Notice

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

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

© Intel Corporation. Third-party brands and names are the property of their respective owners.

Acknowledgement

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

October 23, 2008  -  2.0 Revision. Publication on DCP LLC website

July 18, 2011  -  2.1 Revision. Publication on DCP LLC website

October 16, 2012  -  2.2 Revision. Publication on DCP LLC website
1 Introduction ................................................................................................................................. 5
  1.1 Scope ........................................................................................................................................ 5
  1.2 Definitions .................................................................................................................................. 5
  1.3 Overview .................................................................................................................................... 8
  1.4 Terminology .................................................................................................................................. 9
  1.5 References .................................................................................................................................... 9

2 Authentication Protocol ..................................................................................................................... 11
  2.1 Overview ...................................................................................................................................... 11
  2.2 Authentication and Key Exchange ............................................................................................... 12
    2.2.1 Pairing ...................................................................................................................................... 17
  2.3 Locality Check ............................................................................................................................... 18
  2.4 Session Key Exchange .................................................................................................................. 21
  2.5 Authentication with Repeaters ...................................................................................................... 22
    2.5.1 Upstream Propagation of Topology Information .................................................................. 22
    2.5.2 Downstream Propagation of Content Stream Management Information ......................... 28
  2.6 Link Synchronization .................................................................................................................... 29
  2.7 Key Derivation ............................................................................................................................. 29
  2.8 HDCP Transmitter State Diagram ............................................................................................... 30
  2.9 HDCP Receiver State Diagram .................................................................................................... 36
  2.10 HDCP Repeater State Diagram ................................................................................................. 37
    2.10.1 Propagation of Topology Errors ...................................................................................... 38
    2.10.2 HDCP Repeater Downstream State Diagram .................................................................. 38
    2.10.3 HDCP Repeater Upstream State Diagram .................................................................. 44
  2.11 Converters ................................................................................................................................. 47
    2.11.1 HDCP 2 – HDCP 1.x Converters ...................................................................................... 47
    2.11.2 HDCP 1.x – HDCP 2 Converters ...................................................................................... 49
  2.12 Session Key Validity .................................................................................................................... 51
  2.13 Random Number Generation ..................................................................................................... 51

3 HDCP Encryption ............................................................................................................................. 52
  3.1 Description ...................................................................................................................................... 52
  3.2 AV Stream ...................................................................................................................................... 52
  3.3 Abbreviations ............................................................................................................................... 53
  3.4 HDCP Cipher .................................................................................................................................. 53
  3.5 HDCP Cipher Block ...................................................................................................................... 55
  3.6 MPEG System Multiplexing .......................................................................................................... 56
    3.6.1 HDCP Registration Descriptor .......................................................................................... 56
    3.6.2 Transport Stream ................................................................................................................ 57
    3.6.3 Program Stream ................................................................................................................ 57
  3.7 Uniqueness of $k_s$ and $r_v$ ......................................................................................................... 57

4 Authentication Protocol Messages ................................................................................................... 60
  4.1 Abbreviations .................................................................................................................................. 60
  4.2 Control / Status Stream ................................................................................................................ 60
  4.3 Message Format ............................................................................................................................. 61
    4.3.1 AKE_Init (Transmitter to Receiver) ................................................................................... 61
    4.3.2 AKE_Send_Cert (Receiver to Transmitter) ........................................................................ 61
    4.3.3 AKE_No_Stored_km (Transmitter to Receiver) ................................................................. 61
    4.3.4 AKE_Stored_km (Transmitter to Receiver) ........................................................................ 61
    4.3.5 AKE_Send_rrx (Receiver to Transmitter) ........................................................................... 61
    4.3.6 AKE_Send_H_prime (Receiver to Transmitter) ................................................................. 62
    4.3.7 AKE_Send_Pairing_Info (Receiver to Transmitter) ......................................................... 62
    4.3.8 LC_Init (Transmitter to Receiver) ....................................................................................... 62
4.3.9 LC_Send_L_prime (Receiver to Transmitter) ............................................................................... 62
4.3.10 SKE_Send_Eks (Transmitter to Receiver) ................................................................................. 63
4.3.11 RepeaterAuth_Send_ReceiverID_List (Receiver to Transmitter) ........................................ 63
4.3.12 RTT_Ready (Receiver to Transmitter) ..................................................................................... 65
4.3.13 RTT_Challenge (Transmitter to Receiver) ................................................................................. 65
4.3.14 RepeaterAuth_Send_Ack (Transmitter to Receiver) ................................................................ 65
4.3.15 RepeaterAuth_Stream_Manage (Transmitter to Receiver) ...................................................... 65
4.3.16 RepeaterAuth_Stream_Ready (Receiver to Transmitter) ......................................................... 66
4.3.17 Receiver_AuthStatus (Receiver to Transmitter) ........................................................................ 66
4.3.18 AKE_Transmitter_Info (Transmitter to Receiver) .................................................................. 67
4.3.19 AKE_Receiver_Info (Receiver to Transmitter) ....................................................................... 67

5 Renewability ..................................................................................................................................... 69
  5.1 SRM Size and Scalability............................................................................................................. 70
  5.2 Updating SRMs.......................................................................................................................... 71

Appendix A. Core Functions and Confidentiality and Integrity of Values .................................................. 73
Appendix B. DCP LLC Public Key ........................................................................................................ 76
Appendix C. Bibliography (Informative) ............................................................................................... 77
Appendix D. Test Vectors ................................................................................................................... 78
  D.1 Facsimile Keys ............................................................................................................................ 78
  D.2 Authentication Protocol ............................................................................................................. 81
  D.3 Encryption .................................................................................................................................... 89
1 Introduction

1.1 Scope

This specification describes an interface independent adaptation of the High-bandwidth Digital Content Protection (HDCP) system, Revision 2.30. This specification can be applied over any wired or wireless interface as explained in subsequent chapters.

For the purpose of this specification, it is assumed that the Audiovisual content is transmitted over any wired or wireless display link. For example, this specification can be applied for the protection of Audiovisual content over an IP based wireless interface.

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.

Except when specified otherwise, HDCP 2.3-compliant Devices must interoperate with other devices compliant with HDCP2.3 and lower that are connected to their HDCP-protected Interface Ports using the same protocol. HDCP Transmitters must support HDCP Repeaters.

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

Implementations must include all elements of the content protection system described herein, unless the element is specifically identified as informative or optional. Adopters must also ensure that implementations satisfy the robustness and compliance rules described in the 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 described by Revision 2.00 of this specification along with its 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 2.3.** *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 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 $r_{\text{tx}}$ as part 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 $r_{\text{tx}}$ as part of the 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 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 logical connection point on an HDCP Device that supports an HDCP-protected Interface is referred to as an **HDCP-protected Interface Port**. A single connection can be made over 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.
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


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 7 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 $l_{c128}$. All HDCP Devices share the same Global Constant. $l_{c128}$ is provided only to HDCP adopters.

The HDCP Transmitter contains the 3072-bit RSA public key of DCP LLC denoted by $k_{pubdcp}$. 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 $cert_r$. Table 2.1 gives the fields contained in the certificate. All values are stored in big-endian format.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (bits)</th>
<th>Bit position</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver ID</td>
<td>40</td>
<td>4175:4136</td>
<td>Unique receiver identifier. It has the same format as an HDCP 1.x KSV i.e. it contains 20 ones and 20 zeroes</td>
</tr>
<tr>
<td>Receiver Public Key</td>
<td>1048</td>
<td>4135:3088</td>
<td>Unique RSA public key of HDCP Receiver denoted by $k_{pubrx}$. 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</td>
</tr>
<tr>
<td>Protocol Descriptor</td>
<td>4</td>
<td>3087:3084</td>
<td>Protocol descriptor field. Possible values are 0x0 or 0x1. 0x2 – 0xF – Reserved for future use</td>
</tr>
<tr>
<td>Reserved</td>
<td>12</td>
<td>3083:3072</td>
<td>Reserved for future definition. Must be 0x000</td>
</tr>
<tr>
<td>DCP LLC Signature</td>
<td>3072</td>
<td>3071:0</td>
<td>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</td>
</tr>
</tbody>
</table>

Table 2.1. Public Key Certificate of HDCP Receiver

The secret RSA private key is denoted by $k_{privrx}$. 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 a new \( r_x \) as part of the authentication initiation message, AKE_Init. Message formats are defined in Section 4.3. The HDCP Transmitter must ignore out of sequence authentication protocol messages received from the HDCP Receiver.

![Figure 2.1. Authentication and Key Exchange (Without Stored \( km \)](image-url)
Figure 2.2. Authentication and Key Exchange (With Stored $km$)

The HDCP Transmitter

- Initiates authentication by sending the initiation message, AKE_Init, containing a 64-bit pseudo-random value ($rtx$).

- Sends AKE_Transmitter_Info message to the HDCP Receiver before sending either AKE_No_Stored_km or AKE_Stored_km message to the receiver.

Note: The HDCP Transmitter may use mechanisms outside the scope of the HDCP Specification to determine whether the HDCP Receiver is an HDCP 2.0-compliant Device. If the HDCP Transmitter determines, using mechanisms outside the scope of the HDCP Specification, that the HDCP Receiver is an HDCP 2.0-compliant Device, it need not send the AKE_Transmitter_Info message to the HDCP Receiver.

- Receives AKE_Send_Cert from the receiver containing REPEATER and $cert_{rx}$ values. REPEATER indicates whether the connected receiver is an HDCP Repeater.

- Receives AKE_Receiver_Info message from the receiver if the receiver is not an HDCP 2.0-compliant Device. If AKE_Receiver_Info message is not received within 100 ms from the transmission of AKE_Transmitter_Info message, it indicates to the HDCP Transmitter that the attached HDCP Receiver is an HDCP 2.0-compliant Device.

- 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 \( k_{pubdcp} \). 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 \( k_{pubrx} (E_{pubdcp}(km)) \) and sends AKE_No_Stored_km message to the receiver containing the 1024-bit \( E_{pubdcp}(km) \). 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 \( k_{pubdcp} \). 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.

- Receives AKE_Send_rrx message from the receiver containing the 64-bit pseudo-random value \( r_{rx} \).

- 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 \( H = \text{HMAC-SHA256}(r_{rx} \text{ XOR REPEATER} || \text{AKEReceiver}_\text{Info.VERSION} || \text{AKEReceiver}_\text{Info.RECEIVER\_CAPABILITY\_MASK} || \text{AKETransmitter}_\text{Info.VERSION} || \text{AKETransmitter}_\text{Info.TRANSMITTER\_CAPABILITY\_MASK}, k_d) \) if the Protocol Descriptor field in \( certrx \) is equal to 0x1. HMAC-SHA256 is computed over \( r_{rx} \) XOR REPEATER concatenated with AKE_Receiver_Info.VERSION, AKE_Receiver_Info.RECEIVER\_CAPABILITY\_MASK, AKE_Transmitter_Info.VERSION, AKE_Transmitter_Info.TRANSMITTER\_CAPABILITY\_MASK, and the key used for HMAC is \( k_d \). REPEATER is XORed with the least significant byte of \( r_{rx} \). All values are in big-endian order.

Computes 256-bit \( H = \text{HMAC-SHA256}(r_{rx} \text{ XOR REPEATER}, k_d) \) if the Protocol Descriptor field in \( certrx \) is equal to 0x0. HMAC-SHA256 is computed over \( r_{rx} \) XOR REPEATER and the key used for HMAC is \( k_d \). REPEATER is XORed with the least significant byte of \( r_{rx} \).
- Receives AKE_Send_H_prime message from the receiver containing the 256-bit $H'$. This message must be received within one second after sending $E_{\text{pub}}(km)$ (AKE_No_Stored_km) to the receiver. Authentication fails and the authentication protocol is aborted if the message is not received within one second or there is a mismatch between $H$ and $H'$.

  - If the HDCP Transmitter has a 128-bit Master Key $km$ stored corresponding to the Receiver ID (See Section 2.2.1)
    - Sends AKE_Stored_km message to the receiver with the 128-bit $E_{kh}(km)$ 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 $k_{pub,dcp}$. 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.

- Receives AKE_Send_rrx message from the receiver containing the 64-bit pseudo-random value ($rrx$) from the receiver.

- Performs key derivation as explained in Section 2.7 to generate 256-bit $kd$. $kd = dkey_0 \| dkey_1$, 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 $H = \text{HMAC-SHA256}(rrx \oplus \text{REPEATER} \mid AKE_{\text{Receiver}}_{\text{Info}}.\text{VERSION} \mid AKE_{\text{Receiver}}_{\text{Info}}.\text{RECEIVER\_CAPABILITY\_MASK} \mid AKE_{\text{Transmitter}}_{\text{Info}}.\text{VERSION} \mid AKE_{\text{Transmitter}}_{\text{Info}}.\text{TRANSMITTER\_CAPABILITY\_MASK} \mid k_d)$ if the Protocol Descriptor field in $cert_{rx}$ is equal to 0x1. HMAC-SHA256 is computed over $rrx \oplus \text{REPEATER}$ concatenated with $AKE_{\text{Receiver}}_{\text{Info}}.\text{VERSION}$, $AKE_{\text{Receiver}}_{\text{Info}}.\text{RECEIVER\_CAPABILITY\_MASK}$, $AKE_{\text{Transmitter}}_{\text{Info}}.\text{VERSION}$ and $AKE_{\text{Transmitter}}_{\text{Info}}.\text{TRANSMITTER\_CAPABILITY\_MASK}$ and the key used for HMAC is $kd$. REPEATER is XORed with the least significant byte of $rrx$. All values are in big-endian order.

  Computes 256-bit $H = \text{HMAC-SHA256}(rrx \oplus \text{REPEATER}, kd)$ if the Protocol Descriptor field in $cert_{rx}$ is equal to 0x0. HMAC-SHA256 is computed over $rrx \oplus \text{REPEATER}$ and the key used for HMAC is $kd$. REPEATER is XORed with the least significant byte of $rrx$.

- Receives AKE_Send_H_prime message from the receiver containing the 256-bit $H'$. This message must be received within 200 ms after sending the AKE_Stored_km message to the receiver. Authentication
fails and the authentication protocol is aborted if the message is not received within 200 ms or there is a mismatch between H and $H'$. The HDCP Receiver

- Sends AKE_Send_Cert message in response to AKE_Init

- If AKE_Transmitter_Info message is received, sends AKE_Receiver_Info message to the transmitter after sending the AKE_Send_Cert message to the transmitter.

- Generates and sends 64-bit $r_{rx}$ as part of the AKE_Send_rxx message immediately after receiving either AKE_No_Stored_km or AKE_Stored_km message from the transmitter.
  - If AKE_No_Stored_km is received, the HDCP Receiver
    - Decrypts $k_{nr}$ with $k_{priv_{rx}}$ using RSAES-OAEP decryption scheme.
    - Performs key derivation as explained in Section 2.7 to generate 256-bit $k_d = dkey_0 \parallel dkey_1$, 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 $H' = \text{HMAC-SHA256}(r_{rx} \oplus \text{REPEATER} \parallel \text{AKE_Receiver_Info.VERSION} \parallel \text{AKE_Receiver_Info.RECEIVER_CAPABILITY_MASK} \parallel \text{AKE_Transmitter_Info.VERSION} \parallel \text{AKE_Transmitter_Info.TRANSMITTER_CAPABILITY_MASK}, k_d)$ if the Protocol Descriptor field in its public key certificate, $cert_{rx}$, is equal to 0x1 and the HDCP Transmitter is not HDCP2.0-compliant and is not HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is not equal to 0x01).
  
