Decryption, in plaintext form
V	Yes	Yes	No	N/A
V'	No	No	No	N/A
Receiver ID list	No	Yes	No	N/A
DEPTH	No	Yes	No	N/A
DEVICE_COUNT	No	Yes	No	N/A
MAX_DEVS_EXCEEDED	No	Yes	No	N/A
MAX_CASCADE_EXCEEDED	No	Yes	No	N/A
inputCtr	No	Yes*	Yes	HDCP Encryption and Decryption

Type	No	Yes*	Yes	HDCP Encryption and Decryption
p	No	Yes*	Yes	HDCP Encryption and Decryption

Table A-1. Core Functions and Confidentiality and Integrity of Values

Appendix B. DCP LLC Public Key

Table B-1 gives the production DCP LLC public key.

Parameter	Value (hexadecimal)
Modulus n	B0E9 AA45 F129 BA0A 1CBE 1757 28EB 2B4E 8FD0 C06A AD79 980F 8D43 8D47 04B8 2BF4 1521 5619 0140 013B D091 9062 9E89 C227 8ECF B6DB CE3F 7210 5093 8C23 2983 7B80 64A7 59E8 6167 4CBC D858 B8F1 D4F8 2C37 9816 260E 4EF9 4EEE 24DE CCD1 4B4B C506 7AFB 4965 E6C0 0083 481E 8E42 2A53 A0F5 3729 2B5A F973 C59A A1B5 B574 7C06 DC7B 7CDC 6C6E 826B 4988 D41B 25E0 EED1 79BD 3985 FA4F 25EC 7019 23C1 B9A6 D97E 3EDA 48A9 58E3 1814 1E9F 307F 4CA8 AE53 2266 2BBE 24CB 4766 FC83 CF5C 2D1E 3AAB AB06 BE05 AA1A 9B2D B7A6 54F3 632B 97BF 93BE C1AF 2139 490C E931 90CC C2BB 3C02 C4E2 BDBD 2F84 639B D2DD 783E 90C6 C5AC 1677 2E69 6C77 FDED 8A4D 6A8C A3A9 256C 21FD B294 0C84 AA07 2926 46F7 9B3A 1987 E09F EB30 A8F5 64EB 07F1 E9DB F9AF 2C8B 697E 2E67 393F F3A6 E5CD DA24 9BA2 7872 F0A2 27C3 E025 B4A1 046A 5980 27B5 DAB4 B453 973B 2899 ACF4 9627 0F7F 300C 4AAF CB9E D871 2824 3EBC 3515 BE13 EBAF 4301 BD61 2454 349F 733E B510 9FC9 FC80 E84D E332 968F 8810 2325 F3D3 3E6E 6DBB DC29 66EB
Public Exponent e	03

Table B-1. DCP LLC Public Key

Appendix C. Bibliography (Informative)

These documents are not normatively referenced in this specification, but may provide useful supplementary information.

ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 (2006) Amendment 1 (Jan. 2007), *Transport of MPEG-4 streaming text and MPEG-4 lossless audio over MPEG-2 systems*

ITU-T Recommendation H.222.0 / ISO/IEC 13818-1 (2006) Amendment 2 (Aug. 2007), *Carriage of auxiliary video data*

SMPTE 2022-1-2007, *Forward Error Correction for Real-Time Video/Audio Transport Over IP Networks*, May 2007

SMPTE 2022-2-2007, *Unidirectional Transport of Constant Bit Rate MPEG-2 Transport Streams on IP Networks*, May 2007

Interoperability for Professional Video Streaming over IP Networks, SMPTE Motion Imaging Journal, Feb./March 2005, http://www.broadcastpapers.com/whitepapers/Path1InteropVideoIP.pdf?CFID=16660544&CFTOKEN=dd0a39cb99517fc5-3203F7CF-F879-0B3E-45C4A402626C372C

Appendix D. Test Vectors

D.1 Facsimile Keys

Note: The facsimile keys provided must be used ONLY for test purposes.

All values are provided in big-endian order.

Table D.1 provides facsimile key information for transmitter T1.

	Value in Hex
Global Constant lc_{128}	93 ce 5a 56 a0 a1 f4 f7 3c 65 8a 1b d2 ae f0 f7

Table D.1

Table D.2 provides the facsimile public parameters associated with the DCP LLC key $kpub_{dcp}$. These parameters are used only for test purposes. They are not used in production devices or SRMs.

	Value in Hex
Modulus n	A2 C7 55 57 54 CB AA A7 7A 27 92 C3 1A 6D C2 31 CF 12 C2 24 BF 89 72 46 A4 8D 20 83 B2 DD 04 DA 7E 01 A9 19 EF 7E 8C 47 54 C8 59 72 5C 89 60 62 9F 39 D0 E4 80 CA A8 D4 1E 91 E3 0E 2C 77 55 6D 58 A8 9E 3E F2 DA 78 3E BA D1 05 37 07 F2 88 74 0C BC FB 68 A4 7A 27 AD 63 A5 1F 67 F1 45 85 16 49 8A E6 34 1C 6E 80 F5 FF 13 72 85 5D C1 DE 5F 01 86 55 86 71 E8 10 33 14 70 2A 5F 15 7B 5C 65 3C 46 3A 17 79 ED 54 6A A6 C9 DF EB 2A 81 2A 80 2A 46 A2 06 DB FD D5 F3 CF 74 BB 66 56 48 D7 7C 6A 03 14 1E 55 56 E4 B6 FA 38 2B 5D FB 87 9F 9E 78 21 87 C0 0C 63 3E 8D 0F E2 A7 19 10 9B 15 E1 11 87 49 33 49 B8 66 32 28 7C 87 F5 D2 2E C5 F3 66 2F 79 EF 40 5A D4 14 85 74 5F 06 43 50 CD DE 84 E7 3C 7D 8E 8A 49 CC 5A CF 73 A1 8A 13 FF 37 13 3D AD 57 D8 51 22 D6 32 1F C0 68 4C A0 5B DD 5F 78 C8 9F 2D 3A A2 B8 1E 4A E4 08 55 64 05 E6 94 FB EB 03 6A 0A BE 83 18 94 D4 B6 C3 F2 58 9C 7A 24 DD D1 3A B7 3A B0 BB E5 D1 28 AB AD 24 54 72 0E 76 D2 89 32 EA 46 D3 78 D0 A9 67 78 C1 2D 18 B0 33 DE DB 27 CC B0 7C C9 A4 BD DF 2B 64 10 32 44 06 81 21 B3 BA CF 33 85 49 1E 86 4C BD F2 3D 34 EF D6 23 7A 9F 2C

	DA 84 F0 83 83 71 7D DA 6E 44 96 CD 1D 05 DE 30 F6 1E 2F 9C 99 9C 60 07
Public Exponent e	03

Table D.2

Table D.3 and Table D.4 provide the facsimile certificates ($cert_{rx}$) for receivers R1 and R2.

As provided in Table 2.1 of High-bandwidth Digital Content Protection System, Revision 2.2, Mapping HDCP to DisplayPort specification, the certificate format consists of 40-bit Receiver ID, followed by 1048-bit Receiver Public Key, 4-bit Reserved2, 12-bit Reserved1 and 3072-bit Signature fields. All values are stored in big-endian format.

For example, in Table D.3, 0x745bb8bd04 is the Receiver ID which is followed by Receiver Public Key, Reserved2, Reserved1 and Signature fields.

	Value (Sequence of Hexadecimal bytes) for R1
Certificate ($cert_{rx}$)	74 5b b8 bd 04 af b5 c5 c6 7b c5 3a 34 90 a9 54 c0 8f b7 eb a1 54 d2 4f 22 de 83 f5 03 a6 c6 68 46 9b c0 b8 c8 6c db 26 f9 3c 49 2f 02 e1 71 df 4e f3 0e c8 bf 22 9d 04 cf bf a9 0d ff 68 ab 05 6f 1f 12 8a 68 62 eb fe c9 ea 9f a7 fb 8c ba b1 bd 65 ac 35 9c a0 33 b1 dd a6 05 36 af 00 a2 7f bc 07 b2 dd b5 cc 57 5c dc c0 95 50 e5 ff 1f 20 db 59 46 fa 47 c4 ed 12 2e 9e 22 bd 95 a9 85 59 a1 59 3c c7 83 01 00 01 10 00 0b a3 73 77 dd 03 18 03 8a 91 63 29 1e a2 95 74 42 90 78 d0 67 25 b6 32 2f cc 23 2b ad 21 39 3d 14 ba 37 a3 65 14 6b 9c cf 61 20 44 a1 07 bb cf c3 4e 95 5b 10 cf c7 6f f1 c3 53 7c 63 a1 8c b2 e8 ab 2e 96 97 c3 83 99 70 d3 dc 21 41 f6 0a d1 1a ee f4 cc eb fb a6 aa b6 9a af 1d 16 5e e2 83 a0 4a 41 f6 7b 07 bf 47 85 28 6c a0 77 a6 a3 d7 85 a5 c4 a7 e7 6e b5 1f 40 72 97 fe c4 81 23 a0 c2 90 b3 49 24 f5 b7 90 2c bf fe 04 2e 00 a9 5f 86 04 ca c5 3a cc 26 d9 39 7e a9 2d 28 6d c0 cc 6e 81 9f b9 b7 11 33 32 23 47 98 43 0d a5 1c 59 f3 cd d2 4a b7 3e 69 d9 21 53 9a f2 6e 77 62 ae 50 da 85 c6 aa c4 b5 1c cd a8 a5 dd 6e 62 73 ff 5f 7b d7 3c 17 ba 47 0c 89 0e 62 79 43 94 aa a8 47 f4 4c 38 89 a8 81 ad 23 13 27 0c 17 cf 3d 83 84 57 36 e7 22 26 2e 76 fd 56 80 83 f6 70 d4 5c 91 48 84 7b 18 db 0e 15 3b 49 26 23 e6 a3 e2 c6 3a 23 57 66 b0 72 b8 12 17 4f 86 fe 48 0d 53 ea fe 31 48 7d 86 de eb 82 86 1e 62 03 98 59 00 37 eb 61 e9 f9 7a 40 78 1c ba bc 0b 88 fb fd 9d d5 01 11 94 e0 35 be 33 e8 e5 36 fb 9c 45 cb 75 af d6 35 ff 78 92 7f a1 7c a8 fc b7 f7 a8 52 a9 c6 84 72 3d 1c c9 df 35 c6 e6 00 e1 48 72 ce 83 1b cc f8 33 2d 4f 98 75 00 3c 41 df 7a ed 38 53 b1

Table D.3

	Value (Sequence of Hexadecimal bytes) for R2
Certificate ($cert_{rx}$)	8b a4 47 42 fb e4 68 63 8a da 97 2d de 9a 8d 1c b1 65 4b 85 8d e5 46 d6 db 95 a5 f6 66 74 ea 81 0b 9a 58 58 66 26 86 a6 b4 56 2b 29 43 e5 bb 81 74 86 a7 b7 16 2f 07 ec d1 b5 f9 ae 4f 98 89 a9 91 7d 58 5b 8d 20 d5 c5 08 40 3b 86 af f4 d6 b9 20 95 e8 90 3b 8f 9f 36 5b 46 b6 d4 1e f5 05 88 80 14 e7 2c 77 5d 6e 54 e9 65 81 5a 68 92 a5 d6 40 78 11 97 65 d7 64 36 5e 8d 2a 87 a8 eb 7d 06 2c 10 f8 0a 7d 01 00 01 10 00 06 40 99 8f 5a 54 71 23 a7 6a 64 3f bd dd 52 b2 79 6f 88 26 94 9e af a4 de 7d 8d 88 10 c8 f6 56 f0 8f 46 28 48 55 51 c5 af a1 a9 9d ac 9f b1 26 4b eb 39 ad 88 46 af bc 61 a8 7b f9 7b 3e e4 95 d9 a8 79 48 51 00 be a4 b6 96 7f 3d fd 76 a6 b7 bb b9 77 dc 54 fb 52 9c 79 8f ed d4 b1 bc 0f 7e b1 7e 70 6d fc b9 7e 66 9a 86 23 3a 98 5e 32 8d 75 18 54 64 36 dd 92 01 39 90 b9 e3 af 6f 98 a5 c0 80 c6 2f a1 02 ad 8d f4 d6 66 7b 45 e5 74 18 b1 27 24 01 1e ea d8 f3 79 92 e9 03 f5 57 8d 65 2a 8d 1b f0 da 58 3f 58 a0 f4 b4 be cb 21 66 e9 21 7c 76 f3 c1 7e 2e 7c 3d 61 20 1d c5 c0 71 28 2e b7 0f 1f 7a c1 d3 6a 1e

	a3 54 34 8e 0d d7 96 93 78 50 c1 ee 27 72 3a bd 57 22 f0 d7 6d 9d 65 c4 07 9c 82 a6 d4 f7 6b 9a e9 c0 6c 4a 4f 6f be 8e 01 37 50 3a 66 d9 e9 d9 f9 06 9e 00 a9 84 a0 18 b3 44 21 24 a3 6c cd b7 0f 31 2a e8 15 b6 93 6f b9 86 e5 28 01 1a 5e 10 3f 1f 4d 35 a2 8d b8 54 26 68 3a cd cb 5f fa 37 4a 60 10 b1 0a fe ba 9b 96 5d 7e 99 cf 01 98 65 87 ad 40 d5 82 1d 61 54 a2 d3 16 3e f7 e3 05 89 8d 8a 50 87 47 be 29 18 01 b7 c3 dd 43 23 7a cd 85 1d 4e a9 c0 1a a4 77 ab e7 31 9a 33 1b 7a 86 e1 e5 ca 0c 43 1a fa ec 4c 05 c6 d1 43 12 f9 4d 3e f7 d6 05 9c 1c dd

Table D.4

Table D.5 and Table D.6 provide the private keys for receivers R1 and R2.

	Value in Hex for R1
P	ec be e5 5b 9e 7a 50 8a 96 80 c8 db b0 ed 44 f2 ba 1d 5d 80 c1 c8 b3 c2 74 de ee 28 ec dc 78 c8 67 53 07 f2 f8 75 9c 4c a5 6c 48 94 c8 eb ad d7 7d d2 ea df 74 20 62 c9 81 a8 3c 36 b9 ea 40 fd
Q	be 00 19 76 c6 b4 ba 19 d4 69 fa 4d e2 f8 30 27 36 2b 4c c4 34 ab d3 d9 8c d6 b8 0d 37 5e 59 4b 76 70 68 2b 1f 4c 3d 47 5f a5 b1 cd 74 56 88 fe 7c f8 3b 30 6f fd c3 ed 87 3c a1 53 84 c3 d2 7f
d mod (p-1)	60 71 9b e9 e8 f3 97 1f fe 13 d4 bf 7a a2 0d f6 7b cf 3e aa 17 47 75 c3 7f ec d9 44 9e c9 6a 02 e9 e4 af 56 51 d5 47 a9 09 b2 c5 16 a7 8b 2b 34 a0 33 6e 2f 3d 95 7b e8 ef 02 e4 14 bf 44 28 d9
d mod (q-1)	10 0e 2e 18 ad 5d e4 43 fe 81 1e 17 aa d0 52 31 5e 10 76 a2 35 d9 37 43 b0 f5 0c 04 81 e3 45 24 6d 53 be 59 b6 81 58 c4 49 3e d5 31 89 5d 2e a2 62 a9 0f 47 5e 8f 51 19 27 4e 66 4b 8a 72 89 bd
q^-1 mod p	3e 53 0a f4 8e 75 e1 52 c6 24 e9 f7 bb ac 3f 22 5f e8 e0 79 35 ff 91 ee 22 56 d2 00 68 32 c4 e1 5f ff f8 b1 1d ee dc 57 81 d1 ab 8b 37 22 e3 9f d0 a1 c1 ce 1d d0 24 23 a0 0e f7 a6 db a3 ea d3

Table D.5

	Value in Hex for R2
P	f5 f6 fa 44 a2 16 2f a7 1f 7f 16 05 99 26 c4 1b 80 7f fa 52 4e 3e aa 3d 1e b0 f1 9a c6 3d 8f 57 2b 9e cd e8 03 d6 f3 91 75 e2 19 44 9e 11 58 5f d6 88 7c c4 c1 5b 45 9b 84 cf 72 1d 35 bf 24 d5
q	ed ba 08 bf 42 2c 0e fa 3a c4 d2 c7 01 51 25 ae b0 a1 cc db 67 9b aa 50 f0 80 ac 4b 9f 5c ba 1e f4 7f a9 b3 21 8b 62 2c 36 da cd a7 4d a4 d6 44 ed b1 34 e7 69 10 77 5a 6a ff f5 63 8a 2c 43 09
d mod (p-1)	61 5a c4 6c 6e 0b 82 09 10 3a 69 29 06 19 85 fd ac ba fb 05 a0 da c4 df 34 4a ad 16 a9 e8 ab d7 c0 f8 36 5f e3 45 2d 5b 21 e1 c0 46 9c 9a 18 f4 b6 21 87 e1 08 f7 6b 71 c6 fb a5 1b 52 ae b9 91
d mod (q-1)	5a 83 7f bb 1a bd dd c2 06 c8 54 1c b3 72 ab 2f 55 4f 75 c9 80 2c 73 ef b7 72 b6 a7 60 79 14 e0 9e 65 51 3e c4 21 e6 f2 40 bc 94 9b 03 e4 24 35 40 6f 3d 5e 72 d1 73 30 39 17 55 de 5d 88 b6 c9
q^-1 mod p	bc 91 2a 93 6a 8d 24 3c d5 7d 12 3b a3 71 c7 3a f0 64 72 50 7e 18 71 e1 b4 3b 1e fc 38 ca e6 8c 16 51 97 d6 3f 04 ee 23 8b 45 0c 4b 98 36 18 27 29 1b 4d 73 7e e8 b0 1a c7 fb 5c ea 78 d0 6e 97

Table D.6

Table D.7 provides the global constant (lc_{128}) used for receivers R1 and R2. Note that the same global constant is used in T1, R1 and R2.

	Value in Hex for R1	**Value in Hex for R2**
Global Constant lc_{128}	93 ce 5a 56 a0 a1 f4 f7 3c 65 8a 1b d2 ae f0 f7	93 ce 5a 56 a0 a1 f4 f7 3c 65 8a 1b d2 ae f0 f7

Table D.7

D.2 Authentication Protocol

Table D.8 provides test vectors generated during the authentication protocol between T1-R1 and T1-R2. The values provided in the table are as generated or received on the transmitter (T1) side.

	Value in Hex for R1	**Value in Hex for R2**
Authentication and Key Exchange (Without stored k_m)		
r_{tx}	18 fa e4 20 6a fb 51 49	f9 f1 30 a8 2d 5b e5 c3