  Computes $H' = \text{HMAC-SHA256}(r_{rx} \oplus \text{REPEATER}, k_d)$ if the Protocol Descriptor field in its public key certificate, $cert_{rx}$, is equal to 0x0 or the HDCP Transmitter is HDCP2.0-compliant or is HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is equal to 0x01).

  The HDCP Receiver sends AKE_Send_H_prime message immediately after computation of $H'$ to ensure that the message is received by the transmitter within the specified one second timeout at the transmitter.
  - If AKE_Stored_km is received, the HDCP Receiver
    - Computes 128-bit $k_h = \text{SHA-256}(k_{priv_{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 $k_d = dkey_0 \parallel dkey_1$, 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 $H' = \text{HMAC-SHA256}(r_t \text{ XOR REPEATER} \ || \ \text{AKE_Receiver_Info.VERSION} \ || \ \text{AKE_Receiver_Info. RECEIVER\_CAPABILITY\_MASK} \ || \ \text{AKE_Transmitter_Info.VERSION} \ || \ \text{AKE_Transmitter_Info. TRANSMITTER\_CAPABILITY\_MASK}, \ k_d)$ if the Protocol Descriptor field in its public key certificate, $cert_{rx}$, is equal to 0x1 and the HDCP Transmitter is not HDCP2.0-compliant and is not HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is not equal to 0x01).

Computes $H' = \text{HMAC-SHA256}(r_t \text{ XOR REPEATER}, \ k_d)$ if the Protocol Descriptor field in its public key certificate, $cert_{rx}$, is equal to 0x0 or the HDCP Transmitter is HDCP2.0-compliant or HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is equal to 0x01).

The HDCP Receiver sends AKE_Send_H_prime message immediately after computation of $H'$ to ensure that the message is received by the transmitter within the specified 200 ms timeout at the transmitter.

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

If the HDCP Receiver does not receive AKE_Transmitter_Info message before the reception of AKE_No_Stored_km or AKE_Stored_km message, it indicates that the HDCP Transmitter is an HDCP 2.0-compliant Device.

If the HDCP Transmitter does not receive AKE_Receiver_Info message within 100 ms of the transmission of AKE_Transmitter_Info message, it indicates that the HDCP Receiver is an HDCP 2.0-compliant Device.

### 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 $k_b = \text{SHA-256}(k_{privrx})[127:0]$.
- Generates 128-bit $E_{id}(k_m)$ by encrypting $k_m$ with $k_b$ using AES as illustrated in Figure 2.3.
- Sends AKE_Send_Pairing_Info to the transmitter containing the 128-bit $E_{id}(k_m)$.

On receiving AKE_Send_Pairing_Info message, the HDCP Transmitter may persistently store $m$ (which is $r_t$ concatenated with $r_x (r_t \ || \ r_x)$, $k_m$ and $E_{id}(k_m)$ along with Receiver ID.

If AKE_Send_Pairing_Info is not received by the HDCP Transmitter within 200 ms of the reception of AKE_Send_H_prime, authentication fails and the authentication protocol is aborted.

Note: The HDCP Transmitter may store in its non-volatile storage $m$, $k_m$ and $E_{id}(k_m)$ along with corresponding Receiver IDs of all HDCP Receivers with which pairing was implemented by the HDCP Transmitter.
If the HDCP Receiver is HDCP 2.0-compliant or AKE_Receiver_Info.VERSION = 0x01, the HDCP Transmitter must not store pairing information \((m, k_m, E_{id}(k_m) \text{ and } \text{Receiver ID})\) corresponding to the receiver.

Figure 2.3 illustrates the encryption of \(k_m\) with \(k_h\).

\[ m \]

\[ 128 \]

\[ k_h \]

\[ 128 \]

\[ \text{AES} \]

\[ \text{Encrypted } k_m \]

If HDCP Transmitter not HDCP 2.0-compliant and not HDCP 2.1-compliant, HDCP Receiver sets
\[ m = r_s || r_x \]

If HDCP Transmitter is HDCP 2.0-compliant or is HDCP 2.1-compliant, HDCP Receiver sets
\[ m = r_s || 0x0000000000000000 \]

**Figure 2.3. \(E_{id}(k_m)\) Computation**

If the HDCP Transmitter is not HDCP 2.0-compliant and is not HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is not equal to 0x01), the HDCP Receiver constructs 128-bit \(m\) by concatenating \(r_s\) with \(r_x\) \((r_s || r_x)\). Both values are in big-endian order.

If the HDCP Transmitter is HDCP 2.0-compliant or is HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is equal to 0x01), 128-bit \(m\) is constructed by the HDCP Receiver by appending 64 0s to \(r_s\). \(r_s\) is 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 \(r_s\) to the downstream receiver.

If AKE_Receiver_Info.VERSION = 0x01 and the HDCP Transmitter has set its TRANSMITTER_LOCALITY_PRECOMPUTE_SUPPORT bit to one in the AKE_Transmitter_Info message transmitted to the HDCP Receiver, the HDCP Transmitter must initiate re-authentication with the HDCP Receiver with the TRANSMITTER_LOCALITY_PRECOMPUTE_SUPPORT bit set to zero.
If the HDCP Receiver is HDCP 2.0-compliant or if the RECEIVER_LOCALITY_PRECOMPUTE_SUPPORT bit received as part of the AKEReceiver_Info message is set to zero or the transmitter has set the TRANSMITTER_LOCALITY_PRECOMPUTE_SUPPORT bit to zero in its AKE_Transmitter_Info message, the HDCP Transmitter

- Initiates locality check by sending LC_Init message containing a 64-bit pseudo-random nonce \( r_n \) to the HDCP Receiver.

- Sets its watchdog timer to 7 ms. Locality check fails if the watchdog timer expires before LC_Send_L_prime message is received.

- Computes \( L = \text{HMAC-SHA256}(r_n, k_d \oplus r_{rx}) \) where HMAC-SHA256 is computed over \( r_n \) and the key used for HMAC is \( k_d \oplus r_{rx} \), where \( r_{rx} \) is XORed with the least-significant 64-bits of \( k_d \).

- On receiving LC_Send_L_prime message, compares \( L \) and \( L' \). Locality check fails if \( L \) is not equal to \( L' \).

If the RECEIVER_LOCALITY_PRECOMPUTE_SUPPORT bit received as part of the AKEReceiver_Info message is set to one and the transmitter has set the TRANSMITTER_LOCALITY_PRECOMPUTE_SUPPORT bit to one in its AKE_Transmitter_Info message, the HDCP Transmitter

- Initiates locality check by sending LC_Init message containing a 64-bit pseudo-random nonce \( r_n \) to the HDCP Receiver.

- Computes 256-bit \( L = \text{HMAC-SHA256}(r_n || r_n, k_d \oplus r_{rx}) \) where HMAC-SHA256 is computed over \( r_n || r_n \) and the key used for HMAC is \( k_d \oplus r_{rx} \), where \( r_{rx} \) is XORed with the least-significant 64-bits of \( k_d \). All values are in big-endian order.

- On receiving the RTT_Ready message from the receiver, the transmitter sends an RTT_Challenge message containing the least significant 128-bits of \( L \).

- Sets its watchdog timer to 7 ms. Locality check fails if the watchdog timer expires before LC_Send_L_prime message message is received.

- On receiving LC_Send_L_prime message, the HDCP Transmitter compares the received value with the most significant 128-bits of \( L \) and locality check fails if there is a mismatch.

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 and Figure 2.5 illustrate locality check between the HDCP Transmitter and HDCP Receiver.
If the HDCP Transmitter is HDCP 2.0-compliant or if the TRANSmitter_LOCALITY_PRECOMPUTE_SUPPORT bit received as part of the AKE_Transmitter_Info message is set to zero or the receiver has set the RECEIVER_LOCALITY_PRECOMPUTE_SUPPORT bit to zero in its AKE_Receiver_Info message, the HDCP Receiver

- Computes a 256-bit value $L' = \text{HMAC-SHA256}(r_n, k_d \text{XOR} r_n)$.
- Sends LC_Send_L_prime message containing 256-bit $L'$. 

If the TRANSmitter_LOCALITY_PRECOMPUTE_SUPPORT bit received as part of the AKE_Transmitter_Info message is set to one and the receiver has set the
RECEIVER_LOCALITY_PRECOMPUTE_SUPPORT bit to one in its AKE_Receiver_Info message, the HDCP Receiver

- Computes 256-bit $L' = \text{HMAC-SHA256}(r_n \text{ xor } r_x)$ if AKE_Transmitter_Info.VERSION = 0x01.

  Computes 256-bit $L' = \text{HMAC-SHA256}(r_n||r_o \text{ xor } r_x)$ if AKE_Transmitter_Info.VERSION is not equal to 0x01,

- Sends RTT_Ready message to the transmitter when $L'$ calculation is complete and the receiver is ready for the RTT Challenge.

- On receiving the RTT_Challenge message from the transmitter, if the value received in the RTT_Challenge message matches the least significant 128 bits of $L'$, the receiver sends an LC_Send_L_prime message containing the most significant 128-bits of $L'$.

In the case of a locality check failure due to expiration of the watchdog timer or due to mismatch of L and $L'$ (or the most significant 128-bits 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. HDCP Encryption may be enabled 200 ms after the transmission of the encrypted Session Key to the HDCP Receiver and at no time prior. Content encrypted with the Session Key $k_s$ 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 $r_v$.

- Computes a 256-bit HMAC of $r_v$ - HMAC-SHA256($r_v$, $k_d$) where HMAC-SHA256 is computed over $r_v$ and the key used for HMAC is $k_d$, if the HDCP Receiver is compliant with HDCP2.3 or higher (i.e. AKE_Receiver_Info.VERSION = 0x03 or higher)

- Performs key derivation as explained in Section 2.7 to generate 128-bit dkey2 where dkey2 is the derived key when ctr = 2.

- Computes 128-bit $E_{dkey2}(k_s) = k_s \text{ xor } (dkey2 \text{ xor } r_x)$, where $r_x$ is XORED with the least-significant 64-bits of dkey2.

- Sends SKE_Send_Eks message containing $E_{dkey2}(k_s) \text{ xor } r_v$ and HMAC of $r_v$ to the HDCP Receiver.

On receiving SKE_Send_Eks message, the HDCP Receiver

- Computes a 256-bit HMAC of $r_v$ - HMAC-SHA256($r_v$, $k_d$), if the HDCP Transmitter is compliant with HDCP2.3 or higher (i.e. AKE_Transmitter_Info.VERSION = 0x03 or
higher). Compares the computed HMAC with the HMAC received as part of the SKE_Send_Eks message. A mismatch of the HMAC values results in an authentication failure and the authentication protocol is aborted.

- Performs key derivation as explained in Section 2.7 to generate 128-bit dkey2 where dkey2 is the derived key when ctr = 2.

- Computes \( k_i = E_{dkey2}(k_i) \) XOR (dkey2 XOR rrx)

### 2.5 Authentication with Repeaters

The HDCP Transmitter executes authentication with repeaters after Session Key exchange and only when REPEATER is ‘true’, 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 Synchronization. The Link Synchronization process is explained in Section 2.6.

#### 2.5.1 Upstream Propagation of Topology Information

**Figure 2.6. Upstream Propagation of Topology Information**

Figure 2.6 illustrates the upstream propagation of topology information. This stage assembles a list of all downstream Receiver IDs 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.

In order to add the Receiver ID list of the connected downstream HDCP Repeater, it is necessary for the HDCP Repeater to verify the integrity of the list. If the connected HDCP Repeater is not an HDCP 2.0-compliant Device, the HDCP Repeater verifies the integrity of the list by computing $V$ and checking the most significant 128-bits of $V$ against the most significant 128 bits of $V'$ received as part of the RepeaterAuth_Send_ReceiverID List message from the connected downstream HDCP Repeater. If the connected HDCP Repeater is an HDCP 2.0-compliant Device, the HDCP Repeater verifies the integrity of the list by computing $V$ and comparing $V$ against $V'$. If the values do not match, the downstream Receiver ID list integrity check fails, and the HDCP Repeater must not add the Receiver ID list received from the downstream HDCP Repeater to its Receiver ID list.

When the HDCP Repeater has assembled the complete list of Receiver IDs 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 connected to an HDCP 2.0-compliant upstream HDCP Transmitter and an HDCP Transmitter connected to an HDCP 2.0-compliant HDCP Repeater computes respective $V'$ and $V$ values as given below. HMAC-SHA256 is computed over the concatenation of Receiver ID list, DEPTH, DEVICE_COUNT, MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED received as part of the RepeaterAuth_Send_ReceiverID List message. The key used for HMAC is $k_d$.

$$V' \text{ (or } V) = \text{HMAC-SHA256}(\text{Receiver ID list} \| \text{DEPTH} \| \text{DEVICE_COUNT} \| \text{MAX DEVS EXCEEDED} \| \text{MAX CASCADE EXCEEDED}, k_d)$$

An HDCP Repeater connected to an upstream HDCP Transmitter that is not HDCP 2.0-compliant and an HDCP Transmitter connected to an HDCP Repeater that is not HDCP 2.0-compliant computes respective $V'$ and $V$ values as given below. HMAC-SHA256 is computed over the concatenation of Receiver ID list, DEPTH, DEVICE_COUNT, MAX DEVS_EXCEEDED, MAX_CASCADE_EXCEEDED, HDCP2_LEGACY DEVICE_DOWNSTREAM, HDCP1 DEVICE DOWNSTREAM and $\text{seq_num}_V$ received as part of the RepeaterAuth_Send_ReceiverID List message. The key used for HMAC is $k_d$.

$$V' \text{ (or } V) = \text{HMAC-SHA256}(\text{Receiver ID list} \| \text{DEPTH} \| \text{DEVICE_COUNT} \| \text{MAX DEVS EXCEEDED} \| \text{MAX CASCADE EXCEEDED} \| \text{HDCP2 LEGACY DEVICE DOWNSTREAM} \| \text{HDCP1 DEVICE DOWNSTREAM} \| \text{seq_num}_V, k_d)$$

Receiver ID list is formed by appending downstream Receiver IDs in big-endian order. When the Receiver ID list, $V'$, DEPTH, DEVICE_COUNT, and if applicable, HDCP2_LEGACY DEVICE_DOWNSTREAM and HDCP1 DEVICE_DOWNSTREAM are available, the HDCP Repeater sends RepeaterAuth_Send_ReceiverID List message to the upstream HDCP Transmitter. The HDCP Repeater sends $V'$ if the upstream transmitter is HDCP.
2.0-compliant and the most significant 128-bits of $V'$ if the upstream transmitter is not HDCP 2.0-compliant.

The HDCP Repeater initializes $seq\_num\_V$ to 0 at the beginning of the HDCP Session i.e. after $r_{tx}$ 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 received from the HDCP Repeater and the transmitter must disable HDCP Encryption if encryption is enabled, restart authentication by the transmission of a new $r_{tx}$ as part of the AKE_Init message.

When an HDCP Repeater receives HDCP2_LEGACY_DEVICE_DOWNSTREAM = ‘true’ or HDCP1DEVICE_DOWNSTREAM = ‘true’ from a downstream HDCP Repeater, it must propagate this information to the upstream HDCP Transmitter by setting the corresponding values to ‘true’ in the RepeaterAuth_Send_ReceiverID_List message.