Receiver ID(Extracted from certificate *cert*$_{rx}$)	74 5b b8 bd 04	8b a4 47 42 fb
RxCaps	02 00 03 (Device is an HDCP Repeater, *RxCaps*.REPEATER is set to one)	02 00 02 (Device is not an HDCP Repeater, *RxCaps*.REPEATER is zero)
TxCaps	02 00 00	02 00 00
Certificate signature verification	Hash: 4d d6 ef 0f 5a dd d4 82 a8 ce bc 62 1d c3 b5 f0 50 f6 39 f1 d3 8b a6 a4 4f ca 58 4b 45 a9 e9 39 Encoded Message EM: 00 01 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff f f ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 30 31 30 0d 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20 4d d6 ef 0f 5a dd d4 82 a8 ce bc 62 1d c3 b5 f0 50 f6 39 f1 d3 8b a6 a4 4f ca 58 4b 45 a9 e9 39	Hash: 08 fe 9c 34 64 c3 ca 9a 76 38 43 b0 61 4f 55 b4 db 7b 9b 3e c2 8f 72 09 04 0e 81 4e 5c 2e 75 bd Encoded Message EM: 00 01 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 30 31 30 0d 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20 08 fe 9c 34 64 c3 ca 9a 76 38 43 b0 61 4f 55 b4 db 7b 9b 3e c2 8f 72 09 04 0e 81 4e 5c 2e 75 bd
kpub$_{rx}$ (Extracted from certificate *cert*$_{rx}$)	n: af b5 c5 c6 7b c5 3a 34 90 a9 54 c0 8f b7 eb a1 54 d2 4f 22 de 83 f5 03 a6 c6 68 46 9b c0 b8 c8 6c db 26 f9 3c 49 2f 02 e1 71 df 4e f3 0e c8 bf 22 9d 04 cf bf a9 0d ff 68 ab 05 6f 1f 12 8a 68 62 eb fe c9 ea 9f	n: e4 68 63 8a da 97 2d de 9a 8d 1c b1 65 4b 85 8d e5 46 d6 db 95 a5 f6 66 74 ea 81 0b 9a 58 58 66 26 86 a6 b4 56 2b 29 43 e5 bb 81 74 86 a7 b7 16 2f 07 ec d1 b5 f9 ae 4f 98 89 a9 91 7d 58 5b 8d 20 d5 c5 08 40 3b 86 af f4 d6 b9

	a7 fb 8c ba b1 bd 65 ac 35 9c a0 33 b1 dd a6 05 36 af 00 a2 7f bc 07 b2 dd b5 cc 57 5c dc c0 95 50 e5 ff 1f 20 db 59 46 fa 47 c4 ed 12 2e 9e 22 bd 95 a9 85 59 a1 59 3c c7 83 e: 01 00 01	20 95 e8 90 3b 8f 9f 36 5b 46 b6 d4 1e f5 05 88 80 14 e7 2c 77 5d 6e 54 e9 65 81 5a 68 92 a5 d6 40 78 11 97 65 d7 64 36 5e 8d 2a 87 a8 eb 7d 06 2c 10 f8 0a 7d e: 01 00 01
k_m	68 bc c5 1b a9 db 1b d0 fa f1 5e 9a d8 a5 af b9	ca 9f 83 95 70 d0 d0 f9 cf e4 eb 54 7e 09 fa 3b
$E_{kpub}(km)$	Seed: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F lhash: e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55 $E_{kpub}(km)$: 9b 9f 80 19 ad 0e a2 f0 dd a0 29 33 d9 6d 1c 77 31 37 57 e0 e5 b2 bd dd 36 3e 38 4e 7d 40 78 66 97 7a 4c ce c5 c7 5d 01 57 26 cc a2 f6 de 34 dd 29 be 5e 31 e8 f 1 34 e8 1a 63 a3 6d 46 dc 0a 06 08 99 9d db 3c a2 9c 04 dd 4e d9 02 7d 20 54 ec ca 86 42 1b 18 da 30 9c c4 cb ac b4 54 de 84 68 71 53 6d 92 17 ca 08 8a 7a f9 98 9a b6 7b 22 92 ac 7d 0d 6b d6 7f 31 ab f0 10 c5 2a 0f 6d 27 a0	Seed: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F lhash: e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55 $E_{kpub}(km)$: a8 55 c2 c4 c6 be ef cd cb 9f e3 9f 2a b7 29 76 fe d8 da c9 38 fa 39 f0 ab ca 8a ed 95 7b 93 b2 df d0 7d 09 9d 05 96 66 03 6e ba e0 63 0f 30 77 c2 bb e2 11 39 e5 27 78 ee 64 f2 85 36 57 c3 39 d2 7b 79 03 b7 cc 82 cb f0 62 82 43 38 09 9b 71 aa 38 a6 3f 48 12 6d 8c 5e 07 90 76 ac 90 99 51 5b 06 a5 fa 50 e4 f9 25 c3 07 12 37 64 92 d7 db d3 34 1c e4 fa dd 09 e6 28 3d 0c ad a9 d8 e1 b5
r_{rx}	3b a0 be de 0c 46 a9 91	e1 7a b0 fd 0f 54 40 52
$dkey_0$	4f 14 8d 11 dd 49 18 10 6f ab 16 6f f6 fd a6 ed	2a 04 d7 eb 0a 0b 4e 20 26 45 84 01 1e ab 0a 4a
$dkey_1$	b5 02 0c 0d f2 81 ba df e4 19 77 fa d0 ac 61 17	f9 dc 18 97 e8 ee d8 f9 ec 6a 5d 34 a9 62 02 c9
k_d	4f 14 8d 11 dd 49 18 10 6f ab 16 6f f6 fd a6 ed b5 02 0c 0d f2 81 ba df e4 19 77 fa d0 ac 61 17	2a 04 d7 eb 0a 0b 4e 20 26 45 84 01 1e ab 0a 4a f9 dc 18 97 e8 ee d8 f9 ec 6a 5d 34 a9 62 02 c9
H	2e f5 ed f8 7f d8 a3 d0 f4 a9 d8 ac 3a d0 b4 56 2e 32 19 11 41 16 f1 ef 0f 02 3d 3a 78 e2 2a c6	82 b8 1a ca ed fc 87 72 7d 17 23 53 cb 81 83 bf db ba fb 90 b2 4e 96 fe ba 6d ad 67 aa 2b 2a 56
H'	2e f5 ed f8 7f d8 a3 d0 f4 a9 d8 ac 3a d0 b4 56 2e 32 19 11 41 16 f1 ef 0f 02 3d 3a 78	82 b8 1a ca ed fc 87 72 7d 17 23 53 cb 81 83 bf db ba fb 90 b2 4e 96 fe ba 6d ad 67 aa 2b 2a 56

	e2 2a c6	
Pairing		
$E_{kh}(k_m)$	Hash of private = SHA256 hash on concatenation of p, q, d mod (p-1), d mod (q-1), q^-1 mod p i.e. SHA-256(p ||q||d mod (p-1)||d mod (q-1)||q^-1 mod p): db e7 c0 f2 32 e8 dd 33 43 00 c3 9b 20 57 7a da 85 86 c7 b6 6d 9f b3 66 a0 76 0c fb c2 ab 4d 34 k_h: 85 86 c7 b6 6d 9f b3 66 a0 76 0c fb c2 ab 4d 34 $E_{kh}(k_m)$: b8 9f f9 72 6a 6f 2c 1e 29 b6 44 8d dc a3 10 bd	Hash of private = SHA256 hash on concatenation of p, q, d mod (p-1), d mod (q-1), q^-1 mod p i.e. SHA-256(p ||q||d mod (p-1)||d mod (q-1)||q^-1 mod p): 8a da 77 4a e0 1b 26 f8 c8 9d e1 f3 23 fd e2 15 c6 aa 14 eb b0 35 4d 50 83 f5 de 74 2a 8c 1b a2 k_h: c6 aa 14 eb b0 35 4d 50 83 f5 de 74 2a 8c 1b a2 $E_{kh}(k_m)$: e6 57 8e bc c7 68 44 87 88 8a 9b d7 d6 ae 38 be
m	18 fa e4 20 6a fb 51 49 3b a0 be de 0c 46 a9 91	f9 f1 30 a8 2d 5b e5 c3 e1 7a b0 fd 0f 54 40 52
Locality Check		
r_n	32 75 3e a8 78 a6 38 1c	a0 fe 9b b8 20 60 58 ca
L	bc 20 92 33 54 91 c1 9e a4 de 8b 30 49 c2 06 6a d8 11 a2 2a b1 46 df 74 58 47 05 a8 b7 67 fb dd	f2 0f 13 6e 85 53 c1 0c d3 dd b2 f9 6d 33 31 f9 cb 6e 97 8c cd 5e da 13 dd ea 41 44 10 9b 51 b0
L'	bc 20 92 33 54 91 c1 9e a4 de 8b 30 49 c2 06 6a d8 11 a2 2a b1 46 df 74 58 47 05 a8 b7 67 fb dd	f2 0f 13 6e 85 53 c1 0c d3 dd b2 f9 6d 33 31 f9 cb 6e 97 8c cd 5e da 13 dd ea 41 44 10 9b 51 b0
Session Key Exchange		
k_s	f3 df 1d d9 57 96 12 3f 98 97 89 b4 21 e1 2d e1	f3 df 1d d9 57 96 12 3f 98 97 89 b4 21 e1 2d e1
r_{iv}	40 2b 6b 43 c5 e8 86 d8	9a 6d 11 00 a9 b7 6f 64
$dkey_2$	bf ed 5a cb 93 28 d4 56 a9 f5 2e 0e f3 36 75 f3	45 54 97 7d 85 5d a8 c0 2a de f8 90 95 02 7d 1a
$E_{dkey}(k_s)$	4c 32 47 12 c4 be c6 69 0a c2 19 64 de 91 f1 83	b6 8b 8a a4 d2 cb ba ff 53 33 c1 d9 bb b7 10 a9
Authentication with Repeaters		
Upstream Propagation of Topology Information		
Receiver ID$_0$	47 8e 71 e2 0f	N/A as R2 is not an HDCP Repeater
Receiver ID$_1$	35 79 6a 17 0e	
Receiver ID$_2$	74 e8 53 97 a2	
Receiver ID list	47 8e 71 e2 0f 35 79 6a 17 0e 74 e8 53 97 a2	
RxInfo *RxInfo* fields Rsvd DEPTH DEVICE_COUNT	02 31 Values in binary 0000_b 001_b 00011_b	

MAX_DEVS_EXCEEDED MAX_CASCADE_EXCEEDED HDCP2_LEGACY_DEVICE_DOWNSTREAM HDCP1_DEVICE_DOWNSTREAM	0_b 0_b 0_b 1_b	
seq_num_V	00 00 00	
V	63 6d c5 08 4d 6c b1 0e 93 a5 28 67 0f 34 1f 88	
V'	bc cc 7d 16 e6 bc b9 02 60 08 1d f7 4a b4 5c 8a	
Downstream Propagation of Content Stream Management Information		
STREAM_ID	00	
Type	01	
seq_num_M	00 00 00	
StreamID_Type‖seq_num_M	00 01 00 00 00	
SHA256(k_d)	1e 6c 5c a4 40 9a 66 a6 20 96 fe cd fc f3 f6 b0 45 e4 44 6b f5 45 c8 45 2b 4a ee 48 0c 53 c4 dd	
M'	dd 26 e9 52 6e 0e 1d 69 c8 84 e4 cc c8 09 aa c7 71 e9 97 b5 61 89 09 6e 4d 94 24 c2 1b 64 58 c6	

Table D.8

Table D.9 provides an HDCP 2 facsimile SRM signed with the facsimile DCP LLC key in Table D.2. The SRM revokes Receiver IDs of receivers R1 and R2

Receiver IDs revoked	74 5b b8 bd 04, 8b a4 47 42 fb
SRM Version	00 01
SRM Value	91 00 00 01 01 00 01 91 00 80 00 00 74 5b b8 bd 04 8b a4 47 42 fb 17 07 e9 ea 61 ad b4 2e 9a 44 a9 1e 44 ba ab 6f 6b 37 27 50 bb 17 8e ad c5 7f e8 f5 21 b4 60 1e 54 80 da 2a 1a 59 f3 9d e3 98 54 43 24 70 ca 83 47 64 2d c6 26 6d 30 05 b4 ee 9b b6 69 a2 f3 7c 7d 13 cf f3 a7 c7 89 ef 50 0d 32 e1 d2 2c d1 b5 46 d6 36 44 25 52 65 06 b6 31 e7 26 d2 5c 1c 17 b7 26 6f 73 e5 8a 6e e9 db 00 27 70 11 bb 75 a0 49 30 ff 38 b3 d8 2d 03 c1 78 50 74 ca 60 aa 32 03 28 8d e4 a1 8c 62 2d f3 61 6c bf a1 9c ab b3 d7 e7 d6 dc 31 f6 74 1a ab c0 9c 6a 0a c2 65 e4 29 bd 4e 22 73 5a 2c c7 75 96 e9 7a 16 a6 70 6b 3c ba 50 60 5a 33 d5 d7 f2 76 5b 5e 2d 45 e2 0e f1 4c 6d 8d cf af 39 23 79 a5 fb ae cb 8f d8 de df db 24 20 52 10 74 ba 42 6f ad bb 3e 3a a5 ce 99 0d ff 41 a6 0f 60 7f ae 05 00 3c 2c f3 ba 5a 86 1b 04 7f 53 e5 e3 68 dc 6b 36 25 73 69 95 5d 15 37 d6 98 d6 6b a8 d0 35 37 2d 2f f1 53 90 aa 32 87 99 3b b7 33 48 0e ce e4 f1 d8 93 51 eb 98 92 e7 2d ac a2 32 fb e1 f9 f1 7d 92 26 37 5d 4c 0c f8 a2 11 4c 2a 49 b6 48 bc cc 44 01 06 cb 74 da 0d 70 fa 06 64 a6 54 e6 7c 65 34 ff 97 ed db 8a 8d 40 be fe 60 e5 0c bf 50 60 7b 16 71 bd ff c3 1b fb 15 a5 10 07 6a 5d 1a 6c fa 67 dc b3 cd a3 85 6e eb 77 5f 92 8f ee fb d5 34 58 72 b1 55
SRM Signature Verification	Hash: 3b 11 c9 ee f0 b6 ec 5b 68 34 b2 67 95 7c 2d 03 1d 83 0a d7 38 78 07 24 c9 14 c6 74 4e f6 70 b0 Encoded Message: 00 01 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

	ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff 00 30 31 30 0d 06 09 60 86 48 01 65 03 04 02 01 05 00 04 20 3b 11 c9 ee f0 b6 ec 5b 68 34 b2 67 95 7c 2d 03 1d 83 0a d7 38 78 07 24 c9 14 c6 74 4e f6 70 b0

Table D.9

D.3 Encryption

Table D.10, D.11 and D.12 provides encryption test vectors generated by transmitter T1 for Receiver R2. The test vectors are generated for 4-lane, 2-lane and 1-lane main link configurations, respectively, in SST mode with the Type input to the HDCP Cipher set to 0x00 (Refer to Section 3.2).

4 Lane, Inter-BS spacing = 15, Inter-SR spacing = 3

	Link Clock	**Stream**	**Stream Type**	**Cipher Key [127:0]**	**Cipher Key Name [127:0]**	**Encrypted Stream**			
						Lane3	**Lane2**	**Lane1**	**Lane0**
	-3	1c1c1c1c	SR	--	--	1c	1c	1c	1c
	-2	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	-1	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	0	1c1c1c1c	SR	--	--	1c	1c	1c	1c
Line 1 encrypted	1	39393939	VB-ID	69cfe2f4	key0_w0	50	f6	db	cd
	2	00000000	Mvid	af464ed8	key0_w1	af	46	4e	d8
	3	00000000	Maud	29d2b86d	key0_w2	29	d2	b8	6d
	4	00000000	null stream	218b93a3	key0_w3	21	8b	93	a3
	5	00000000	null stream	3b643a4f	key1_w0	3b	64	3a	4f
	6	00000000	null stream	61d2ec7b	key1_w1	61	d2	ec	7b
	7	00000000	null stream	245b7198	key1_w2	24	5b	71	98
	8	00000000	null stream	d0fe7230	key1_w3	d0	fe	72	30
	9	00000000	null stream	c5caaccd	key2_w0	c5	ca	ac	cd
	10	00000000	null stream	a339cf1b	key2_w1	a3	39	cf	1b
	11	00000000	null stream	16cc1d5c	key2_w2	16	cc	1d	5c
	12	bcbcbcbc	BS	4bf3f06c	key2_w3	bc	bc	bc	bc

	13	3c3c3c3c	CP	83fffdd6	key3_w0	3c	3c	3c	3c
	14	3c3c3c3c	CP	a9fbd4c0	key3_w1	3c	3c	3c	3c
	15	bcbcbcbc	BS	b8b6ab60	key3_w2	bc	bc	bc	bc
Line 2 encrypted	16	39393939	VB-ID	85046ba5	key3_w3	bc	3d	52	9c
	17	00000000	Mvid	a93d210c	key4_w0	a9	3d	21	0c
	18	00000000	Maud	18081647	key4_w1	18	08	16	47
	19	00000000	null stream	c3aed66b	key4_w2	c3	ae	d6	6b
	20	00000000	null stream	7150f614	key4_w3	71	50	f6	14
	21	00000000	null stream	5493ec02	key5_w0	54	93	ec	02
	22	00000000	null stream	75a45c61	key5_w1	75	a4	5c	61
	23	00000000	null stream	4d0a6492	key5_w2	4d	0a	64	92
	24	00000000	null stream	70875305	key5_w3	70	87	53	05
	25	00000000	null stream	bc0ae724	key6_w0	bc	0a	e7	24
	26	00000000	null stream	19e1c0de	key6_w1	19	e1	c0	de
	27	bcbcbcbc	BS	4eacadd7	key6_w2	bc	bc	bc	bc
	28	3c3c3c3c	CP	576649cf	key6_w3	3c	3c	3c	3c
	29	3c3c3c3c	CP	f4493ca1	key7_w0	3c	3c	3c	3c
	30	bcbcbcbc	BS	a1b39698	key7_w1	bc	bc	bc	bc
Line 3 encrypted	31	39393939	VB-ID	8139ccb8	key7_w2	b8	0	f5	81
	32	00000000	Mvid	1cf0557a	key7_w3	1c	f0	55	7a
	33	00000000	Maud	f7fb2377	key8_w0	f7	fb	23	77
	34	00000000	null stream	3a2584a6	key8_w1	3a	25	84	a6
	35	00000000	null stream	2e2e8d7a	key8_w2	2e	2e	8d	7a
	36	00000000	null stream	807c95ce	key8_w3	80	7c	95	ce
	37	00000000	null stream	8bfe8444	key9_w0	8b	fe	84	44
	38	00000000	null stream	e9113fdb	key9_w1	e9	11	3f	db
	39	00000000	null stream	bcd89d52	key9_w2	bc	d8	9d	52
	40	00000000	null stream	4d18cf3b	key9_w3	4d	18	cf	3b
	41	00000000	null stream	aad0da4b	key10_w0	aa	d0	da	4b
	42	1c1c1c1c	SR	f1732414	key10_w1	1c	1c	1c	1c
	43	7c7c7c7c	BF	8dcca69a	key10_w2	7c	7c	7c	7c
	44	7c7c7c7c	BF	a9ebf689	key10_w3	7c	7c	7c	7c
	45	1c1c1c1c	SR	9dfdad56	key11_w0	1c	1c	1c	1c
Line 4 unencrypted	46	19191919	VB-ID	--	--	19	19	19	19
	47	00000000	Mvid	--	--	00	00	00	00

	48	00000000	Maud	--	--	00	00	00	00
	49	00000000	null stream	--	--	00	00	00	00
	50	00000000	null stream	--	--	00	00	00	00
	51	00000000	null stream	--	--	00	00	00	00
	52	00000000	null stream	--	--	00	00	00	00
	53	00000000	null stream	--	--	00	00	00	00
	54	00000000	null stream	--	--	00	00	00	00
	55	00000000	null stream	--	--	00	00	00	00
	56	00000000	null stream	--	--	00	00	00	00
	57	bcbcbcbc	BS	--	--	bc	bc	bc	bc
	58	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	59	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	60	bcbcbcbc	BS	--	--	bc	bc	bc	bc
Line 5 unencrypted	61	19191919	VB-ID	--	--	19	19	19	19
	62	00000000	Mvid	--	--	00	00	00	00
	63	00000000	Maud	--	--	00	00	00	00
	64	00000000	null stream	--	--	00	00	00	00
	65	00000000	null stream	--	--	00	00	00	00
	66	00000000	null stream	--	--	00	00	00	00
	67	00000000	null stream	--	--	00	00	00	00
	68	00000000	null stream	--	--	00	00	00	00
	69	00000000	null stream	--	--	00	00	00	00
	70	00000000	null stream	--	--	00	00	00	00
	71	00000000	null stream	--	--	00	00	00	00
	72	bcbcbcbc	BS	--	--	bc	bc	bc	bc
	73	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	74	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	75	bcbcbcbc	BS	--	--	bc	bc	bc	bc
Line 6 unencrypted	76	19191919	VB-ID	--	--	19	19	19	19
	77	00000000	Mvid	--	--	00	00	00	00
	78	00000000	Maud	--	--	00	00	00	00
	79	00000000	null stream	--	--	00	00	00	00
	80	00000000	null stream	--	--	00	00	00	00
	81	00000000	null	--	--	00	00	00	00

			stream						
	82	00000000	null stream	--	--	00	00	00	00
	83	00000000	null stream	--	--	00	00	00	00
	84	00000000	null stream	--	--	00	00	00	00
	85	00000000	null stream	--	--	00	00	00	00
	86	00000000	null stream	--	--	00	00	00	00
	87	1c1c1c1c	SR	--	--	1c	1c	1c	1c
	88	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	89	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	90	1c1c1c1c	SR	--	--	1c	1c	1c	1c
Line 7 encrypted	91	39393939	VB-ID	a848e938	key11_w1	91	71	d0	01
	92	00000000	Mvid	7dac8830	key11_w2	7d	ac	88	30
	93	00000000	Maud	ed4fbedb	key11_w3	ed	4f	be	db
	94	00000000	null stream	b6c43464	key12_w0	b6	c4	34	64
	95	00000000	null stream	85efb23a	key12_w1	85	ef	b2	3a
	96	00000000	null stream	e921d3a8	key12_w2	e9	21	d3	a8
	97	00000000	null stream	aa67e16b	key12_w3	aa	67	e1	6b
	98	00000000	null stream	0d49ff31	key13_w0	0d	49	ff	31