If HDCP2_LEGACY_DEVICE_DOWNSTREAM = ‘true’ or HDCP1DEVICE_DOWNSTREAM = ‘true’, 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 RepeaterAuth_Send_ReceiverID_List message is received, the HDCP Transmitter verifies the integrity of the Receiver ID list by computing $V$ and comparing either $V$ and $V'$ (if the connected HDCP Repeater is HDCP 2.0-compliant) or the most significant 128-bits of $V$ and $V'$ (if the connected HDCP Repeater is not HDCP 2.0-compliant). 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 IDs 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) and the HDCP Repeater is not HDCP 2.0-compliant, the HDCP Transmitter (including downstream port of HDCP Repeater) sends 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 one second from the transmission of the RepeaterAuth_Send_ReceiverID_List message to the HDCP Transmitter if the HDCP Transmitter is not HDCP 2.0-compliant 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 one second, the HDCP Repeater must send the Receiver_AuthStatus message with the REAUTH_REQ set to ‘true’ and must transition in to an unauthenticated state (See Section 2.10.3).

If the upstream HDCP Transmitter receives a Receiver_AuthStatus message with REAUTH_REQ set to ‘true’, it may initiate re-authentication with the HDCP Repeater by the transmission of a new $r_{tx}$.

After transmitting the SKE_Send_Eks message, the HDCP Transmitter, having determined that REPEATER received earlier in the protocol is ‘true’, sets a three second watchdog timer. If the
RepeaterAuth_Send_ReceiverID_List message 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 and the upstream HDCP Transmitter is not HDCP 2.0-compliant, the HDCP Repeater must send the RepeaterAuth_Send_ReceiverID_List message to the upstream HDCP Transmitter which 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 a HCOP 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 2.7, R1 has three downstream HDCP Receivers connected to it. It reports a DEPTH of one and a DEVICE_COUNT of three.

In Figure 2.8, R1 reports a DEPTH of two and a DEVICE_COUNT of four.
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 = ‘true’ 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 = ‘true’ 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 $V'$ (or the most significant 128-bits of $V'$), DEPTH, DEVICE_COUNT, Receiver ID list and if applicable, HDCP2_LEGACY_DEVICE_DOWNSTREAM and HDCP1DEVICE_DOWNSTREAM.

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 IDs, including the Receiver ID of the attached downstream HDCP Receiver or HDCP Repeater and any Receiver IDs 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.
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 has been provided, to receive the RepeaterAuth_Send_ReceiverID_List message by the upstream transmitter, 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

![Diagram](image)

Figure 2.10. 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 only if the attached HDCP Repeater is not an HDCP 2.0-compliant Device. 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 4.3.15) may be transmitted by the HDCP Repeater to all HDCP Devices. Type 1 Content Streams (see Section 4.3.15) 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 as instructed by the corresponding Upstream Content Control Function.

The most upstream HDCP Transmitter must not transmit Type 1 Content Streams to HDCP 1.x-compliant Devices and HDCP 2.0-compliant Repeaters as instructed by the corresponding Upstream Content Control Function.

The HDCP Transmitter must send the RepeaterAuth_Stream_Manage message specifying Type values assigned to Content Streams, to the attached HDCP Repeater at least 100ms 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 $r_n$ 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 4.3.15) 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(kd). 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 r_c as part of the AKE_Init message.

\[ M' = \text{HMAC-SHA256(STREAMID_TYPE || seq_num_M, SHA256(kd)).} \]

M' must be sent by the HDCP Repeater to the HDCP Transmitter as part of the RepeaterAuth_Stream_Ready message.

The HDCP Transmitter must receive the RepeaterAuth_Stream_Ready message within 100 ms after the transmission of RepeaterAuth_Stream_Manage message. 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 received, 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. If the RepeaterAuth_Stream_Ready message is not received within 100 ms or if M is not equal to M', the HDCP Transmitter must not transmit the Content Streams identified in the corresponding RepeaterAuth_Stream_Manage message. 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 Synchronization

After successful completion of SKE, HDCP Encryption is enabled and encrypted content starts to flow between the HDCP Transmitter and the HDCP Receiver. As explained in Section 3.4, the presence of the PES Header HDCP Private Data block indicates that HDCP Encryption is enabled and the PES payload is encrypted. Once encrypted content starts to flow, a periodic Link Synchronization is performed to maintain cipher synchronization between the HDCP Transmitter and the HDCP Receiver.

Link Synchronization is achieved every time a PES Header is transmitted, by the inclusion of inputCtr and streamCtr in the header. (See Section 3.4 for details about inputCtr and streamCtr). The HDCP Receiver updates its inputCtr corresponding to the stream (as indicated by the streamCtr value) with the inputCtr value received from the transmitter.

2.7 Key Derivation

Key derivation is illustrated in Figure 2.11.
The HDCP Transmitter sets IV = $r_{tx} \parallel (r_{tx} \text{ XOR } \text{ctr})$ if the HDCP Receiver is not HDCP 2.0-compliant and is not HDCP 2.1-compliant (i.e. AKE_Receiver_Info.VERSION is not equal to 0x01). If the HDCP Receiver is HDCP 2.0-compliant or is HDCP 2.1-compliant (i.e. AKE_Receiver_Info.VERSION = 0x01), the HDCP Transmitter sets IV = $r_{tx} \parallel \text{ctr}$. All values are in big-endian order.

The HDCP Receiver sets IV = $r_{tx} \parallel (r_{tx} \text{ XOR } \text{ctr})$ if the HDCP Transmitter is not HDCP2.0-compliant and is not HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is not equal to 0x01). If the HDCP Transmitter is HDCP2.0-compliant or is HDCP 2.1-compliant (i.e. AKE_Transmitter_Info.VERSION is equal to 0x01), the HDCP Receiver sets IV = $r_{tx} \parallel \text{ctr}$. 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 $r_{tx}$ 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. The method used by the HDCP Transmitter to determine whether the receiver is HDCP-capable is outside the scope of this specification.

Figure 2.12. HDCP Transmitter Link 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 may begin sending an unencrypted signal with HDCP Encryption disabled. The transmitted signal can be a low value content or informative on-screen display. If low-value content is transmitted, this will ensure that a
valid video signal is displayed to the user before and during authentication. At any time a Receiver Connected Indication received from the connected HDCP 2.0-compliant HDCP Repeater causes the transmitter to transition in to this state.

**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. This step may be defined as part of the setup and discovery procedures and is outside the scope of this specification. 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 may continue 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 km.** In this state, the HDCP Transmitter initiates authentication by sending AKE_Init message containing $r_x$ to the HDCP Receiver and sends AKE_Transmitter_Info message to the HDCP Receiver. It receives AKE_Send_Cert from the receiver containing REPEATER and certrx and AKE_Receiver_Info message (if the HDCP Receiver is not HDCP 2.0-compliant). If the HDCP Transmitter does not receive AKE_Receiver_Info message within 100 ms of the transmission of AKE_Transmitter_Info message, it indicates that the HDCP Receiver is an HDCP 2.0-compliant Device.

If the HDCP Transmitter does not have $k_m$ stored corresponding to the Receiver ID, it generates $E_{pub}(km)$ and sends $E_{pub}(km)$ as part of the AKE_No_Stored_km message to the receiver after verification of signature on certrx. 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 receives AKE_Send_rrx message containing $r_x$ from the receiver. It computes H, receives AKE_Send_H_prime message from the receiver containing $H'$ within one second after sending AKE_No_Stored_km to the receiver and compares $H'$ against H.

If the HDCP Transmitter has $k_m$ stored corresponding to the Receiver ID, it sends AKE_Stored_km message containing $E_{hash}(km)$ and $m$ to the receiver, performs integrity check on the SRM, checks to see whether the Receiver ID of the connected HDCP Device is in the revocation list, and receives $r_x$ as part of AKE_Send_rrx message from the receiver. It computes H, receives AKE_Send_H_prime message from the receiver containing $H'$ within 200 ms after sending AKE_Stored_km to the receiver 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 certrx, 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 one second after sending AKE_No_Stored_km or within 200 ms after sending AKE_Stored_km to the receiver.

**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 $L'$ (or the most significant 128-bits of $L$) is not received within 7 ms and the watchdog timer at the HDCP Transmitter expires or on a mismatch between L and $L'$ (or the most significant 128-bits of $L'$).

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, $E_{\text{key}}(k_s)$, and $r_w$ 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 is ‘false’ (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 Synchronization is performed to maintain cipher synchronization between the HDCP Transmitter and the HDCP Receiver.

Transition A4:A6. REPEATER is ‘true’ (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 the RepeaterAuth_Send_ReceiverID_List is received.

Transition A6:A7. RepeaterAuth_Send_ReceiverID_List message is received.

State A7: Verify Receiver ID List. If a transition in to this state occurs from State A6, the watchdog timer is cleared. If both MAX_DEVS_EXCEEDED and MAXCASCADE_EXCEEDED are not ‘true’, computes $V$. If the connected HDCP Repeater is HDCP 2.0-compliant, compares $V$ and $V'$. If the connected HDCP Repeater is not HDCP 2.0-compliant, compares the most significant 128-bits of $V$ and $V'$. The Receiver IDs from the Receiver ID list are compared against the current revocation list.

Transition A7:H1. This transition is made if a mismatch occurs between $V$ and $V'$ (if the connected HDCP Repeater is HDCP 2.0-compliant) or the most significant 128-bits of $V$ and $V'$ (if the connected HDCP Repeater is not HDCP 2.0-compliant). This transition is also made if any of the Receiver IDs in the Receiver ID list are found in the current revocation list or if the HDCP Transmitter detects a roll-over of $\text{seq\_num\_V}$ (if the repeater is not HDCP 2.0-compliant). A MAXCASCADE_EXCEEDED or MAXDEVS_EXCEEDED error also causes this transition.

Transition A7:A5. This transition occurs if the connected HDCP Repeater is HDCP 2.0-compliant, on successful verification of $V$ and $V'$, none of the reported Receiver IDs are in the current revocation list, and the downstream topology does not exceed specified maximums.
**Transition A7:A8.** This transition occurs if the connected HDCP Repeater is not HDCP 2.0-compliant, on successful verification of the most significant 128-bits of $V$ and $V'$, none of the reported Receiver IDs 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 one second from the transmission of the RepeaterAuth_Send_ReceiverID_List message to the HDCP Transmitter.

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

**Transition A5:H1.** This transition occurs if a Receiver_AuthStatus message with the REAUTH_REQ set to ‘true’ is received.

**Transition A5:A7.** This transition occurs whenever a RepeaterAuth_Send_ReceiverID_List message is received from the connected HDCP Repeater that is not HDCP 2.0-compliant.

**State A9: Content Stream Management.** This stage is implemented if Content Stream is to be transmitted and the connected HDCP Repeater is not HDCP 2.0-compliant. The HDCP Transmitter sends the RepeaterAuth_Stream_Manage message specifying Type values assigned to Content Streams, to the attached HDCP Repeater at least 100ms before the transmission of the corresponding Content Streams after HDCP Encryption. It must receive the RepeaterAuth_Stream_Ready message from the HDCP Repeater within 100 ms after the transmission of RepeaterAuth_Stream_Manage message and verifies $M'$. This step fails if the RepeaterAuth_Stream_Ready message is not received within 100 ms 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 Synchronization (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 and the connected HDCP Repeater is not HDCP 2.0-compliant. Also, the transition from State A9 must return to the appropriate state to allow for undisrupted operation.

**Note:** The HDCP Transmitter must not transmit Type 1 Content Streams to HDCP 1.x-compliant Devices and HDCP 2.0-compliant Repeaters as instructed by the corresponding Upstream Content Control Function.

**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 Synchronization (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 synchronization process (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 \( r_{tx} \) across all its HDCP-protected interface ports, as explained in Section 3.7. 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 an \( r_{tx} \) as part 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 \( r_{tx} \) from the HDCP Transmitter to trigger the authentication protocol.

**Transition B0:B1.** \( r_{tx} \) is received as part of the AKE_Init message from the HDCP Transmitter.

**State B1: Compute \( k_m \).** In this state, the HDCP Receiver sends AKE_Send_Cert message in response to AKE_Init, sends AKE_Receiver_Info message to the transmitter if AKE_Transmitter_Info message is received from the transmitter, generates and sends \( r_{tx} \) as part of AKE_Send_rrx message. If AKE_No_Stored_km is received, it decrypts \( k_m \) with \( k_{priv_{rx}} \) calculates \( H' \). It sends AKE_Send_H_prime message immediately after computation of \( H' \) to ensure that the message is received by the transmitter within the specified one second timeout at the transmitter.

If the HDCP Receiver does not receive AKE_Transmitter_Info message before the reception of AKE_No_Stored_km or AKE_Stored_km message, it indicates that the HDCP Transmitter is an HDCP 2.0-compliant Device.

If AKE_Stored_km is received, the HDCP Receiver decrypts \( E_{tid}(k_m) \) to derive \( k_m \) and calculates \( H' \). It sends AKE_Send_H_prime message immediately after computation of \( H' \) to ensure that the message is received by the transmitter within the specified 200 ms timeout at the transmitter.
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 send 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 sends LC_Send_L_prime message.

**Transition B2: B1.** Should the HDCP Transmitter send 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 \( Edkey(k_s) \) to derive \( k_s \).

**Transition B3: B1.** Should the HDCP Transmitter send 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 updates its inputCtr corresponding to the stream (as indicated by the streamCtr value) with the inputCtr value received from the transmitter.

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

### 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 by propagating the Receiver Connected Indication to the upstream HDCP Transmitter.

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 from its downstream ports. Whenever authentication is attempted by the upstream transmitter by sending an \( r_n \) value, 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 set to ‘false’ if it wishes to receive HDCP Content, but must not pass HDCP Content to downstream devices.

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 = ‘true’ 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 = ‘true’ 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.
Reset

Receiver Disconnected
Indication

Receiver Connected
Indication

P0:
No Rx
Attached

Upstream Auth Request

Not HDCP
Capable

Fail
Authentication

P1:
Transmit Low-
value Content

Receiver Connected
Indication from
connected HDCP 2.0-compliant Repeater

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. At any time a Receiver Connected Indication received from the connected HDCP 2.0-compliant HDCP Repeater causes the downstream side to transition in to this state.

**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. This step may be defined as part of the setup and discovery procedures and is outside the scope of this specification. 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 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 if low value content is 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 sending AKE_Init message containing $r_x$ to the HDCP Receiver and sends AKE_Transmitter_Info message to the HDCP Receiver. It receives AKE_Send_Cert from the receiver containing REPEATER and $cert_r$, and AKE_Receiver_Info message (if the HDCP Receiver is not HDCP 2.0-compliant). If the downstream side does not receive AKE_Receiver_Info message within 100 ms of the transmission of AKE_Transmitter_Info message, it indicates that the HDCP Receiver is an HDCP 2.0-compliant Device.

If the downstream side does not have $k_m$ stored corresponding to the Receiver ID, it generates $E_{k_{pub}}(km)$ and sends $E_{k_{pub}}(km)$ as part of the AKE_No_Stored_km message to the receiver after verification of signature on $cert_r$. It receives AKE_Send_rrx message containing $r_x$ from the receiver. It computes H, receives AKE_Send_H_prime message from the receiver containing $H'$ within one second after sending AKE_No_Stored_km to the receiver 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 $E_{kh}(km)$ and $m$ to the receiver and receives $r_x$ as part of AKE_Send_rrx message from the receiver. It computes H, receives AKE_Send_H_prime message from the receiver containing $H'$ within 200 ms after sending AKE_Stored_km to the receiver 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 $cert_r$ or if there is a mismatch between H and $H'$. This transition also occurs if AKE_Send_H_prime message is not received within one second after sending AKE_No_Stored_km or within 200 ms after sending AKE_Stored_km to the receiver.