Table D.10

	2 Lane, Inter-BS spacing = 18						
	Link Clock	**Stream**	**Stream Type**	**Cipher Key [127:0]**	**Cipher Key Name [127:0]**	**Encrypted Stream**	
						Lane1	**Lane0**
	-3	1c1c	SR	--	--	1c	1c
	-2	3c3c	CP	--	--	3c	3c
	-2	3c3c	CP	--	--	3c	3c
	0	1c1c	SR	--	--	1c	1c
Line 1 encrypted	1	3939	VB-ID	e2f4	key0_w0_0	db	cd
	2	0000	Mvid	69cf	key0_w0_1	69	cf
	3	0000	Maud	4ed8	key0_w1_0	4e	d8
	4	3939	VB-ID	af46	key0_w1_1	96	7f
	5	0000	Mvid	b86d	key0_w2_0	b8	6d
	6	0000	Maud	29d2	key0_w2_1	29	d2
	7	0000	null stream	93a3	key0_w3_0	93	a3
	8	0000	null stream	218b	key0_w3_1	21	8b
	9	0000	null stream	3a4f	key1_w0_0	3a	4f
	10	0000	null stream	3b64	key1_w0_1	3b	64
	11	0000	null stream	ec7b	key1_w1_0	ec	7b
	12	0000	null stream	61d2	key1_w1_1	61	d2
	13	0000	null stream	7198	key1_w2_0	71	98
	14	0000	null stream	245b	key1_w2_1	24	5b
	15	bcbc	BS	7230	key1_w3_0	bc	bc
	16	3c3c	CP	d0fe	key1_w3_1	3c	3c
	17	3c3c	CP	accd	key2_w0_0	3c	3c
	18	bcbc	BS	c5ca	key2_w0_1	bc	bc
Line 2 encrypted	19	3939	VB-ID	cf1b	key2_w1_0	f6	22
	20	0000	Mvid	a339	key2_w1_1	a3	39
	21	0000	Maud	1d5c	key2_w2_0	1d	5c
	22	3939	VB-ID	16cc	key2_w2_1	2f	f5
	23	0000	Mvid	f06c	key2_w3_0	f0	6c
	24	0000	Maud	4bf3	key2_w3_1	4b	f3
	25	0000	null stream	fdd6	key3_w0_0	fd	d6
	26	0000	null stream	83ff	key3_w0_1	83	ff
	27	0000	null stream	d4c0	key3_w1_0	d4	c0
	28	0000	null stream	a9fb	key3_w1_1	a9	fb
	29	0000	null stream	ab60	key3_w2_0	ab	60
	30	0000	null stream	b8b6	key3_w2_1	b8	b6
	31	0000	null stream	6ba5	key3_w3_0	6b	a5
	32	0000	null stream	8504	key3_w3_1	85	04
	33	bcbc	BS	210c	key4_w0_0	bc	bc
	34	3c3c	CP	a93d	key4_w0_1	3c	3c
	35	3c3c	CP	1647	key4_w1_0	3c	3c

	36	bcbc	BS	1808	key4_w1_1	bc	bc
Line 3 encrypted	37	3939	VB-ID	d66b	key4_w2_0	ef	52
	38	0000	Mvid	c3ae	key4_w2_1	c3	ae
	39	0000	Maud	f614	key4_w3_0	f6	14
	40	3939	VB-ID	7150	key4_w3_1	48	69
	41	0000	Mvid	ec02	key5_w0_0	ec	02
	42	0000	Maud	5493	key5_w0_1	54	93
	43	0000	null stream	5c61	key5_w1_0	5c	61
	44	0000	null stream	75a4	key5_w1_1	75	a4
	45	0000	null stream	6492	key5_w2_0	64	92
	46	0000	null stream	4d0a	key5_w2_1	4d	0a
	47	0000	null stream	5305	key5_w3_0	53	05
	48	0000	null stream	7087	key5_w3_1	70	87
	49	0000	null stream	e724	key6_w0_0	e7	24
	50	0000	null stream	bc0a	key6_w0_1	bc	0a
	51	1c1c	SR	c0de	key6_w1_0	1c	1c
	52	7c7c	BF	19e1	key6_w1_1	7c	7c
	53	7c7c	BF	add7	key6_w2_0	7c	7c
	54	1c1c	SR	4eac	key6_w2_1	1c	1c
Line 4 unencrypted	55	1919	VB-ID	--	--	19	19
	56	0000	Mvid	--	--	00	00
	57	0000	Maud	--	--	00	00
	58	1919	VB-ID	--	--	19	19
	59	0000	Mvid	--	--	00	00
	60	0000	Maud	--	--	00	00
	61	0000	null stream	--	--	00	00
	62	0000	null stream	--	--	00	00
	63	0000	null stream	--	--	00	00
	64	0000	null stream	--	--	00	00
	65	0000	null stream	--	--	00	00
	66	0000	null stream	--	--	00	00
	67	0000	null stream	--	--	00	00
	68	0000	null stream	--	--	00	00
	69	bcbc	BS	--	--	bc	bc
	70	7c7c	BF	--	--	7c	7c
	71	7c7c	BF	--	--	7c	7c
	72	bcbc	BS	--	--	bc	bc
Line 5 unencrypted	73	1919	VB-ID	--	--	19	19
	74	0000	Mvid	--	--	00	00
	75	0000	Maud	--	--	00	00
	76	1919	VB-ID	--	--	19	19
	77	0000	Mvid	--	--	00	00
	78	0000	Maud	--	--	00	00
	79	0000	null stream	--	--	00	00
	80	0000	null stream	--	--	00	00
	81	0000	null stream	--	--	00	00
	82	0000	null stream	--	--	00	00

	83	0000	null stream	--	--	00	00
	84	0000	null stream	--	--	00	00
	85	0000	null stream	--	--	00	00
	86	0000	null stream	--	--	00	00
	87	bcbc	BS	--	--	bc	bc
	88	7c7c	BF	--	--	7c	7c
	89	7c7c	BF	--	--	7c	7c
	90	bcbc	BS	--	--	bc	bc
Line 6 unencrypted	91	1919	VB-ID	--	--	19	19
	92	0000	Mvid	--	--	00	00
	93	0000	Maud	--	--	00	00
	94	1919	VB-ID	--	--	19	19
	95	0000	Mvid	--	--	00	00
	96	0000	Maud	--	--	00	00
	97	0000	null stream	--	--	00	00
	98	0000	null stream	--	--	00	00
	99	0000	null stream	--	--	00	00
	100	0000	null stream	--	--	00	00
	101	0000	null stream	--	--	00	00
	102	0000	null stream	--	--	00	00
	103	0000	null stream	--	--	00	00
	104	0000	null stream	--	--	00	00
	105	1c1c	SR	--	--	1c	1c
	106	3c3c	CP	--	--	3c	3c
	107	3c3c	CP	--	--	3c	3c
	108	1c1c	SR	--	--	1c	1c
Line 7 encrypted	109	3939	VB-ID	49cf	key6_w3_0	70	f6
	110	0000	Mvid	5766	key6_w3_1	57	66
	111	0000	Maud	3ca1	key7_w0_0	3c	a1
	112	3939	VB-ID	f449	key7_w0_1	cd	70
	113	0000	Mvid	9698	key7_w1_0	96	98
	114	0000	Maud	a1b3	key7_w1_1	a1	b3
	115	0000	null stream	ccb8	key7_w2_0	cc	b8
	116	0000	null stream	8139	key7_w2_1	81	39
	117	0000	null stream	557a	key7_w3_0	55	7a
	118	0000	null stream	1cf0	key7_w3_1	1c	f0
	119	0000	null stream	2377	key8_w0_0	23	77

Table D.11

	1 Lane, Inter-BS spacing = 24					
	Link Clock	**Stream**	**Stream Type**	**Cipher Key [127:0]**	**Cipher Key Name [127:0]**	**Encrypted Stream**
						Lane0
	-3	1c	SR	--	--	1c
	-2	3c	CP	--	--	3c
	-2	3c	CP	--	--	3c
	0	1c	SR	--	--	1c
Line 1 encrypted	1	39	VB-ID	f4	key0_b0	cd
	2	00	Mvid	e2	key0_b1	e2
	3	00	Maud	cf	key0_b2	cf
	4	39	VB-ID	69	key0_b3	50
	5	00	Mvid	d8	key0_b4	d8
	6	00	Maud	4e	key0_b5	4e
	7	39	VB-ID	46	key0_b6	7f
	8	00	Mvid	af	key0_b7	af
	9	00	Maud	6d	key0_b8	6d
	10	39	VB-ID	b8	key0_b9	81
	11	00	Mvid	d2	key0_b10	d2
	12	00	Maud	29	key0_b11	29
	13	00	null stream	a3	key0_b12	a3
	14	00	null stream	93	key0_b13	93
	15	00	null stream	8b	key0_b14	8b
	16	00	null stream	21	key0_b15	21
	17	00	null stream	4f	key1_b0	4f
	18	00	null stream	3a	key1_b1	3a
	19	00	null stream	64	key1_b2	64
	20	00	null stream	3b	key1_b3	3b
	21	bc	BS	7b	key1_b4	bc
	22	3c	CP	ec	key1_b5	3c
	23	3c	CP	d2	key1_b6	3c
	24	bc	BS	61	key1_b7	bc
Line 2 encrypted	25	39	VB-ID	98	key1_b8	a1
	26	00	Mvid	71	key1_b9	71
	27	00	Maud	5b	key1_b10	5b
	28	39	VB-ID	24	key1_b11	1d
	29	00	Mvid	30	key1_b12	30
	30	00	Maud	72	key1_b13	72
	31	39	VB-ID	fe	key1_b14	c7
	32	00	Mvid	d0	key1_b15	d0
	33	00	Maud	cd	key2_b0	cd
	34	39	VB-ID	ac	key2_b1	95
	35	00	Mvid	ca	key2_b2	ca