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 $L'$ (or the most significant 128-bits of $L'$) is not received within 7 ms and the watchdog timer at the downstream side expires or on a mismatch between $L$ and $L'$ (or the most significant 128-bits of $L'$).
**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, \( E_{\text{dec}}(k_s) \), and \( r_v \) 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 is ‘false’ (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 Synchronization is performed to maintain cipher synchronization between the downstream side and the HDCP Receiver.

**Transition F4:F6.** REPEATER is ‘true’ (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 the RepeaterAuth_Send_ReceiverID_List is received.

**Transition F6:F7.** RepeaterAuth_Send_ReceiverID_List message is received.

**State F7: Verify Receiver ID List.** If a transition in to this state occurs from State F6, the watchdog timer is cleared. If both MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED are not ‘true’, computes \( V \). If the connected HDCP Repeater is HDCP 2.0-compliant, compares \( V \) and \( V' \). If the connected HDCP Repeater is not HDCP 2.0-compliant, compares the most significant 128-bits of \( V \) and \( V' \). The Receiver IDs 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 \( V \) and \( V' \) (if the connected HDCP Repeater is HDCP 2.0-compliant) or the most significant 128-bits of \( V \) and \( V' \) (if the connected HDCP Repeater is not HDCP 2.0-compliant). This transition is also made if the downstream side detects a roll-over of \( \text{seq\_num}_V \) (if the repeater is not HDCP 2.0-compliant). A MAX_CASCADE_EXCEEDED or MAX_DEVS_EXCEEDED error also causes this transition.

**Transition F7:F5.** This transition is made if the connected HDCP Repeater is HDCP 2.0-compliant, on successful verification of \( V \) and \( V' \) and the downstream topology does not exceed specified maximums.

**Transition F7:F8.** This transition occurs if the connected HDCP Repeater is not HDCP 2.0-compliant, on successful verification of the most significant 128-bits of \( V \) and \( V' \), the downstream side does not detect a roll-over of \( \text{seq\_num}_V \) 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 HDCP Repeater as part of the RepeaterAuth_Send_Ack message.
The RepeaterAuth_Send_Ack message must be received by the HDCP Repeater within one second from the transmission of the RepeaterAuth_Send_ReceiverID_List message to the downstream side.

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

**Transition F5:P1.** This transition occurs if a Receiver_AuthStatus message with the REAUTH_REQ set to ‘true’ is received.

**Transition F5:F7.** This transition occurs whenever a RepeaterAuth_Send_ReceiverID_List message is received from the connected HDCP Repeater that is not HDCP 2.0-compliant.

**State F9: Content Stream Management.** This stage is implemented if Content Stream is to be transmitted and the connected HDCP Repeater is not HDCP 2.0-compliant. 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 100ms 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 100 ms after the transmission of RepeaterAuth_Stream_Manage message and verifies $M'$. This step fails if the RepeaterAuth_Stream_Ready message is not received within 100 ms 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 Synchronization (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 connected HDCP Repeater is not HDCP 2.0-compliant 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.

Note: Type 1 Content Streams must not be transmitted by the downstream side through its HDCP-protected Interface Ports connected to HDCP 1.x-compliant Devices and HDCP 2.0-compliant Repeaters.

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

**Transition F9:P1.** This transition occurs if $\text{seq\_num\_M}$ rolls over.

Note: Since Link Synchronization 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 synchronization process (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.

![HDCP Repeater Upstream State Diagram](image)

**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 $r_{tx}$ from the HDCP Transmitter to trigger the authentication protocol.

**Transition C0:C1.** $r_{tx}$ is received as part of the AKE_Init message from the HDCP Transmitter.

**State C1: Compute $k_m$.** In this state, the upstream (HDCP Receiver) side sends AKE_Send_Cert message in response to AKE_Init, sends AKE_Receiver_Info message to the transmitter if AKE_Transmitter_Info message is received from the transmitter, generates and sends $r_x$ as part of AKE_Send_rxx message. If AKE_No_Stored_km is received, it decrypts $k_m$ with $k_{priv_{rx}}$ calculates $H'$. It sends AKE_Send_H_prime immediately after computation of $H'$ to ensure that the message is received by the transmitter within the specified one second timeout at the transmitter.

If the upstream side does not receive AKE_Transmitter_Info message before the reception of AKE_No_Stored_km or AKE_Stored_km message, it indicates that the HDCP Transmitter is an HDCP 2.0-compliant Device.
If AKE_Stored_km is received, the upstream side decrypts $E_{id}(km)$ to derive $km$ and calculates $H'$. It sends AKE_Send_H_prime message immediately after computation of $H'$ to ensure that the message is received by the transmitter within the specified 200 ms timeout at the transmitter.

If AKE_No_Stored_km is received, this is an indication to the upstream side that the HDCP Transmitter does not contain a $km$ 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 $k_e$.** The upstream side decrypts $E_{dkey}(k_e)$ to derive $k_e$.

**Transition C3: C4.** Successful computation of $k_e$ 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 sends RepeaterAuth_Send_ReceiverID_List message to the upstream HDCP Transmitter. In the case of a MAX_DEVS_EXCEEDED or a MAXCASCADE_EXCEEDED error, it does not transmit $V'$ (or the most significant 128-bits of $V'$), DEPTH, DEVICE_COUNT, Receiver ID list and if applicable, HDCP2_LEGACYDEVICE_DOWNSTREAM and HDCP1DEVICE_DOWNSTREAM. When an HDCP Repeater receives a MAXDEVS_EXCEEDED or MAXCASCADE_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 receives HDCP2_LEGACYDEVICE_DOWNSTREAM = ‘true’ or HDCP1DEVICE_DOWNSTREAM = ‘true’, the upstream side sets the corresponding values to ‘true’ in the RepeaterAuth_Send_ReceiverID_List message to the upstream HDCP Transmitter.
Transition C5:C0. This transition occurs if RepeaterAuth_Send_ReceiverID_List message has been sent to the upstream HDCP Transmitter and topology maximums are exceeded i.e. on a MAX_DEVS_EXCEEDED or MAX.Cascade.EXCEEDED error. This transition also occurs 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 sent to the upstream HDCP Transmitter and topology maximums are not exceeded and upstream transmitter is not HDCP 2.0-compliant.

Transition C5:C8. RepeaterAuth_Send_ReceiverID_List message has been sent to the upstream HDCP Transmitter and topology maximums are not exceeded and upstream transmitter is HDCP 2.0-compliant.

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 RepeaterAuth_Send_Ack message must be received by the upstream side within one second from the transmission of the RepeaterAuth_Send_ReceiverID_List message to the upstream transmitter if the transmitter is not HDCP 2.0-compliant.

Transition C6:C0. This transition occurs if the RepeaterAuth_Send_Ack message is not received by the upstream side within one second or on a mismatch between the least significant 128-bits of $V$ and $V'$. If this transition occurs, the upstream side must send the Receiver_AuthStatus message with the REAUTH_REQ set to ‘true’ to the upstream transmitter.

Transition C6:C7. This transition occurs if the RepeaterAuth_Send_Ack message is received by the upstream side within one second, 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 one second and on a successful match between the least significant 128-bits of $V$ and $V'$.

State C7: Content Stream Management. On receiving the RepeaterAuth_Stream_Manage message, the upstream side computes $M'$ and sends it to the upstream Transmitter 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 synchronization (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 synchronization if these stages are implemented in parallel by the upstream transmitter.

Transition C7:C8. This transition occurs after RepeaterAuth_Stream_Ready message has been sent to the upstream transmitter.
**State C8: Authenticated.** The upstream side has completed the authentication protocol. Periodically, it updates its `inputCtr` corresponding to the elementary stream (as indicated by the `streamCtr` value) with the `inputCtr` value received from the transmitter.

**Transition C8:C5.** This transition occurs only if the upstream HDCP Transmitter is not HDCP 2.0-compliant and 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 also occurs when a downstream port that was previously in an authenticated state transitions in to an unauthenticated on unconnected state. For example, the transition may occur when an active, authenticated HDCP Receiver attached to the downstream port is disconnected.

Reception of a RepeaterAuth_Send_ReceiverID_List message on a downstream port from the connected downstream HDCP Repeater also causes this transition.

**Transition C8:C0.** This transition occurs only if the upstream HDCP Transmitter is HDCP 2.0-compliant and 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.

Reception of a RepeaterAuth_Send_ReceiverID_List message on a downstream port from the connected downstream HDCP Repeater also causes this transition.

If this transition occurs, the upstream side must propagate a Receiver Connected Indication to the upstream HDCP Transmitter.

Note: Since Link Synchronization 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 synchronization process (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 synchronization process 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 $Bksam$ 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 $Bksam$ of the connected HDCP Repeater itself, to be added to the list. KSVs are used in place of Receiver IDs 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 sends RepeaterAuth_Send_ReceiverID_List message to the upstream HDCP Transmitter. In the case of a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED error, it does not transmit $V'$ (or the most significant 128-bits of $V'$), DEPTH, DEVICE_COUNT, Receiver ID list and if applicable, HDCP2_LEGACY_DEVICE_DOWNSTREAM and HDCP1DEVICE_DOWNSTREAM. 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](image)

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Max Delay</th>
<th>Conditions and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKE_Send_Eks1 Session Key received from Upstream HDCP Transmitter</td>
<td>AKSV1 HDCP Repeater’s $Aksam$ transmitted downstream</td>
<td>100 ms</td>
<td>Downstream propagation time.</td>
</tr>
<tr>
<td>AKSV1 HDCP Repeater’s $Aksam$ transmitted downstream</td>
<td>RepeaterAuth_Send_ReceiverID_List $Receivers ID$s and topology information transmitted upstream</td>
<td>200 ms</td>
<td>Upstream propagation time when no downstream HDCP Repeaters are attached (no downstream KSV lists to process)</td>
</tr>
</tbody>
</table>

| 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

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Max Delay</th>
<th>Conditions and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKE_Send_Eks1</td>
<td>AKSV1</td>
<td>100 ms</td>
<td>Downstream propagation time.</td>
</tr>
<tr>
<td>Session Key received from</td>
<td>HDCP Repeater’s Aksv</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upstream HDCP Transmitter</td>
<td>transmitted downstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDY1</td>
<td>ReceiverAuth_Send_ReceiverID_List1</td>
<td>200 ms</td>
<td>Upstream propagation time when one or more HDCP 1.x-compliant Repeaters are attached. Downstream KSV lists must be processed.</td>
</tr>
<tr>
<td>Downstream Receiver IDs and</td>
<td>Receiver IDs and topology</td>
<td></td>
<td></td>
</tr>
<tr>
<td>topology information</td>
<td>transmitted upstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>received</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

An HDCP1.x-HDCP2 Converter, that receives a DEVICE_COUNT value equal to 1 and DEPTH equal to 3 from its downstream port, must set DEVICE_COUNT value equal to 1 and DEPTH equal to 7 on its upstream port for transmission to the HDCP1.x-compliant upstream transmitter.

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. If both MAX_DEVS_EXCEEDED and MAX_CASCADE_EXCEEDED are not ‘true’, computes $V$. If the connected HDCP Repeater is HDCP 2.0-compliant, compares $V$ and $V'$. If the connected HDCP Repeater is not HDCP 2.0-compliant, compares the most significant 128-bits of $V$ and $V'$. The Receiver IDs 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 IDs 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

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Max Delay</th>
<th>Conditions and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKSV1</td>
<td>SKE_Send_Eks1</td>
<td>400 ms</td>
<td>Downstream propagation time.</td>
</tr>
<tr>
<td>Upstream HDCP Transmitter</td>
<td>Aksv received</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>k_s generated by HDCP Repeater transmitted downstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SKE_Send_Eks1</td>
<td>RDY1</td>
<td>500 ms</td>
<td>Upstream propagation time when no downstream HDCP Repeaters are attached. No downstream Receiver ID lists to process.</td>
</tr>
<tr>
<td>k_s generated by HDCP Repeater transitted downstream</td>
<td>Upstream READY asserted</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Max Delay</th>
<th>Conditions and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKSV1</td>
<td>SKE_Send_Eks1</td>
<td>400 ms</td>
<td>Downstream propagation time.</td>
</tr>
<tr>
<td>Upstream HDCP Transmitter</td>
<td>Aksv received</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>k_s generated by HDCP Repeater transmitted downstream</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RepeaterAuth_Send_ReceiverID_List</td>
<td>RDY1</td>
<td>500 ms</td>
<td>Upstream propagation time when one or more HDCP Repeaters are attached. From latest downstream RepeaterAuth_Send_ReceiverID_List message. (downstream Receiver ID lists must be processed)</td>
</tr>
<tr>
<td>Downstream Receiver IDs and topology information received</td>
<td>Upstream READY asserted</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 (Section 3.4) and stops incrementing the \textit{inputCnt}.

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 by the transmission of a new \textit{rtx}. 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 (associated with elementary streams) used during the HDCP Session i.e. \textit{inputCnt} value after the last HDCP Encryption operation (after which HDCP Encryption was disabled), \textit{streamCnt}, \textit{k}, and \textit{r}. When encryption is re-enabled, HDCP Encryption may be applied seamlessly, without requiring re-authentication, by using the same encryption parameters.

If HDCP Encryption was disabled, from its enabled state, the HDCP Receiver must maintain \textit{k} and \textit{r} 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 \textit{k} and \textit{r}. It must update its \textit{inputCnt} corresponding to the elementary stream (as indicated by the \textit{streamCnt} value) with the \textit{inputCnt} value received from the transmitter. (See Section 2.6 on Link Synchronization).

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. \textit{rt}, \textit{r}, \textit{r}, \textit{r}, \textit{r}) 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 \textit{rt}, \textit{r}, \textit{r}, \textit{r}, \textit{r} 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 (\textit{km}, \textit{k}) 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 \textit{km} or \textit{k} 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.
3 HDCP Encryption

3.1 Description

Figure 3.1 shows how HDCP fits in to the protocol stack. The link consists of two constituent links: a unidirectional high-speed stream transporting the AV Content, and a lower-speed bidirectional link used for control / status.

Figure 3.1. Transport Protocol w. HDCP Block Diagram (Informative)

Video in the HDCP Transmitter, together with any associated audio or data streams, are carried as MPEG Packetized Elementary Streams (PES), as specified in [3]. Each PES stream is encrypted as specified in Section 3.4. One or more PES streams, together with timing and any other ancillary information, may be multiplexed using MPEG Transport Stream (MPEG-TS) or MPEG Program Stream (MPEG-PS) mechanisms. The multiplexed stream may be encapsulated using a mechanism such as, in the case of an IP connection, RTP [8] as described in [6]. Control and Status messages are transported over a reliable bidirectional mechanism, e.g., as specified in [5] and [9].

3.2 AV Stream

MPEG AV streams consist of Packetized Elementary Streams (PES). Associated PES streams are grouped into a Program. Aside from the AV streams, various control, status and timing and
formatting information is also transported. Only the AV streams are subject to HDCP Encryption.

Note: A PES Stream may contain audio or video elements encoded by one of the standard codecs enumerated in [3], or it may contain non-standard codec data using the procedures in [3] for private stream data.

### 3.3 Abbreviations

- bslbf – as defined in [3].
- uimsbf – as defined in [3].
- byte – a digital word 8 bits in length.

### 3.4 HDCP Cipher

The HDCP cipher consists of a 128-bit AES module that is operated in a Counter (CTR) mode as illustrated in Figure 3.2.

\[
p = (r_y \text{ XOR } \text{streamCtr}) || \text{inputCtr}
\]

\(k_s\) is the 128-bit Session Key which is XORed with \(l_c_{128}\). Elementary streams within a given program or across multiple programs may use the same \(k_s\) and \(r_y\).

**Figure 3.2. HDCP Cipher Structure**

\(\text{streamCtr}\) is a 32-bit counter. The HDCP Transmitter assigns a distinct \(\text{streamCtr}\) value for each PES. The \(\text{streamCtr}\) value is distinct for elementary streams within a given program and across multiple programs i.e. no two elementary streams contained in a given program or different programs can have the same \(\text{streamCtr}\) if those elementary streams share the same \(k_s\) and \(r_y\).
HDCP Transmitter assigns \textit{streamCtr} values for video and audio portions of Audiovisual Content as per the following guidelines. The HDCP Transmitter assigns \textit{streamCtr} values where the least significant bit is zero to the video PES. It assigns \textit{streamCtr} values where the least significant bit is one to the audio PES. \textit{streamCtr} is initialized to zero after SKE and it must not be reset at any other time. It is XORed with the least significant 32-bits of \textit{riv}. If AKE_Transmitter_Info.TRANSMITTER_CONTENTCATEGORY_SUPPORT bit is set, the HDCP Receiver must verify that the assignment of \textit{streamCtr} complies with the guidelines described above. The HDCP Receiver must not decrypt the received HDCP Content if the \textit{streamCtr} assignment does not comply with the guidelines.

\textit{inputCtr} is a 64-bit counter. It is initialized to zero after SKE and must not be reset at any other time. Each elementary stream within a given program is associated with its own \textit{inputCtr}.

HDCP Encryption must be applied to PES payload data; PES Headers must not be encrypted.

During HDCP Encryption, the key stream produced by the AES-CTR module is XORed with 128-bit (16 Byte) block of payload data to produce the 128-bit encrypted output. \textit{inputCtr} associated with an elementary stream is incremented by one following encryption of 128-bit block of payload data for that stream. The value of \textit{inputCtr} must never be reused for a given set of encryption parameters i.e. \textit{ks}, \textit{riv}, and \textit{streamCtr}.

The 16 Byte encryption block boundary must be aligned with the start of the PES payload (if present) in each PES packet. If size of the encrypted contents of the PES payload is not an integer multiple of 16 Bytes, the unused key stream bits produced by the AES-CTR module must be discarded, and not carried over to a subsequent PES packet.

Bit ordering is such that the most-significant bit of the 128-bit key stream produced by AES-CTR module is XORed with the first bit in time (as defined in [3]) in the 16 Byte payload data block.

Any PES packet containing an HDCP encrypted payload must include the 128-bit PES\_private\_data field in its header. It must contain the value of \textit{streamCtr} for that stream, and the value of \textit{inputCtr} used to encrypt the first 16 Byte block of the PES payload, as shown in \textbf{Error! Reference source not found.} Table 3.1.
Syntax No. of Bits Identifier

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bits</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>PES_private_data()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reserved_bits</td>
<td>13</td>
<td>bslbf</td>
</tr>
<tr>
<td>streamC0r[31..30]</td>
<td>2</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>streamC0r[29..15]</td>
<td>15</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>streamC0r[14..0]</td>
<td>15</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>reserved_bits</td>
<td>11</td>
<td>bslbf</td>
</tr>
<tr>
<td>inputC0r[63..60]</td>
<td>4</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>inputC0r[59..45]</td>
<td>15</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>inputC0r[44..30]</td>
<td>15</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>inputC0r[29..15]</td>
<td>15</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
<tr>
<td>inputC0r[14..0]</td>
<td>15</td>
<td>bslbf</td>
</tr>
<tr>
<td>marker_bit</td>
<td>1</td>
<td>bslbf</td>
</tr>
</tbody>
</table>

Table 3.1. PES Header HDCP Private Data

Marker bits have a value of ‘1’. All bits in the reserved_bits field have a value of ‘0’.

Note:

- Marker bits serve to prevent packet_start_code emulation, and are used here in a form similar to their use in other PES header fields (e.g., PTS).

- The presence of the PES Header HDCP Private Data block, as shown in Table 3.1, serves to indicate that HDCP Encryption is enabled and the PES payload is encrypted. When HDCP Encryption is disabled, the PES Header HDCP Private Data block, as shown in Table 3.1, is not included.

- HDCP does not use the PES_scrambling_control bits.

3.5 HDCP Cipher Block

The HDCP cipher block consists of multiple HDCP cipher (AES-CTR) modules. The input encryption parameters to each HDCP cipher module satisfy the requirements in Section 3.4 i.e. the streamC0r value is distinct for each PES within an HDCP Cipher Block, an inputC0r is associated with each elementary stream, the same ks and riv is used for encryption of all elementary streams within an HDCP Cipher Block.

Figure 3.3 illustrates an HDCP cipher block used for encryption of multiple programs.
This section defines procedures used when MPEG System multiplexing (MPEG-TS or MPEG-PS) is used.

### 3.6.1 HDCP Registration Descriptor

For MPEG System multiplexing (MPEG-TS or MPEG-PS), streams subject to HDCP Encryption must include a registration descriptor of the form shown in Table 3.2. It serves to indicate that the private data in the PES header (see Table 3.1) is defined by this document.

The inclusion of the HDCP Registration Descriptor is required only when MPEG-TS or MPEG-PS is used. The HDCP Transmitter must not include the registration descriptor unless it determines that the receiver is HDCP-capable.
Table 3.2. HDCP Registration Descriptor

The descriptor_length must be equal to 5, with one additional_identification_info (namely, HDCP_version) equal to 0x20.

3.6.2 Transport Stream

MPEG Transport Streams may contain multiple programs. Each program subject to HDCP Encryption must include the registration descriptor defined in Section 3.6.1 in the program loop (i.e., the “outer loop”) of its PMT.

For Transport Stream, the TS headers and Adaptation fields must not be encrypted. Payload data for PIDs containing control, status, management information (e.g., PAT and PMT data) must not be encrypted. For Transport Stream, the adaptation field must be padded such that the payload (excluding the PES header, if any) is an integral multiple of 16 Bytes.

A complete AKE, Locality Check and SKE procedure is performed on one program, prior to enabling HDCP Encryption for any program. The same $k_s$ and $r_{iv}$ is used for all programs. Encryption may be enabled and disabled separately for each program that includes the HDCP registration descriptor in its PMT, and for PES stream within those programs.

For Transport Stream, a PES header must not exceed 184 bytes, and the Adaptation Field must not be so long as to cause the PES header to extend into the next TS packet. The 16 Byte encryption block boundary must be aligned with the start of the PES payload (if present) in each TS packet. If the last block in the encrypted TS packet is less than 16 Bytes, only the encrypted payload bytes must be transmitted; i.e., the unused key stream bits produced by the AES-CTR module must be discarded, and not carried over to a subsequent packet.

Note: This constraint simplifies packet assembly and parsing.

Note: For Transport Stream, only in the TS packet containing the end of the PES packet does the possibility exist that the last block in the packet might be less than 16 Bytes.

3.6.3 Program Stream

MPEG Program Streams contain a single program. Each program subject to HDCP Encryption must include the registration descriptor defined in Section 3.6.1 in the program info loop (i.e., the “outer loop”) of its PSM.

The Pack Header and PSM packets of the Program Stream must not be encrypted.

3.7 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 $r_{iv}$ used during each HDCP Session is required to be fresh, transmitters with multiple downstream HDCP links must ensure that each link receives a distinct $r_{iv}$ value.

As illustrated in Figure 3.4, 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 block is transmitted to those links.
As illustrated in Figure 3.5, HDCP Transmitters, including downstream side of HDCP Repeaters, with multiple downstream HDCP links must ensure that each link receives distinct $k_s$ and $r_{iv}$ values if HDCP Content from different HDCP cipher blocks is transmitted to those links.
Figure 3.5. Unique $k_s$ and $r_v$ across HDCP Links
4 Authentication Protocol Messages

4.1 Abbreviations
bslf – as defined in [3].

uimbslf – as defined in [3].

byte – a digital word 8 bits in length.

uint – unsigned integer, an integral number of bytes in length.

bool – a parameter one byte in length. The parameter is ‘true’ if the least-significant bit is non-zero, and false otherwise.

4.2 Control / Status Stream
Each Control/Status message begins with a msg_id field. Valid values of msg_id are shown in Table 4.1.

<table>
<thead>
<tr>
<th>Message Type</th>
<th>msg_id Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null message</td>
<td>1</td>
</tr>
<tr>
<td>AKE_Init</td>
<td>2</td>
</tr>
<tr>
<td>AKE_Send_Cert</td>
<td>3</td>
</tr>
<tr>
<td>AKE_No_Stored_km</td>
<td>4</td>
</tr>
<tr>
<td>AKE_Stored_km</td>
<td>5</td>
</tr>
<tr>
<td>AKE_Send_rrx</td>
<td>6</td>
</tr>
<tr>
<td>AKE_Send_H_prime</td>
<td>7</td>
</tr>
<tr>
<td>AKE_Send_Pairing_Info</td>
<td>8</td>
</tr>
<tr>
<td>LC_Init</td>
<td>9</td>
</tr>
<tr>
<td>LC_Send_L_prime</td>
<td>10</td>
</tr>
<tr>
<td>SKE_Send_Eks</td>
<td>11</td>
</tr>
<tr>
<td>RepeaterAuth_Send_ReceiverID_List</td>
<td>12</td>
</tr>
<tr>
<td>RTT_Ready</td>
<td>13</td>
</tr>
<tr>
<td>RTT_Challenge</td>
<td>14</td>
</tr>
<tr>
<td>RepeaterAuth_Send_Ack</td>
<td>15</td>
</tr>
<tr>
<td>RepeaterAuth_Stream_Manage</td>
<td>16</td>
</tr>
<tr>
<td>RepeaterAuth_Stream_Ready</td>
<td>17</td>
</tr>
<tr>
<td>Receiver_AuthStatus</td>
<td>18</td>
</tr>
<tr>
<td>AKE_Transmitter_Info</td>
<td>19</td>
</tr>
<tr>
<td>AKE_Receiver_Info</td>
<td>20</td>
</tr>
<tr>
<td>Reserved</td>
<td>21-31</td>
</tr>
</tbody>
</table>

Table 4.1. Values for msg_id

A reliable, bidirectional packet protocol (e.g., TCP/IP) is used to transport messages used for the HDCP authentication protocol from the HDCP Transmitter to the HDCP Receiver, and vice versa.

Each packet must contain exactly one message. Each packet payload commences with a msg_id specifying the message type, followed by parameters specific to each message.

In the case of TCP/IP, packets use an IP address and port number determined by procedures above the HDCP layer. Also, parameter values spanning more than one byte follow the convention in [5] of sending the most-significant byte first.

Note:
• The use of the Null message and Reserved values for msg_id are not defined in this specification. HDCP Devices must be capable of receiving Null message and messages with reserved msg_id values and must ignore these messages.

4.3 Message Format

4.3.1 AKE_Init (Transmitter to Receiver)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_Init { msg_id ( r_i[65..0] ) }</td>
<td>1 ( r_i[65..0] )</td>
<td>( r_i[65..0] )</td>
</tr>
</tbody>
</table>

Table 4.2. AKE_Init Payload

4.3.2 AKE_Send_Cert (Receiver to Transmitter)

The HDCP Receiver sets REPEATER to ‘true’ if it is an HDCP Repeater and ‘false’ if it is an HDCP Receiver that is not an HDCP Repeater. When REPEATER = ‘true’, the HDCP Receiver supports downstream connections as permitted by the Digital Content Protection LLC license.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_Send_Cert { msg_id REPEATER</td>
<td>cert_r[4175..0] }</td>
<td>1 ( \text{msg_id} )</td>
</tr>
</tbody>
</table>

Table 4.3. AKE_Send_Cert Payload

4.3.3 AKE_No_Stored_km (Transmitter to Receiver)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_No_Stored_km { msg_id ( E_{\text{auth}}.k_m[1023..0] ) }</td>
<td>1 ( \text{msg_id} )</td>
<td>128</td>
</tr>
</tbody>
</table>

Table 4.4. AKE_No_Stored_km Payload

4.3.4 AKE_Stored_km (Transmitter to Receiver)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_Stored_km { msg_id ( E_{\text{auth}}.k_m[127..0] ) m[127..0] }</td>
<td>1 ( \text{msg_id} )</td>
<td>16 ( \text{m}[127..0] )</td>
</tr>
</tbody>
</table>

Table 4.5. AKE_Stored_km Payload
4.3.5 AKE_Send_rrx (Receiver to Transmitter)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_Send_rrx{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>(r_{rx}[63..0])</td>
<td>8</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6. AKE_Send_rrx Payload

4.3.6 AKE_Send_H_prime (Receiver to Transmitter)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AK_Send_H_prime{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>(H[255..0])</td>
<td>32</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7. AKE_Send_H_prime Payload

4.3.7 AKE_Send_Pairing_Info (Receiver to Transmitter)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_Send_Pairing_Info{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>(E_{kh,_km}[127..0])</td>
<td>16</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.8. AKE_Send_Pairing_Info Payload

4.3.8 LC_Init (Transmitter to Receiver)

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC_Init{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>(r_{n}[63..0])</td>
<td>8</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.9. LC_Init Payload

4.3.9 LC_Send_L_prime (Receiver to Transmitter)

If the upstream HDCP Transmitter is HDCP 2.0-compliant or the \(\text{TRANSMITTER\_LOCALITY\_PRECOMPUTE\_SUPPORT}\) bit received as part of the AKE_Transmitter_Info message is set to zero or the receiver has set the \(\text{RECEIVER\_LOCALITY\_PRECOMPUTE\_SUPPORT}\) bit to zero in its AKE_Receiver_Info message, the HDCP Receiver constructs the LC_Send_L_prime message as given below.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC_Send_L_prime{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>(L[255..0])</td>
<td>32</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.10. LC_Send_L_prime Payload
If the TRANSMITTER_LOCALITY_PRECOMPUTE_SUPPORT bit received as part of the AKE_Transmitter_Info message is set to one and the receiver has set the RECEIVER_LOCALITY_PRECOMPUTE_SUPPORT bit to one in its AKE_Receiver_Info message, the HDCP Receiver constructs the LC_Send_L_prime message as given below.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC_Send_L_prime{</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>msg_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L[255..128]</td>
<td>16</td>
<td>uint</td>
</tr>
</tbody>
</table>

Table 4.11. LC_Send_L_prime Payload

4.3.10 SKE_Send_Eks (Transmitter to Receiver)

The HDCP Transmitter constructs the SKE_Send_Eks message as given in Table 4.12 if the HDCP Receiver is compliant with HDCP2.3 or higher, else the HDCP Transmitter constructs the message as given in Table 4.13.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SKE_Send_Eks{</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>msg_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edkey_k[127..0]</td>
<td>16</td>
<td>uint</td>
</tr>
<tr>
<td>r0[63..0]</td>
<td>8</td>
<td>unit</td>
</tr>
<tr>
<td>HMAC(r0)[255..0]</td>
<td>32</td>
<td>uint</td>
</tr>
</tbody>
</table>

Table 4.12. SKE_Send_Eks Payload

4.3.11 RepeaterAuth_Send_ReceiverID_List (Receiver to Transmitter)

The HDCP Repeater constructs the RepeaterAuth_Send_ReceiverID_List message as given in Table 4.14 if the upstream HDCP Transmitter is an HDCP 2.0-compliant Device, else the HDCP Repeater constructs the message as given in Table 4.15.