	36	00	Maud	c5	key2_b3	c5
	37	00	null stream	1b	key2_b4	1b
	38	00	null stream	cf	key2_b5	cf
	39	00	null stream	39	key2_b6	39
	40	00	null stream	a3	key2_b7	a3
	41	00	null stream	5c	key2_b8	5c
	42	00	null stream	1d	key2_b9	1d
	43	00	null stream	cc	key2_b10	cc
	44	00	null stream	16	key2_b11	16
	45	bc	BS	6c	key2_b12	bc
	46	3c	CP	f0	key2_b13	3c
	47	3c	CP	f3	key2_b14	3c
	48	bc	BS	4b	key2_b15	bc
Line 3 encrypted	49	39	VB-ID	d6	key3_b0	ef
	50	00	Mvid	fd	key3_b1	fd
	51	00	Maud	ff	key3_b2	ff
	52	39	VB-ID	83	key3_b3	ba
	53	00	Mvid	c0	key3_b4	c0
	54	00	Maud	d4	key3_b5	d4
	55	39	VB-ID	fb	key3_b6	c2
	56	00	Mvid	a9	key3_b7	a9
	57	00	Maud	60	key3_b8	60
	58	39	VB-ID	ab	key3_b9	92
	59	00	Mvid	b6	key3_b10	b6
	60	00	Maud	b8	key3_b11	b8
	61	00	null stream	a5	key3_b12	a5
	62	00	null stream	6b	key3_b13	6b
	63	00	null stream	04	key3_b14	04
	64	00	null stream	85	key3_b15	85
	65	00	null stream	0c	key4_b0	0c
	66	00	null stream	21	key4_b1	21
	67	00	null stream	3d	key4_b2	3d
	68	00	null stream	a9	key4_b3	a9
	69	1c	SR	47	key4_b4	47
	70	7c	BF	16	key4_b5	16
	71	7c	BF	08	key4_b6	08
	72	1c	SR	18	key4_b7	18
Line 4 unencrypted	73	19	VB-ID	--	--	19
	74	00	Mvid	--	--	00
	75	00	Maud	--	--	00
	76	19	VB-ID	--	--	19
	77	00	Mvid	--	--	00
	78	00	Maud	--	--	00
	79	19	VB-ID	--	--	19
	80	00	Mvid	--	--	00
	81	00	Maud	--	--	00
	82	19	VB-ID	--	--	19

	83	00	Mvid	--	--	00
	84	00	Maud	--	--	00
	85	00	null stream	--	--	00
	86	00	null stream	--	--	00
	87	00	null stream	--	--	00
	88	00	null stream	--	--	00
	89	00	null stream	--	--	00
	90	00	null stream	--	--	00
	91	00	null stream	--	--	00
	92	00	null stream	--	--	00
	93	bc	BS	--	--	bc
	94	7c	BF	--	--	7c
	95	7c	BF	--	--	7c
	96	bc	BS	--	--	bc
Line 5 unencrypted	97	19	VB-ID	--	--	19
	98	00	Mvid	--	--	00
	99	00	Maud	--	--	00
	100	19	VB-ID	--	--	19
	101	00	Mvid	--	--	00
	102	00	Maud	--	--	00
	103	19	VB-ID	--	--	19
	104	00	Mvid	--	--	00
	105	00	Maud	--	--	00
	106	19	VB-ID	--	--	19
	107	00	Mvid	--	--	00
	108	00	Maud	--	--	00
	109	00	null stream	--	--	00
	110	00	null stream	--	--	00
	111	00	null stream	--	--	00
	112	00	null stream	--	--	00
	113	00	null stream	--	--	00
	114	00	null stream	--	--	00
	115	00	null stream	--	--	00
	116	00	null stream	--	--	00
	117	bc	BS	--	--	bc
	118	7c	BF	--	--	7c
	119	7c	BF	--	--	7c
	120	bc	BS	--	--	bc
Line 6 unencrypted	121	19	VB-ID	--	--	19
	122	00	Mvid	--	--	00
	123	00	Maud	--	--	00
	124	19	VB-ID	--	--	19
	125	00	Mvid	--	--	00
	126	00	Maud	--	--	00
	127	19	VB-ID	--	--	19
	128	00	Mvid	--	--	00
	129	00	Maud	--	--	00

	130	19	VB-ID	--	--	19
	131	00	Mvid	--	--	00
	132	00	Maud	--	--	00
	133	00	null stream	--	--	00
	134	00	null stream	--	--	00
	135	00	null stream	--	--	00
	136	00	null stream	--	--	00
	137	00	null stream	--	--	00
	138	00	null stream	--	--	00
	139	00	null stream	--	--	00
	140	00	null stream	--	--	00
	141	1c	SR	--	--	1c
	142	3c	CP	--	--	3c
	143	3c	CP	--	--	3c
	144	1c	SR	--	--	1c
Line 7 encrypted	145	39	VB-ID	6b	key4_b8	52
	146	00	Mvid	d6	key4_b9	d6
	147	00	Maud	ae	key4_b10	ae
	148	39	VB-ID	c3	key4_b11	fa
	149	00	Mvid	14	key4_b12	14
	150	00	Maud	f6	key4_b13	f6
	151	39	VB-ID	50	key4_b14	69
	152	00	Mvid	71	key4_b15	71
	153	00	Maud	02	key5_b0	02
	154	39	VB-ID	ec	key5_b1	d5
	155	00	Mvid	93	key5_b2	93
	156	00	Maud	54	key5_b3	54
	157	00	null stream	61	key5_b4	61
	158	00	null stream	5c	key5_b5	5c
	159	00	null stream	a4	key5_b6	a4
	160	00	null stream	75	key5_b7	75
	161	00	null stream	92	key5_b8	92

Table D.12

Table D.13, D.14 and D.15 provides encryption test vectors generated by transmitter T1 for Receiver R2. The test vectors are generated for 4-lane, 2-lane and 1-lane main link configurations, respectively, in SST mode with the Type input to the HDCP Cipher set to 0x01 (Refer to Section 3.2).

4 Lane, Inter-BS spacing = 15, Inter-SR spacing = 3									
	Link Clock	**Stream**	**Stream Type**	**Cipher Key [127:0]**	**Cipher Key Name [127:0]**	**Encrypted Stream**			
						Lane3	**Lane2**	**Lane1**	**Lane0**
	-3	1c1c1c1c	SR	--	--	1c	1c	1c	1c
	-2	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	-1	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	0	1c1c1c1c	SR	--	--	1c	1c	1c	1c
Line 1 encrypted	1	39393939	VB-ID	f2d19ee4	key0_w0	cb	e8	a7	dd
	2	00000000	Mvid	8d97b2da	key0_w1	8d	97	b2	da
	3	00000000	Maud	b1705349	key0_w2	b1	70	53	49
	4	00000000	null stream	214c1eb0	key0_w3	21	4c	1e	b0
	5	00000000	null stream	f2d3c2c0	key1_w0	f2	d3	c2	c0
	6	00000000	null stream	87f39b60	key1_w1	87	f3	9b	60
	7	00000000	null stream	fe935293	key1_w2	fe	93	52	93
	8	00000000	null stream	6dda674c	key1_w3	6d	da	67	4c
	9	00000000	null stream	fab7598d	key2_w0	fa	b7	59	8d
	10	00000000	null stream	b7083eb3	key2_w1	b7	08	3e	b3
	11	00000000	null stream	f729a5ee	key2_w2	f7	29	a5	ee
	12	bcbcbcbc	BS	1e436134	key2_w3	bc	bc	bc	bc
	13	3c3c3c3c	CP	643aee5e	key3_w0	3c	3c	3c	3c
	14	3c3c3c3c	CP	2fca1d7c	key3_w1	3c	3c	3c	3c
	15	bcbcbcbc	BS	bdd265f4	key3_w2	bc	bc	bc	bc
Line 2 encrypted	16	39393939	VB-ID	9877cbf9	key3_w3	a1	4e	f2	c0
	17	00000000	Mvid	cf16b1aa	key4_w0	cf	16	b1	aa
	18	00000000	Maud	cd18ba83	key4_w1	cd	18	ba	83
	19	00000000	null stream	461f352b	key4_w2	46	1f	35	2b
	20	00000000	null stream	ee18d5ad	key4_w3	ee	18	d5	ad
	21	00000000	null stream	a0585757	key5_w0	a0	58	57	57
	22	00000000	null stream	c55ffc47	key5_w1	c5	5f	fc	47
	23	00000000	null stream	b9d95384	key5_w2	b9	d9	53	84
	24	00000000	null stream	02ebaf35	key5_w3	02	eb	af	35
	25	00000000	null stream	e85d8e5d	key6_w0	e8	5d	8e	5d
	26	00000000	null stream	417aacb3	key6_w1	41	7a	ac	b3
	27	bcbcbcbc	BS	6820b5f7	key6_w2	bc	bc	bc	bc
	28	3c3c3c3c	CP	7933aef5	key6_w3	3c	3c	3c	3c
	29	3c3c3c3c	CP	989e0649	key7_w0	3c	3c	3c	3c
	30	bcbcbcbc	BS	9b1aee75	key7_w1	bc	bc	bc	bc
Line 3 encrypted	31	39393939	VB-ID	cd09c7fa	key7_w2	f4	30	fe	c3
	32	00000000	Mvid	d1994b82	key7_w3	d1	99	4b	82
	33	00000000	Maud	080eeadc	key8_w0	08	0e	ea	dc
	34	00000000	null stream	7c6eb859	key8_w1	7c	6e	b8	59