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

Receiver ID list = Receiver ID₀ || Receiver ID₁ || ... || Receiver IDₙ₋₁, where n is the DEVICE_COUNT.

If the computed DEVICE_COUNT for an HDCP Repeater exceeds 31, the repeater sets MAX_DEVS_EXCEEDED = ‘true’. If the computed DEPTH for an HDCP Repeater exceeds four, the repeater sets MAX_CASCADE_EXCEEDED = ‘true’. If topology maximums are not exceeded, MAX_DEVS_EXCEEDED = ‘false’ and MAX_CASCADE_EXCEEDED = ‘false’.
The HDCP Repeater sets HDCP2_LEGACY_DEVICE_DOWNSTREAM = ‘true’ 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 HDCP2_LEGACY_DEVICE_DOWNSTREAM = ‘false’.

The HDCP Repeater sets HDCP1DEVICE_DOWNSTREAM = ‘true’ 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 ports, else it sets HDCP1DEVICE_DOWNSTREAM = ‘false’.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepeaterAuth_Send_ReceiverID_List{</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>MAX_DEVS_EXCEEDED</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>MAX_CASCADE_EXCEEDED</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>if (MAX_DEVS_EXCEEDED != 1 &amp;&amp; MAX_CASCADE_EXCEEDED != 1) {</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>DEVICE_COUNT</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>DEPTH</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>$V^{(255..0)}</td>
<td>32</td>
<td>uint</td>
</tr>
<tr>
<td>for (j=0; j&lt;DEVICE_COUNT; j++) {</td>
<td>5</td>
<td>uint</td>
</tr>
<tr>
<td>Receiver_ID$^{(39..0)}</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.14. RepeaterAuth_Send_ReceiverID_List Payload

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepeaterAuth_Send_ReceiverID_List{</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>MAX_DEVS_EXCEEDED</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>MAX_CASCADE_EXCEEDED</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>if (MAX_DEVS_EXCEEDED != 1 &amp;&amp; MAX_CASCADE_EXCEEDED != 1) {</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>DEVICE_COUNT</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>DEPTH</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>HDCP2_LEGACYDEVICE_DOWNSTREAM</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>HDCP1DEVICE_DOWNSTREAM</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>seq_num$^{(255..128)}</td>
<td>16</td>
<td>uint</td>
</tr>
<tr>
<td>for (j=0; j&lt;DEVICE_COUNT; j++) {</td>
<td>5</td>
<td>uint</td>
</tr>
<tr>
<td>Receiver_ID$^{(39..0)}</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.15. RepeaterAuth_Send_ReceiverID_List Payload
4.3.12 RTT_Ready (Receiver to Transmitter)

Syntax | No. of Bytes | Identifier
--- | --- | ---
RTT_Ready {  
  msg_id  
} | 1 | uint

Table 4.16. RTT_Ready Payload

4.3.13 RTT_Challenge (Transmitter to Receiver)

Syntax | No. of Bytes | Identifier
--- | --- | ---
RTT_Challenge{  
  msg_id  
  L[127..0]  
} | 1 | uint  
| 16 | uint

Table 4.17. RTT_Challenge Payload

4.3.14 RepeaterAuth_Send_Ack (Transmitter to Receiver)

Syntax | No. of Bytes | Identifier
--- | --- | ---
RepeaterAuth_Send_Ack{  
  msg_id  
  V[127..0]  
} | 1 | uint  
| 16 | uint

Table 4.18. RepeaterAuth_Send_Ack Payload

4.3.15 RepeaterAuth_Stream_Manage (Transmitter to Receiver)

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

Content Streams may be comprised of audio and video elementary streams. Each elementary stream is assigned a streamCtr value by the HDCP Transmitter which is used during HDCP Encryption of the elementary stream. The ContentStreamID, derived from the Packet Identifier (PID), for each elementary stream is associated with its corresponding streamCtr in the RepeaterAuth_Stream_Manage message.

Elementary streams, identified by the streamCtr value which is used during HDCP Encryption of the elementary stream, are assigned the same Type value that is assigned to the corresponding Content Stream by the HDCP Transmitter. All elementary streams transmitted by the HDCP Transmitter to the HDCP Repeater, after HDCP Encryption, are assigned Type values.

The streamCtr assigned to an elementary stream is followed by its corresponding ContentStreamID value and its assigned Type value in the RepeaterAuth_Stream_Manage message.
### Syntax No. of Bytes Identifier

| RepeaterAuth_Stream_Manage{                              | 1    | uint    |
| msg_id                                                  | 3    | uint    |
| seq_num_M                                               | 2    | uint    |
| k                                                       |      |         |
| for(j=0;j<k;j++) {                                      |      |         |
| streamCtrj                                             | 4    | uint    |
| ContentStreamIDj                                       | 2    | uint    |
| Type                                                   | 1    | uint    |
| }                                                       |      |         |

Table 4.19. RepeaterAuth_Stream_Manage Payload

STREAMID_TYPE = streamCtr || ContentStreamID || Type || streamCtr || ContentStreamID || Type ||...

streamCtr assigned to an elementary stream is concatenated with its corresponding ContentStreamID value and its assigned Type value. All values are in big-endian order.

**k** is the number of elementary streams that are being transmitted by the HDCP Transmitter to the attached HDCP Repeater during the HDCP Session.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>No. of Bytes</th>
<th>Description</th>
</tr>
</thead>
</table>
| ContentStreamID | 2            | Content Stream Identification. This parameter must be assigned the PID value as follows.  
|               |              | ContentStreamID[15:13] = 0                                                  |
|               |              | ContentStreamID[12:0] = PID                                                  |
| 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 |

Table 4.20. RepeaterAuth_Stream_Manage Parameters

### 4.3.16 RepeaterAuth_Stream_Ready (Receiver to Transmitter)

| RepeaterAuth_Stream_Ready{                              | 1    | uint    |
| msg_id                                                  | 32   | uint    |
| M[255..0]                                               |      |         |

Table 4.21. RepeaterAuth_Stream_Ready Payload

### 4.3.17 Receiver_AuthStatus (Receiver to Transmitter)

LENGTH parameter is the size of the Receiver_AuthStatus message in bytes. An HDCP 2.3-compliant Receiver will set the LENGTH parameter equal to four bytes i.e. the combined size of the msg_id, LENGTH and REAUTH_REQ parameters. An HDCP 2.3-compliant transmitter that receives a Receiver_AuthStatus message with the LENGTH parameter greater than four bytes
must read the msg_id, LENGTH and REAUTH_REQ parameters and must ignore the remaining parameters.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver_AuthStatus{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>LENGTH</td>
<td>2</td>
<td>uint</td>
</tr>
<tr>
<td>REAUTH_REQ</td>
<td>1</td>
<td>bool</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.22. Receiver_AuthStatus Payload

### 4.3.18 AKE_Transmitter_Info (Transmitter to Receiver)

LENGTH parameter is the size of the AKE_Transmitter_Info message in bytes. An HDCP 2.3-compliant Transmitter will set the LENGTH parameter equal to six bytes i.e. the combined size of the msg_id, LENGTH, VERSION and TRANSMITTER_CAPABILITY_MASK parameters. An HDCP 2.3-compliant Receiver that receives an AKE_Transmitter_Info message with the LENGTH parameter greater than six bytes must read the msg_id, LENGTH, VERSION and TRANSMITTER_CAPABILITY_MASK parameters and must ignore the remaining parameters.

The HDCP Transmitter must set VERSION to 0x03.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>No. of Bytes</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKE_Transmitter_Info{</td>
<td></td>
<td></td>
</tr>
<tr>
<td>msg_id</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>LENGTH</td>
<td>2</td>
<td>uint</td>
</tr>
<tr>
<td>VERSION</td>
<td>1</td>
<td>uint</td>
</tr>
<tr>
<td>TRANSMITTER_CAPABILITY_MASK</td>
<td>2</td>
<td>uint</td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.23. AKE_Transmitter_Info Payload

<table>
<thead>
<tr>
<th>Parameter</th>
<th>No. of Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSMITTER_CAPABILITY_MASK</td>
<td>2</td>
<td>Bits 15:2: Reserved zeros. Bit 1: TRANSMITTER_CONTENT_CATEGORY_SUPPORT. This bit must be set to 1 by the HDCP Transmitter. Bit 0: TRANSMITTER_LOCALITY_PRECOMPUTE_SUPPORT. When this bit is set to one, it indicates that the HDCP Transmitter supports pre-computation of L during the locality check protocol.</td>
</tr>
</tbody>
</table>

Table 4.24. TRANSMITTER_CAPABILITY_MASK Parameter

### 4.3.19 AKE_Receiver_Info (Receiver to Transmitter)

LENGTH parameter is the size of the AKE_Receiver_Info message in bytes. An HDCP 2.3-compliant Receiver will set the LENGTH parameter equal to six bytes i.e. the combined size of the msg_id, LENGTH, VERSION and RECEIVER_CAPABILITY_MASK parameters. An HDCP 2.3-compliant transmitter that receives an AKE_Receiver_Info message with the LENGTH parameter greater than six bytes must read the msg_id, LENGTH, VERSION and RECEIVER_CAPABILITY_MASK parameters and must ignore the remaining parameters.

The HDCP Receiver must set VERSION to 0x03.
### Table 4.25. AKE_Receiver_Info Payload

<table>
<thead>
<tr>
<th>Parameter</th>
<th>No. of Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECEIVER_CAPABILITY_MASK</td>
<td>2</td>
<td>Bits 15:1: Reserved zeros. Bit 0: RECEIVER_LOCALITY_PRECOMPUTE_SUPPORT. When this bit is set to one, it indicates that the HDCP Receiver supports pre-computation of $L'$ during the locality check protocol.</td>
</tr>
</tbody>
</table>

Table 4.26. RECEIVER_CAPABILITY_MASK Parameter
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 certrx. 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, kprivrx, 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

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (bits)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRM ID</td>
<td>4</td>
<td>A value of 0x9 signifies that the message is for HDCP 2. All other</td>
</tr>
</tbody>
</table>
values are reserved. SRMs with values other than 0x9 must be ignored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (bits)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDCP2 Indicator</td>
<td>4</td>
<td>A value of 0x1 signifies that the message is for HDCP2</td>
</tr>
<tr>
<td>Reserved</td>
<td>8</td>
<td>Reserved for future definition. Must be 0x00</td>
</tr>
<tr>
<td>SRM Version</td>
<td>16</td>
<td>Sequentially increasing unique SRM numbers. Higher numbered SRMs are more recent</td>
</tr>
<tr>
<td>SRM Generation</td>
<td>8</td>
<td>Indicates the generation of the SRM. The generation number starts at 1 and increases sequentially</td>
</tr>
<tr>
<td>Length</td>
<td>24</td>
<td>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</td>
</tr>
<tr>
<td>Number of Devices</td>
<td>10</td>
<td>Specifies the number (N1) of Receiver IDs / KSVs contained in the first-generation SRM</td>
</tr>
<tr>
<td>Reserved</td>
<td>22</td>
<td>Reserved for future definition. All bits set to 0</td>
</tr>
<tr>
<td>Device IDs</td>
<td>40 * N1</td>
<td>40-bit Receiver IDs / KSVs</td>
</tr>
<tr>
<td>Max size for this</td>
<td></td>
<td>Max size for this field is 37760 (4720 bytes)</td>
</tr>
<tr>
<td>field is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3072</td>
<td>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</td>
</tr>
</tbody>
</table>

Table 5.1. System Renewability Message Format

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Size (bits)</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>16</td>
<td>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</td>
</tr>
<tr>
<td>Reserved</td>
<td>6</td>
<td>Reserved for future definition. All bits set to 0</td>
</tr>
<tr>
<td>Number of Devices</td>
<td>10</td>
<td>Specifies the number (N2) of Receiver IDs / KSVs contained in this next-generation extension</td>
</tr>
<tr>
<td>Device IDs</td>
<td>40 * N2</td>
<td>40-bit Receiver IDs / KSVs</td>
</tr>
<tr>
<td>DCP LLC Signature</td>
<td>3072</td>
<td>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</td>
</tr>
</tbody>
</table>

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. 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 the version number of the new SRM is greater than the version number of the current 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Confidentiality Required(\hat{\dagger})?</th>
<th>Integrity Required(\hat{\dagger})?</th>
<th>Value used by Core Functions?</th>
<th>Core Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>lc128</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>HDCP Encryption and Decryption</td>
</tr>
<tr>
<td>kpubdcp</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>certrx</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>kpubrx</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>Receiver ID</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>kprivrx</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Handling of Device Secret Key, during AKE, in plaintext form</td>
</tr>
<tr>
<td>r_t</td>
<td>No</td>
<td>Yes(^*)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>r_r</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>REPEATER</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>r_rx</td>
<td>No</td>
<td>Yes(^{**})</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>k_m</td>
<td>Yes</td>
<td>Yes(^*)</td>
<td>Yes</td>
<td>Handling of Master Key, during AKE (including Pairing) and Key Derivation, in plaintext form</td>
</tr>
<tr>
<td>k_d</td>
<td>Yes</td>
<td>Yes(^*)</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>dkey0,dkey1</td>
<td>Yes</td>
<td>Yes(^*)</td>
<td>No</td>
<td>N/A</td>
</tr>
</tbody>
</table>

\(\dagger\) According to the robustness rules in the HDCP Adopter’s License

\(^*\) Only within the transmitter

\(^{**}\) Only within the receiver
<table>
<thead>
<tr>
<th>dkey2</th>
<th>Yes</th>
<th>Yes*</th>
<th>Yes</th>
<th>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</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctr</td>
<td>No</td>
<td>Yes*</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>H</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$H'$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$m$</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>$k_h$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>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</td>
</tr>
<tr>
<td>$r_n$</td>
<td>No</td>
<td>Yes*</td>
<td>Yes</td>
<td>N/A</td>
</tr>
<tr>
<td>L</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$L'$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$k_s$</td>
<td>Yes</td>
<td>Yes*</td>
<td>Yes</td>
<td>Handling of Session Key, during SKE and HDCP Encryption/Decryption, in plaintext form</td>
</tr>
<tr>
<td>$V[255:128]$</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$V'[127:0]$</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$V[127:0]$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$V'[255:128]$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$M$</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>$M'$</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>Receiver ID list</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>DEPTH</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>DEVICE_COUNT</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>MAX_DEVS_EXCEEDED</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>MAX_CASCADE_EXCEEDED</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>inputCtr</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>HDCP Encryption and Decryption</td>
</tr>
<tr>
<td>streamCtr</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>HDCP Encryption and Decryption</td>
</tr>
<tr>
<td>p</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>HDCP Encryption and Decryption</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value (hexadecimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus n</td>
<td>B0E9 AA45 F129 BA0A 1CBE 1757 26EB 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 4C8C 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 3E0A</td>
</tr>
<tr>
<td>Public Exponent e</td>
<td>03</td>
</tr>
</tbody>
</table>
Appendix C. Bibliography (Informative)

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


*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](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.