	35	00000000	null stream	8b223dcf	key8_w2	8b	22	3d	cf
	36	00000000	null stream	9567f928	key8_w3	95	67	f9	28
	37	00000000	null stream	c6eed14d	key9_w0	c6	ee	d1	4d
	38	00000000	null stream	237450000	key9_w1	02	37	45	e4
	39	00000000	null stream	b33d3067	key9_w2	b3	3d	30	67
	40	00000000	null stream	5b9749ef	key9_w3	5b	97	49	ef
	41	00000000	null stream	eb1789c3	key10_w0	eb	17	89	c3
	42	1c1c1c1c	SR	3460077d	key10_w1	1c	1c	1c	1c
	43	7c7c7c7c	BF	ce5be2c6	key10_w2	7c	7c	7c	7c
	44	7c7c7c7c	BF	da8f478a	key10_w3	7c	7c	7c	7c
	45	1c1c1c1c	SR	33f82ab2	key11_w0	1c	1c	1c	1c
Line 4 unencrypted	46	19191919	VB-ID	--	--	19	19	19	19
	47	00000000	Mvid	--	--	00	00	00	00
	48	00000000	Maud	--	--	00	00	00	00
	49	00000000	null stream	--	--	00	00	00	00
	50	00000000	null stream	--	--	00	00	00	00
	51	00000000	null stream	--	--	00	00	00	00
	52	00000000	null stream	--	--	00	00	00	00
	53	00000000	null stream	--	--	00	00	00	00
	54	00000000	null stream	--	--	00	00	00	00
	55	00000000	null stream	--	--	00	00	00	00
	56	00000000	null stream	--	--	00	00	00	00
	57	bcbcbcbc	BS	--	--	bc	bc	bc	bc
	58	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	59	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	60	bcbcbcbc	BS	--	--	bc	bc	bc	bc
Line 5 unencrypted	61	19191919	VB-ID	--	--	19	19	19	19
	62	00000000	Mvid	--	--	00	00	00	00
	63	00000000	Maud	--	--	00	00	00	00
	64	00000000	null stream	--	--	00	00	00	00
	65	00000000	null stream	--	--	00	00	00	00
	66	00000000	null stream	--	--	00	00	00	00
	67	00000000	null stream	--	--	00	00	00	00
	68	00000000	null stream	--	--	00	00	00	00
	69	00000000	null stream	--	--	00	00	00	00
	70	00000000	null stream	--	--	00	00	00	00
	71	00000000	null stream	--	--	00	00	00	00
	72	bcbcbcbc	BS	--	--	bc	bc	bc	bc
	73	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	74	7c7c7c7c	BF	--	--	7c	7c	7c	7c
	75	bcbcbcbc	BS	--	--	bc	bc	bc	bc
Line 6 unencrypted	76	19191919	VB-ID	--	--	19	19	19	19
	77	00000000	Mvid	--	--	00	00	00	00
	78	00000000	Maud	--	--	00	00	00	00
	79	00000000	null stream	--	--	00	00	00	00
	80	00000000	null stream	--	--	00	00	00	00
	81	00000000	null stream	--	--	00	00	00	00

	82	00000000	null stream	--	--	00	00	00	00
	83	00000000	null stream	--	--	00	00	00	00
	84	00000000	null stream	--	--	00	00	00	00
	85	00000000	null stream	--	--	00	00	00	00
	86	00000000	null stream	--	--	00	00	00	00
	87	1c1c1c1c	SR	--	--	1c	1c	1c	1c
	88	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	89	3c3c3c3c	CP	--	--	3c	3c	3c	3c
	90	1c1c1c1c	SR	--	--	1c	1c	1c	1c
Line 7 encrypted	91	39393939	VB-ID	4b6aed30	key11_w1	72	53	d4	09
	92	00000000	Mvid	371d00a2	key11_w2	37	1d	00	a2
	93	00000000	Maud	ac4f116c	key11_w3	ac	4f	11	6c
	94	00000000	null stream	67175daa	key12_w0	67	17	5d	aa
	95	00000000	null stream	4b430585	key12_w1	4b	43	05	85
	96	00000000	null stream	839c59f6	key12_w2	83	9c	59	f6
	97	00000000	null stream	4b562ac6	key12_w3	4b	56	2a	c6
	98	00000000	null stream	965ef150	key13_w0	96	5e	f1	50

Table D.13

2 Lane, Inter-BS spacing = 18							
	Link Clock	**Stream**	**Stream Type**	**Cipher Key [127:0]**	**Cipher Key Name [127:0]**	**Encrypted Stream**	
						Lane1	**Lane0**
	-3	1c1c	SR	--	--	1c	1c
	-2	3c3c	CP	--	--	3c	3c
	-2	3c3c	CP	--	--	3c	3c
	0	1c1c	SR	--	--	1c	1c
Line 1 encrypted	1	3939	VB-ID	9ee4	key0_w0_0	a7	dd
	2	0000	Mvid	f2d1	key0_w0_1	f2	d1
	3	0000	Maud	b2da	key0_w1_0	b2	da
	4	3939	VB-ID	8d97	key0_w1_1	b4	ae
	5	0000	Mvid	5349	key0_w2_0	53	49
	6	0000	Maud	b170	key0_w2_1	b1	70
	7	0000	null stream	1eb0	key0_w3_0	1e	b0
	8	0000	null stream	214c	key0_w3_1	21	4c
	9	0000	null stream	c2c0	key1_w0_0	c2	c0
	10	0000	null stream	f2d3	key1_w0_1	f2	d3
	11	0000	null stream	9b60	key1_w1_0	9b	60
	12	0000	null stream	87f3	key1_w1_1	87	f3
	13	0000	null stream	5293	key1_w2_0	52	93
	14	0000	null stream	fe93	key1_w2_1	fe	93
	15	bcbc	BS	674c	key1_w3_0	bc	bc
	16	3c3c	CP	6dda	key1_w3_1	3c	3c
	17	3c3c	CP	598d	key2_w0_0	3c	3c
	18	bcbc	BS	fab7	key2_w0_1	bc	bc

Line 2 encrypted	19	3939	VB-ID	3eb3	key2_w1_0	07	8a
	20	0000	Mvid	b708	key2_w1_1	b7	08
	21	0000	Maud	a5ee	key2_w2_0	a5	ee
	22	3939	VB-ID	f729	key2_w2_1	ce	10
	23	0000	Mvid	6134	key2_w3_0	61	34
	24	0000	Maud	1e43	key2_w3_1	1e	43
	25	0000	null stream	ee5e	key3_w0_0	ee	5e
	26	0000	null stream	643a	key3_w0_1	64	3a
	27	0000	null stream	1d7c	key3_w1_0	1d	7c
	28	0000	null stream	2fca	key3_w1_1	2f	ca
	29	0000	null stream	65f4	key3_w2_0	65	f4
	30	0000	null stream	bdd2	key3_w2_1	bd	d2
	31	0000	null stream	cbf9	key3_w3_0	cb	f9
	32	0000	null stream	9877	key3_w3_1	98	77
	33	bcbc	BS	b1aa	key4_w0_0	bc	bc
	34	3c3c	CP	cf16	key4_w0_1	3c	3c
	35	3c3c	CP	ba83	key4_w1_0	3c	3c
	36	bcbc	BS	cd18	key4_w1_1	bc	bc
Line 3 encrypted	37	3939	VB-ID	352b	key4_w2_0	0c	12
	38	0000	Mvid	461f	key4_w2_1	46	1f
	39	0000	Maud	d5ad	key4_w3_0	d5	ad
	40	3939	VB-ID	ee18	key4_w3_1	d7	21
	41	0000	Mvid	5757	key5_w0_0	57	57
	42	0000	Maud	a058	key5_w0_1	a0	58
	43	0000	null stream	fc47	key5_w1_0	fc	47
	44	0000	null stream	c55f	key5_w1_1	c5	5f
	45	0000	null stream	5384	key5_w2_0	53	84
	46	0000	null stream	b9d9	key5_w2_1	b9	d9
	47	0000	null stream	af35	key5_w3_0	af	35
	48	0000	null stream	02eb	key5_w3_1	02	eb
	49	0000	null stream	8e5d	key6_w0_0	8e	5d
	50	0000	null stream	e85d	key6_w0_1	e8	5d
	51	1c1c	SR	acb3	key6_w1_0	1c	1c
	52	7c7c	BF	417a	key6_w1_1	7c	7c
	53	7c7c	BF	b5f7	key6_w2_0	7c	7c
	54	1c1c	SR	6820	key6_w2_1	1c	1c
Line 4 unencrypted	55	1919	VB-ID	--	--	19	19
	56	0000	Mvid	--	--	00	00
	57	0000	Maud	--	--	00	00
	58	1919	VB-ID	--	--	19	19
	59	0000	Mvid	--	--	00	00
	60	0000	Maud	--	--	00	00
	61	0000	null stream	--	--	00	00
	62	0000	null stream	--	--	00	00
	63	0000	null stream	--	--	00	00
	64	0000	null stream	--	--	00	00
	65	0000	null stream	--	--	00	00

	66	0000	null stream	--	--	00	00
	67	0000	null stream	--	--	00	00
	68	0000	null stream	--	--	00	00
	69	bcbc	BS	--	--	bc	bc
	70	7c7c	BF	--	--	7c	7c
	71	7c7c	BF	--	--	7c	7c
	72	bcbc	BS	--	--	bc	bc
Line 5 unencrypted	73	1919	VB-ID	--	--	19	19
	74	0000	Mvid	--	--	00	00
	75	0000	Maud	--	--	00	00
	76	1919	VB-ID	--	--	19	19
	77	0000	Mvid	--	--	00	00
	78	0000	Maud	--	--	00	00
	79	0000	null stream	--	--	00	00
	80	0000	null stream	--	--	00	00
	81	0000	null stream	--	--	00	00
	82	0000	null stream	--	--	00	00
	83	0000	null stream	--	--	00	00
	84	0000	null stream	--	--	00	00
	85	0000	null stream	--	--	00	00
	86	0000	null stream	--	--	00	00
	87	bcbc	BS	--	--	bc	bc
	88	7c7c	BF	--	--	7c	7c
	89	7c7c	BF	--	--	7c	7c
	90	bcbc	BS	--	--	bc	bc
Line 6 unencrypted	91	1919	VB-ID	--	--	19	19
	92	0000	Mvid	--	--	00	00
	93	0000	Maud	--	--	00	00
	94	1919	VB-ID	--	--	19	19
	95	0000	Mvid	--	--	00	00
	96	0000	Maud	--	--	00	00
	97	0000	null stream	--	--	00	00
	98	0000	null stream	--	--	00	00
	99	0000	null stream	--	--	00	00
	100	0000	null stream	--	--	00	00
	101	0000	null stream	--	--	00	00
	102	0000	null stream	--	--	00	00
	103	0000	null stream	--	--	00	00
	104	0000	null stream	--	--	00	00
	105	1c1c	SR	--	--	1c	1c
	106	3c3c	CP	--	--	3c	3c
	107	3c3c	CP	--	--	3c	3c
	108	1c1c	SR	--	--	1c	1c
Line 7 encrypted	109	3939	VB-ID	aef5	key6_w3_0	97	cc
	110	0000	Mvid	7933	key6_w3_1	79	33
	111	0000	Maud	0649	key7_w0_0	06	49
	112	3939	VB-ID	989e	key7_w0_1	a1	a7

	113	0000	Mvid	ee75	key7_w1_0	ee	75
	114	0000	Maud	9b1a	key7_w1_1	9b	1a
	115	0000	null stream	c7fa	key7_w2_0	c7	fa
	116	0000	null stream	cd09	key7_w2_1	cd	09
	117	0000	null stream	4b82	key7_w3_0	4b	82
	118	0000	null stream	d199	key7_w3_1	d1	99
	119	0000	null stream	eadc	key8_w0_0	ea	dc

Table D.14

1 Lane, Inter-BS spacing = 24						
	Link Clock	Stream	Stream Type	Cipher Key [127:0]	Cipher Key Name [127:0]	Encrypted Stream
						Lane0
	-3	1c	SR	--	--	1c
	-2	3c	CP	--	--	3c
	-2	3c	CP	--	--	3c
	0	1c	SR	--	--	1c
Line 1 encrypted	1	39	VB-ID	e4	key0_b0	dd
	2	00	Mvid	9e	key0_b1	9e
	3	00	Maud	d1	key0_b2	d1
	4	39	VB-ID	f2	key0_b3	cb
	5	00	Mvid	da	key0_b4	da
	6	00	Maud	b2	key0_b5	b2
	7	39	VB-ID	97	key0_b6	ae
	8	00	Mvid	8d	key0_b7	8d
	9	00	Maud	49	key0_b8	49
	10	39	VB-ID	53	key0_b9	6a
	11	00	Mvid	70	key0_b10	70
	12	00	Maud	b1	key0_b11	b1
	13	00	null stream	b0	key0_b12	b0
	14	00	null stream	1e	key0_b13	1e
	15	00	null stream	4c	key0_b14	4c
	16	00	null stream	21	key0_b15	21
	17	00	null stream	c0	key1_b0	c0
	18	00	null stream	c2	key1_b1	c2
	19	00	null stream	d3	key1_b2	d3
	20	00	null stream	f2	key1_b3	f2
	21	bc	BS	60	key1_b4	bc
	22	3c	CP	9b	key1_b5	3c
	23	3c	CP	f3	key1_b6	3c
	24	bc	BS	87	key1_b7	bc
Line 2 encrypted	25	39	VB-ID	93	key1_b8	aa
	26	00	Mvid	52	key1_b9	52
	27	00	Maud	93	key1_b10	93
	28	39	VB-ID	fe	key1_b11	c7
	29	00	Mvid	4c	key1_b12	4c
	30	00	Maud	67	key1_b13	67
	31	39	VB-ID	da	key1_b14	e3

	32	00	Mvid	6d	key1_b15	6d
	33	00	Maud	8d	key2_b0	8d
	34	39	VB-ID	59	key2_b1	60
	35	00	Mvid	b7	key2_b2	b7
	36	00	Maud	fa	key2_b3	fa
	37	00	null stream	b3	key2_b4	b3
	38	00	null stream	3e	key2_b5	3e
	39	00	null stream	08	key2_b6	08
	40	00	null stream	b7	key2_b7	b7
	41	00	null stream	ee	key2_b8	ee
	42	00	null stream	a5	key2_b9	a5
	43	00	null stream	29	key2_b10	29
	44	00	null stream	f7	key2_b11	f7
	45	bc	BS	34	key2_b12	bc
	46	3c	CP	61	key2_b13	3c
	47	3c	CP	43	key2_b14	3c
	48	bc	BS	1e	key2_b15	bc
Line 3 encrypted	49	39	VB-ID	5e	key3_b0	67
	50	00	Mvid	ee	key3_b1	ee
	51	00	Maud	3a	key3_b2	3a
	52	39	VB-ID	64	key3_b3	5d
	53	00	Mvid	7c	key3_b4	7c
	54	00	Maud	1d	key3_b5	1d
	55	39	VB-ID	ca	key3_b6	f3
	56	00	Mvid	2f	key3_b7	2f
	57	00	Maud	f4	key3_b8	f4
	58	39	VB-ID	65	key3_b9	5c
	59	00	Mvid	d2	key3_b10	d2
	60	00	Maud	bd	key3_b11	bd
	61	00	null stream	f9	key3_b12	f9
	62	00	null stream	cb	key3_b13	cb
	63	00	null stream	77	key3_b14	77
	64	00	null stream	98	key3_b15	98
	65	00	null stream	aa	key4_b0	aa
	66	00	null stream	b1	key4_b1	b1
	67	00	null	16	key4_b2	16

			stream			
	68	00	null stream	cf	key4_b3	cf
	69	1c	SR	83	key4_b4	83
	70	7c	BF	ba	key4_b5	ba
	71	7c	BF	18	key4_b6	18
	72	1c	SR	cd	key4_b7	cd
Line 4 unencrypted	73	19	VB-ID	--	--	19
	74	00	Mvid	--	--	00
	75	00	Maud	--	--	00
	76	19	VB-ID	--	--	19
	77	00	Mvid	--	--	00
	78	00	Maud	--	--	00
	79	19	VB-ID	--	--	19
	80	00	Mvid	--	--	00
	81	00	Maud	--	--	00
	82	19	VB-ID	--	--	19
	83	00	Mvid	--	--	00
	84	00	Maud	--	--	00
	85	00	null stream	--	--	00
	86	00	null stream	--	--	00
	87	00	null stream	--	--	00
	88	00	null stream	--	--	00
	89	00	null stream	--	--	00
	90	00	null stream	--	--	00
	91	00	null stream	--	--	00
	92	00	null stream	--	--	00
	93	bc	BS	--	--	bc
	94	7c	BF	--	--	7c
	95	7c	BF	--	--	7c
	96	bc	BS	--	--	bc
Line 5 unencrypted	97	19	VB-ID	--	--	19
	98	00	Mvid	--	--	00
	99	00	Maud	--	--	00
	100	19	VB-ID	--	--	19
	101	00	Mvid	--	--	00
	102	00	Maud	--	--	00
	103	19	VB-ID	--	--	19
	104	00	Mvid	--	--	00
	105	00	Maud	--	--	00
	106	19	VB-ID	--	--	19

	107	00	Mvid	--	--	00
	108	00	Maud	--	--	00
	109	00	null stream	--	--	00
	110	00	null stream	--	--	00
	111	00	null stream	--	--	00
	112	00	null stream	--	--	00
	113	00	null stream	--	--	00
	114	00	null stream	--	--	00
	115	00	null stream	--	--	00
	116	00	null stream	--	--	00
	117	bc	BS	--	--	bc
	118	7c	BF	--	--	7c
	119	7c	BF	--	--	7c
	120	bc	BS	--	--	bc
Line 6 unencrypted	121	19	VB-ID	--	--	19
	122	00	Mvid	--	--	00
	123	00	Maud	--	--	00
	124	19	VB-ID	--	--	19
	125	00	Mvid	--	--	00
	126	00	Maud	--	--	00
	127	19	VB-ID	--	--	19
	128	00	Mvid	--	--	00
	129	00	Maud	--	--	00
	130	19	VB-ID	--	--	19
	131	00	Mvid	--	--	00
	132	00	Maud	--	--	00
	133	00	null stream	--	--	00
	134	00	null stream	--	--	00
	135	00	null stream	--	--	00
	136	00	null stream	--	--	00
	137	00	null stream	--	--	00
	138	00	null stream	--	--	00
	139	00	null stream	--	--	00
	140	00	null stream	--	--	00

	141	1c	SR	--	--	1c
	142	3c	CP	--	--	3c
	143	3c	CP	--	--	3c
	144	1c	SR	--	--	1c
Line 7 encrypted	145	39	VB-ID	2b	key4_b8	12
	146	00	Mvid	35	key4_b9	35
	147	00	Maud	1f	key4_b10	1f
	148	39	VB-ID	46	key4_b11	7f
	149	00	Mvid	ad	key4_b12	ad
	150	00	Maud	d5	key4_b13	d5
	151	39	VB-ID	18	key4_b14	21
	152	00	Mvid	ee	key4_b15	ee
	153	00	Maud	57	key5_b0	57
	154	39	VB-ID	57	key5_b1	6e
	155	00	Mvid	58	key5_b2	58
	156	00	Maud	a0	key5_b3	a0
	157	00	null stream	47	key5_b4	47
	158	00	null stream	fc	key5_b5	fc
	159	00	null stream	5f	key5_b6	5f
	160	00	null stream	c5	key5_b7	c5
	161	00	null stream	84	key5_b8	84

Table D.15