<table>
<thead>
<tr>
<th>Value in Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Constant lc128 93 ce 5a 56 a0 a1 f4 f7 3c 65 8a 1b d2 ae f0 f7</td>
</tr>
</tbody>
</table>

Table D.1

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

<table>
<thead>
<tr>
<th>Value in Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 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</td>
</tr>
</tbody>
</table>
Table D.2

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

As provided in Table 2.1 of High-bandwidth Digital Content Protection System, Revision 2.2, Interface Independent Adaptation specification, the certificate format consists of 40-bit Receiver ID, followed by 1048-bit Receiver Public Key, 4-bit Protocol Descriptor, 12-bit Reserved 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, Reserved and Signature fields.

<table>
<thead>
<tr>
<th>Value (Sequence of Hexadecimal bytes) for R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate ($\text{cert}_{\text{rx}}$)</td>
</tr>
<tr>
<td>74 5b b8 bd 04 af b5 c5 c6 7b c5 3a 34 90 a9</td>
</tr>
<tr>
<td>54 c0 8f b7 eb a1 54 d2 4f 22 de 83 f5 03 a6</td>
</tr>
<tr>
<td>c6 68 46 9b c0 b8 c8 6c db 26 f9 3c 49 2f 02</td>
</tr>
<tr>
<td>e1 71 df 4e f3 0e c8 bf 22 9d 04 cf bf a9 0d</td>
</tr>
<tr>
<td>ff 68 ab 05 6f 1f 12 8a 68 62 eb fe c9 ea 9f</td>
</tr>
<tr>
<td>a7 fb 8c ba b1 bd 65 ac 35 9c a0 33 b1 dd a6</td>
</tr>
<tr>
<td>05 36 af 00 a2 7f bc 07 b2 dd b5 cc 57 5c dc</td>
</tr>
<tr>
<td>c0 95 50 e5 ff 1f 20 db 59 46 fa 47 c4 ed 12</td>
</tr>
<tr>
<td>2e 9e 22 bd 95 a9 85 59 a1 59 3c c7 83 01 00</td>
</tr>
<tr>
<td>01 10 00 0b a3 73 77 dd 03 18 03 8a 91 63 29</td>
</tr>
<tr>
<td>1e a2 95 74 42 90 78 d0 67 25 b6 32 2f cc 23</td>
</tr>
<tr>
<td>2b ad 21 39 3d 14 ba 37 a3 65 14 6b 9c cf 61</td>
</tr>
<tr>
<td>20 44 a1 07 bb cf c3 4e 95 5b 10 cf c7 6f f1</td>
</tr>
<tr>
<td>c3 53 7c 63 a1 8c b2 e8 ab 2e 96 97 c3 83 99</td>
</tr>
<tr>
<td>70 d3 dc 21 41 f6 0a d1 1a ee f4 cc eb fb a6</td>
</tr>
<tr>
<td>aa b6 9a af 1d 16 5e e2 83 a0 4a 41 f6 7b 07</td>
</tr>
<tr>
<td>bf 47 85 28 6c a0 77 a6 a3 d7 85 a5 c4 a7 e7</td>
</tr>
<tr>
<td>6e b5 1f 40 72 97 fe c4 81 23 a0 c2 90 b3 49</td>
</tr>
<tr>
<td>24 f5 b7 90 2c bf fe 04 2e 00 a9 5f 86 04 ca</td>
</tr>
<tr>
<td>c5 3a cc 26 d9 39 7e a9 2d 28 6d c0 cc 6e 81</td>
</tr>
<tr>
<td>9f b9 b7 11 33 32 23 47 98 43 0d a5 1c 59 f3</td>
</tr>
<tr>
<td>cd d2 4a b7 3e 69 d9 21 53 9a f2 6e 77 62 ae</td>
</tr>
<tr>
<td>50 da 85 c6 aa c4 b5 1c cd a8 a5 dd 6e 62 73</td>
</tr>
<tr>
<td>ff 5f 7b d7 3c 17 ba 47 0c 89 0e 62 79 43 94</td>
</tr>
<tr>
<td>aa a8 47 f4 4c 38 89 a8 81 ad 23 13 27 0c 17</td>
</tr>
</tbody>
</table>
Table D.3

<table>
<thead>
<tr>
<th>Certificate (certRx)</th>
<th>Value (Sequence of Hexadecimal bytes) for R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>8b a4 47 42 fb e4 68 63 8a da 97 2d de 9a 8d 1c</td>
<td>8b a4 47 42 fb e4 68 63 8a da 97 2d de 9a 8d 1c</td>
</tr>
<tr>
<td>b1 65 4b 85 8d e5 46 d6 db 95 a5 f6 66 74 ea 81</td>
<td>b1 65 4b 85 8d e5 46 d6 db 95 a5 f6 66 74 ea 81</td>
</tr>
<tr>
<td>0b 9a 58 58 66 26 86 a6 b4 56 2b 29 43 e5 bb 81</td>
<td>0b 9a 58 58 66 26 86 a6 b4 56 2b 29 43 e5 bb 81</td>
</tr>
<tr>
<td>74 86 a7 b7 16 2f 07 ec d1 b5 f9 ae 4f 98 89 a9</td>
<td>74 86 a7 b7 16 2f 07 ec d1 b5 f9 ae 4f 98 89 a9</td>
</tr>
<tr>
<td>91 7d 58 5b 8d 20 d5 c5 08 40 3b 86 af f4 d6 b9</td>
<td>91 7d 58 5b 8d 20 d5 c5 08 40 3b 86 af f4 d6 b9</td>
</tr>
<tr>
<td>20 95 e8 90 3b 8f 9f 36 5b 46 b6 d4 1e f5 05 88</td>
<td>20 95 e8 90 3b 8f 9f 36 5b 46 b6 d4 1e f5 05 88</td>
</tr>
</tbody>
</table>

Table D.4

<table>
<thead>
<tr>
<th>Value in Hex for R1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>ec be e5 5b 9e 7a 50 8a 96 80 c8 db b0 ed 44 f2 ba 1d 5d 80 c1 c8</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Value in Hex for R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
</tr>
<tr>
<td>f5 f6 f3 a4 a2 16 2f a7 1f 7f 16 05 99 26 c4 1b 80 7f fa 52 4e 3e</td>
</tr>
<tr>
<td>a4 3d 1e b0 f1 9a c6 3d 8f 57 7b 9e cd e8 03 d6 f3 91 75 e2 19 44</td>
</tr>
<tr>
<td>9e 11 58 5f d6 88 7c c4 cl 5b 45 9b 84 cf 72 7d 35 5f 24 d5</td>
</tr>
<tr>
<td>q</td>
</tr>
<tr>
<td>ed ba 08 bf 42 2c 0e fa 3a c4 d2 c7 01 51 25 ae b0 a1 cc db 67 9b</td>
</tr>
<tr>
<td>aa 50 f0 80 ac 4b 9f 5c ba 1e f4 7f a9 b3 21 8a 62 2c 36 da cd a7</td>
</tr>
<tr>
<td>4d a4 d6 44 ed b1 3e 47 10 77 5a 6a f5 f3 63 8a 2c 43 09</td>
</tr>
<tr>
<td>d mod (p^-1)</td>
</tr>
<tr>
<td>61 5a c4 6c 6e 0b 82 09 10 3a 69 29 06 19 85 fd ac ba fb 05 a0 da</td>
</tr>
<tr>
<td>c4 df 3a 4a ad 16 a9 e8 ab d7 c0 f8 36 5f e3 45 2d 5b 21 ad c7</td>
</tr>
<tr>
<td>9c 9a 18 f4 b6 21 87 e1 08 f7 6b 71 c6 fb a5 1b 52 ae 9b 91</td>
</tr>
<tr>
<td>d mod (q^-1)</td>
</tr>
<tr>
<td>5a 83 7f bb 1a bd dd c2 06 c8 54 1c b3 72 ab 2f 55 4f 75 c9 80 2c</td>
</tr>
<tr>
<td>73 ef b7 72 b6 a7 60 79 14 e0 9e 65 51 3e c4 21 e6 f2 40 bc 94 9b</td>
</tr>
<tr>
<td>03 e4 24 35 40 6f 3d 5e 72 d1 73 30 39 17 55 de 5d 88 b6 c9</td>
</tr>
<tr>
<td>q^-1 mod p</td>
</tr>
<tr>
<td>bc 91 2a 93 6a 8d 24 3c d5 7d 12 3b a3 71 c7 3a f0 64 72 50 7e 18</td>
</tr>
<tr>
<td>71 e1 b4 3b 1e fc 38 ca e6 8c 16 51 97 d6 3f 04 ee 23 8b 45 05 6c</td>
</tr>
<tr>
<td>98 36 18 27 29 1b 4d 73 7e e8 b0 1a c7 fb 5c ea 78 7d 6e 97</td>
</tr>
</tbody>
</table>

Table D.6

Table D.7 provides the global constant (lc128) used for receivers R1 and R2. Note that the same global constant is used in T1, R1 and R2.

<table>
<thead>
<tr>
<th>Value in Hex for R1</th>
<th>Value in Hex for R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Constanc</td>
<td></td>
</tr>
<tr>
<td>lc128</td>
<td>93 ce 5a 56 a0 a1 f4 f7 3c</td>
</tr>
<tr>
<td>65 8a 1b d2 ae f0 f7</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Value in Hex for R1</th>
<th>Value in Hex for R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication and Key Exchange (Without stored km)</td>
<td></td>
</tr>
<tr>
<td>rtX</td>
<td></td>
</tr>
<tr>
<td>18 fa e4 20 6a fb 51 49</td>
<td></td>
</tr>
<tr>
<td>f9 f1 30 a8 2d 5b e5 c3</td>
<td></td>
</tr>
<tr>
<td>REPEATER</td>
<td>0x01 (True)</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Receiver ID</td>
<td>74 5b b8 bd 04</td>
</tr>
<tr>
<td>AKE_Transmitter_Info.VERSION</td>
<td>03</td>
</tr>
<tr>
<td>AKE_Receiver_Info.VERSION</td>
<td>03</td>
</tr>
<tr>
<td>AKE_Transmitter_Info.TRANSMITTER_CAPABILITY_MASK</td>
<td>00 03</td>
</tr>
<tr>
<td>AKE_Receiver_Info.RECEIVER_CAPABILITY_MASK</td>
<td>00 01</td>
</tr>
<tr>
<td>Certificate verification</td>
<td>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</td>
</tr>
</tbody>
</table>
\begin{table}
\centering
\begin{tabular}{|c|c|}
\hline
$kpub_{ca}$ & (Extracted from certificate cert_{ca}) \\
\hline
n: & \hfill \\
af b5 c5 c6 7b c5 & e4 68 63 8a da 97 2d \\
3a 34 90 a9 54 c0 & de 9a 8d 1c b1 65 4b \\
8f b7 eb a1 54 d2 & 85 8d e5 46 d6 db 95 \\
4f 22 de 83 f5 03 & a5 f6 66 74 ea 81 0b \\
a6 c6 68 46 9b c0 & 9a 58 58 66 26 86 a6 \\
b8 c8 6c db 26 f9 & b4 56 2b 29 43 e5 bb \\
3c 49 2f 02 e1 71 & 81 74 86 a7 b7 16 2f \\
df 4e f3 0e c8 bf & 07 ec d1 b5 f9 ae 4f \\
22 9d 04 cf bf a9 & 98 89 a9 91 7d 58 5b \\
0d ff 68 ab 05 6f & 8d 20 d5 c5 08 40 3b \\
1f 12 8a 68 62 eb & 86 af f4 d6 b9 20 95 \\
fe c9 ea 9f a7 fb & e8 90 3b 8f 9f 36 5b \\
8c ba b1 bd 65 ac & 46 b6 d4 1e f5 05 88 \\
35 9c a0 33 b1 dd & 80 14 e7 2c 77 5d 6e \\
a6 05 36 af 00 a2 & 54 e9 65 81 5a 68 92 \\
\hline
\end{tabular}
\end{table}

Page 83 of 112
<table>
<thead>
<tr>
<th>$k_m$</th>
<th>Seed:</th>
<th>$E_{\text{pub}}(km)$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>68 bc c5 1b a9 db</td>
<td>ca 9f 83 95 70 d0</td>
<td>a8 55 c2 c4 c6 be</td>
</tr>
<tr>
<td>1b d0 fa f1 5e 9a</td>
<td>d0 f9 cf e4 eb 54</td>
<td>ef cd cb 9f e3 9f</td>
</tr>
<tr>
<td>d8 a5 af b9</td>
<td>7e 09 fa 3b</td>
<td></td>
</tr>
</tbody>
</table>

| $\text{lhash}$: | 
|-----------------|-----------------|
| e3 b0 c4 42 98 fc | e3 b0 c4 42 98 fc |
| 1c 14 9a fb f4 c8 | 1c 14 9a fb f4 c8 |
| 99 6f b9 24 27 ae | 99 6f b9 24 27 ae |
| 41 e4 64 9b 93 4c | 41 e4 64 9b 93 4c |
| a4 95 99 1b 78 52 | a4 95 99 1b 78 52 |
| b8 55            | b8 55            |

<p>| $E_{\text{pub}}(km)$: |
|-----------------|-----------------|
| 9b 9f 80 19 ad 0e | 2a b7 29 76 fe d8 |
| a2 f0 dd a0 29 33 | da c9 38 fa 39 f0 |
| d9 6d 1c 77 31 37 | ab ca 8a ed 95 7b |
| 57 e0 e5 b2 bd dd | 93 b2 df d0 7d 09 |
| 36 3e 38 4e 7d 40 | 9d 05 96 66 03 6e |
| 78 66 97 7a 4c ce | ba e0 63 0f 30 77 |
| c5 c7 5d 01 57 26 | c2 bb e2 11 39 e5 |
| cc a2 f6 de 34 dd | 27 78 ee 64 f2 85 |
| 29 be 5e 31 e8 f | 36 57 c3 39 d2 7b |
| 1 34 e8 1a 63 a3 | 79 03 b7 cc 82 cb |
| 6d 46 dc 0a 06 08 | f0 62 82 43 38 09 |
| 99 9d db 3c a2 9c | 9b 71 aa 38 a6 3f |
| 04 dd 4e d9 02 7d | 48 12 6d 8c 5e 07 |
| 20 54 ec ca 86 42 | 90 76 ac 90 99 51 |
| 1b 18 da 30 9c c4 | 5b 06 a5 fa 50 e4 |
| cb ac b4 54 de 84 | f9 25 c3 07 12 37 |
| 68 71 53 6d 92 17 | 64 92 d7 db d3 34 |
| ca 08 8a 7a f9 98 | 1c e4 fa dd 09 e6 |
| 9a b6 7b 22 92 ac | 28 3d 0c ad a9 d8 |
| 7d 0d 6b d6 7f 31 | e1 b5            |
| ab f0 10 c5 2a 0f |                          |
| 6d 27 a0          |                          |</p>
<table>
<thead>
<tr>
<th></th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{xx}$</td>
<td>3b a0 be de 0c 46 a9 91</td>
<td>e1 7a b0 fd 0f 54 40 52</td>
</tr>
<tr>
<td>dkey0</td>
<td>4f 14 8d 11 dd 49 18 10 6f ab 16 6f f6 fd a6 ed</td>
<td>2a 04 d7 eb 0a 0b 4e 20 26 45 84 01 1e ab 0a 4a</td>
</tr>
<tr>
<td>dkey1</td>
<td>b5 02 0c 0d f2 81 ba df e4 19 77 fa d0 ac 61 17</td>
<td>f9 dc 18 97 e8 ee d8 f9 ec 6a 5d 34 a9 62 02 c9</td>
</tr>
<tr>
<td>k_d</td>
<td>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</td>
<td>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</td>
</tr>
<tr>
<td>H</td>
<td>23 0d 0a e9 de e0 18 6a 68 f2 60 80 08 c6 85 57 9a 1b f7 44 9a cf 2b eb 5a 1e 48 01 06 a6 c9 ad</td>
<td>4e ba d9 a2 43 c3 78 4e 44 46 6c 89 dd 2c 4f e3 7c 14 9e 44 76 80 11 e0 80 93 b8 2f 11 e1 52 75</td>
</tr>
<tr>
<td>$H'$</td>
<td>23 0d 0a e9 de e0 18 6a 68 f2 60 80 08 c6 85 57 9a 1b f7 44 9a cf 2b eb 5a 1e 48 01 06 a6 c9 ad</td>
<td>4e ba d9 a2 43 c3 78 4e 44 46 6c 89 dd 2c 4f e3 7c 14 9e 44 76 80 11 e0 80 93 b8 2f 11 e1 52 75</td>
</tr>
</tbody>
</table>

### Pairing

$E_{kh}(k_{a})$:

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

$k_{b}$:

85 86 c7 b6 6d 9f b3 66 a0 76 0c fb c2 ab 4d 34

$E_{kh}(k_{a})$:

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

$k_{b}$:

c6 aa 14 eb b0 35 4d 50 83 f5 de 74 2a 8c 1b a2
<table>
<thead>
<tr>
<th></th>
<th>$E_{kh}(k_a)$:</th>
<th>$e_6$</th>
<th>57</th>
<th>8e</th>
<th>bc</th>
<th>c7</th>
<th>68</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b8 9f f9 72 6a 6f</td>
<td>44</td>
<td>87</td>
<td>88</td>
<td>8a</td>
<td>9b</td>
<td>d7</td>
</tr>
<tr>
<td></td>
<td>2c 1e 29 b6 44 8d</td>
<td>d6</td>
<td>ae</td>
<td>38</td>
<td>be</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dc a3 10 bd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$m$</td>
<td>18 fa e4 20 6a fb</td>
<td>f9</td>
<td>f1</td>
<td>30</td>
<td>a8</td>
<td>2d</td>
<td>5b</td>
</tr>
<tr>
<td></td>
<td>51 49 3b a0 be de</td>
<td>e5</td>
<td>c3</td>
<td>e1</td>
<td>7a</td>
<td>b0</td>
<td>fd</td>
</tr>
<tr>
<td></td>
<td>0c 46 a9 91</td>
<td>0f</td>
<td>54</td>
<td>40</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Locality Check</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_n$</td>
<td>32 75 3e a8 78 a6 38 1c</td>
<td>a0</td>
<td>fe</td>
<td>9b</td>
<td>b8</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>58 ca</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>e7 28 a0 53 a1 d4 bd cc f9 83 ea b0 4a 48 94 35</td>
<td>f2 0f</td>
<td>13</td>
<td>6e</td>
<td>85</td>
<td>53</td>
<td>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</td>
</tr>
<tr>
<td>$L'$</td>
<td>44 86 38 0e 8d 45 42 6d f2 1f f6 3a 6a e7 0a 46</td>
<td>f2 0f</td>
<td>13</td>
<td>6e</td>
<td>85</td>
<td>53</td>
<td>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</td>
</tr>
<tr>
<td><strong>Session Key Exchange</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_s$</td>
<td>f3 df 1d d9 57 96 12 3f 98 97 89 b4 21 e1 2d e1</td>
<td>f3</td>
<td>df</td>
<td>1d</td>
<td>d9</td>
<td>57</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>12 3f 98 97 89 b4 21 e1 2d e1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{iv}$</td>
<td>40 2b 6b 43 c5 e8 86 d8</td>
<td>9a</td>
<td>6d</td>
<td>11</td>
<td>00</td>
<td>a9</td>
<td>b7</td>
</tr>
<tr>
<td></td>
<td>6f 64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$dkey_2$</td>
<td>bf ed 5a cb 93 28 d4 56 a9 f5 2e 0e f3 36 75 f3</td>
<td>45</td>
<td>54</td>
<td>97</td>
<td>7d</td>
<td>85</td>
<td>5d</td>
</tr>
<tr>
<td></td>
<td>a8 c0 2a de f8 90 95 02 7d 1a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{dkey}(k_s)$</td>
<td>4c 32 47 12 c4 be c6 69 0a c2 19 d4 de 91 f1 83</td>
<td>b6</td>
<td>8b</td>
<td>a4</td>
<td>d2</td>
<td>cb</td>
<td>ba ff 53 33 c1 d9 bb b7 10 a9</td>
</tr>
<tr>
<td></td>
<td>82 dc a9 50 fc c5 b0 8c 86 65 57 c5 fe 2c 7e d6 e4 ca 96 22 51 d5 bc f7 c8 1e 6d 0e ce 4b 64 ee</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Authentication with Repeaters</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Upstream Propagation of Topology Information</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver ID$_0$</td>
<td>47 8e 71 e2 0f</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver ID$_1$</td>
<td>35 79 6a 17 0e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver ID$_2$</td>
<td>74 e8 53 97 a2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receiver ID list</td>
<td>47 8e 71 e2 0f 35 79 6a 17 0e 74 e8 53 97 a2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEPTH</td>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEVICE_COUNT</td>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Receiver IDs revoked</th>
<th>SRM Version</th>
<th>SRM Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>74 5b b8 bd 04, 8b a4 47 42 fb</td>
<td>00 01</td>
<td>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</td>
</tr>
<tr>
<td>SRM Signature Verification</td>
<td>Hash:</td>
<td></td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3b 11 c9 ee f0 b6 ec 5b 68 34 b2 67 95 7c 2d 03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1d 83 0a d7 38 78 07 24 c9 14 c6 74 4e f6 70 b0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encoded Message:</td>
<td></td>
</tr>
<tr>
<td>Table D.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
D.3 Encryption

Provided below is the input PES stream to be encrypted at T1.

First packet

PES Header

```
00 00 01 e0 00 00 8b 80 05 2d 8c d3 96 97
```

PES Payload (Plaintext to be encrypted)

```
00 00 01 00 01 1b 3c 5b b8 00 00 00 01 b5 85 44 3b 98 00 00 00 00
cf be ab f3 f1 73 0a 3f 9a 6e 62 b0 f8 d5 55 c2 8f e5 c9 ae 7d 0d
dc 40 bf 40 bf a4 21 2e 9b c8 ba 1b 91 94 c5 38 9d 1b 42 51 85 30
a4 14 c2 b9 16 35 20 4d 21 70 9a 93 5b 4d 84 a5 65 a6 7b 54 f2 1b
4f 36 1b 95 09 4e c3 6f a1 28 d3 90 ed 42 52 d9 b6 aa 12 85 16
d1 6b ba aa 9e 9a 8a b8 45 f5 d9 34 6f 55 34 5b 4b 5b 90 d3 e0 a0 82
c8 18 db 17 aa b5 38 6b 9b 0a e9 af fc 1e 9f 07 80 1a d7 d0
```

```
30 83 00 31 03 4b a8 a4 30 32 81 de 01 a4 03 0d 00 69 00 c3 50 d4
40 65 cc 00 cc 18 28 ab a0 9f f3 d8 d5 37 72 56 c8 ad f8 fb 7d 33
41 3b 98 c1 ba 2a ca ad b3 26 8f 95 83 3b 06 d1 86 8e 18 b0 66 64
c3 46 da db cb 0b 5c c9 b9 9a 09 fd 47 02 80 17 96 db 5b 88 da da
9a 1a 3f 6d b4 b6 66 d8 d8 37 2b 42 90 cc 01 34 02 68 06 0f 00 5d
d3 72 68 4a 58 47 83 13 02 83 60 cc 95 c0 00 00 01 13 72 57 e7 98
02 18 04 30 08 60 10 c5 9d 5b 2b 67 38 d5 4e fe aa aa e6 80 c8 32 0c
8a ab df e8 a6 91 22 9c dc a9 32 68 a3 ff f7 26 4d 3b d9 f4 ef fb
bb 98 75 d8 8f 26 ae 8a 3f e7 dd 73 e8 90 d5 4f 5d d9 2a a9 55 30
52 2a 0a 62 23 f9 74 68 8c a8 cc 8d f9 4b 7d 63 5b 02 c8 20 82 cc
20 18 aa a8 26 fe 05 56 0b 5b 0a 37 41 bf 30 6f da 63 49 84 88 fc
24 87 a9 83 60 a0 0b 00 aa 0d 9f 86 2b 6c 6c 55 b7 b7 e3 8d fb 93
d7 4a 47 02 80 18 da ds b1 66 94 68 61 aa e9 02 cc 2a 82 d0 2a 82
6f ce ab 75 5b 9d 5b 6b 7e ed b6 34 13 b1 46 c9 5b 5b 56 d4 c3
53 04 80 51 04 80 51 4a dd bd 04 94 44 c9 13 2e 35 50 d3 c0 8f 94 82
3e 54 53 fa 9c 79 32 65 68 d8 27 a0 9a 3f dd 51 bf cf 5c c7 1b 92 8d
99 bf 13 6f d0 a3 53 03 09 d0 30 d1 34 03 64 d8 0e ce 39 d1 b5 9a
d4 1a de ad c3 7e 56 de bb 42 4e 81 88 b0 2c 09 72 e4 48 a1 27 c7
db 0a ce 9d 55 c0 3f 1c c4 bd d5 0e 3b 30 03 08 02 e8 01 f4 01 75
c9 a1 d8 6a 00 bb c0 0a 0b bc 00 ea 4d 1b eb b6 7b 2a ad e2 91
af 46 fc ed 93 4e 6a 00 5d a0 07 81 84 9d 47 02 80 19 00 38 d0 06
45 1f 26 00 00 00 00 14 72 57 1b f9 da da 5b 23 68 6f 89 5f 79 a5
6d cc c4 c8 48 91 22 ae f7 fd 55 54 2a cc c0 c8 32 26 41 95 74 7f ce
0c 5a 31 6f cd db e9 9a 09 d1 aa 6b da 6a e6 6d 1a 09 bc d6 56 d8
c8 d8 37 63 46 ce cc da 19 9a 1b 56 8f 43 3b 06 76 6d 4d 06 cb 9a
d6 a9 9d a9 h2 68 26 dc d7 b6 56 d2 da 36 4d 0e 8e 85 6d ed 82 b6
d6 d1 b9 9a 36 fb b0 36 a1 19 34 67 6a da c6 46 96 b6 bb a8 32
21 41 91 0a e8 fe 39 9d 95 99 bf 24 6f af 68 4d b0 a4 60 29 3b a2
5f 9f 91 3f 51 21 0b 86 2a 15 d2 44 ab 4b b7 64 ca 1a a8 37 7e 16 6d
8d 89 b5 b7 47 02 80 1a e3 8d fb 83 41 37 63 54 d8 d1 b1 cc 61 b1
6a 60 b6 0a 20 b4 0a a9 a2 9a 69 8f f3 ba db 1b 0d 6d cd f8 23 7e
d6 d0 4e e5 b3 36 78 d4 b7 43 43 49 02 bc 7a 0a f1 e9 26 af 9a bf
09 bc 0d 0e 41 01 40 19 4f 43 50 9b b7 37 64 04 d6 de 47 ba d8 5c 40
a9 00 a4 ea f9 e9 d1 a7 02 2c 06 a0 0c 05 74 10 ba e0 13 34 02
61 b0 0c 24 d4 c0 e2 0d 00 e2 0a 27 35 00 69 00 64 d0 06 50 06 92
69 a0 19 00 5d 40 ca 02 ea aa 13 7d 30 f8 3f 0e e7 52 0f 6b ae 02
30 97 ed 70 6d 73 7a 62 e2 e8 52 1f 78 32 2f c8 fa f0 0c 00 1a ba
80 61 00 83 c0 51 a0 18 34 9b df 50 06 4b 80 40 47 02 80 1b 53 be
80 34 f0 08 0a 3f 49 80 00 00 01 15 72 56 83 af fd 55 54 51 df d1
22 a9 09 9a 83 20 c8 32 0c ba 37 e6 e5 6d 0a 5a 6f fc f9 bd c6
9c ff 71 72 69 df a6 4c 98 bb cf fd c9 8b a1 46 e6 17 17 17 17 75
46 fc e6 ca db d8 ab 7e 4e df 66 8d 73 51 36 26 c1 b0 6d 50 ab be
1b 17 06 c5 d0 4f f8 3a 6e 63 5c df 9a 37 d3 34 1b ba 2d b5 b1
b6 f6 d1 a1 ae e6 15 06 45 41 92 24 d0 ab f3 08 50 14 88 50 14 79
12 6b 50 4a 06 40 96 0d 49 0d 55 23 ca 0c a6 92 52 69 82 67 e3 86
2d 42 30 46 fc 70 6a 40 c0 2a 03 00 aa 98 a9 92 6d 4a 8a 2e
aa 4c 17 61 c8 2e 47 02 80 1c c3 94 df c8 99 49 14 91 4c 13 be 75
1b 94 dd 68 dc 96 fc 29 bf 66 69 20 57 8e 01 30 35 48 12 41 40 12
61 54 93 97 cd cc c3 4f 31 45 46 aa 2b 00 4a 82 a8 26 81 54 76 80
61 10 eb 70 51 86 cd cc c3 e4 31 45 46 aa 2b 00 4a 82 a8 26 81 54 76 80
Page 94 of 112
f4 00 00 00 01 19 82 73 06 0c 36 21 24 93 23 55 0e 3b da 92 5e 68
99 c9 a3 94 0d 1a b4 bd 0d 40 71 19 d9 bd bd 50 d3 86 88 88 0c 65
49 08 47 02 80 13 0c 1d a3 15 7a 60 a3 d7 43 8e f9 70 6c 98 ba 73
e1 b0 6c 5c 5b d5 1b 18 d4 37 55 59 15 bf 38 6f a0 68 dc 76 d2 db
55 ad 56 6c da 48 10 d3 c0 46 00 be 85 1f ed 5c dd 48 8a ae 6a 44
88 32 2b 55 09 40 09 61 50 4a 8a ca 92 0d a7 61 94 21 c9 16 a3 5f
83 77 e0 55 6a 4f 95 d4 4d 6d 05 0c 6c 8d d8 6c 6c 33 6a 40
91 14 04 90 6a 48 15 c3 d0 56 8e 42 4e b8 ab cd 9d 54 92 13 26 66
64 90 b2 4a dd a2 2d 28 d8 a2 a9 26 64 6c c8 cd 52 31 7a 84 7b b2
78 8d 2e db 67 51 09 bc 36 99 b8 89 d1 d1 31 bc 1b 44 a0 23 64 0e
69 c1 66 f9 f8 1a a7 a1 b2 01 8b ff e4 83 47 02 80 14 10 00 d6 c7
54 06 6c 1c 83 ea e8 69 d3 56 41 12 f4 e6 d2 b0 0f e8 6e 13 34 26
31 9a 96 2a e8 6e 35 6c 08 50 f5 53 d4 87 d1 3e 8a 6f cd a6 d0
5b 15 6b ca 6b ba 12 84 3c c6 a6 06 12 a2 63 26 2e 4d bd 35 09 be
4f 06 cd ea e3 e5 48 c8 da 56 ad 07 13 0b 29 7a bc 99 34 26 d2 58
5e 4a b1 92 5a da 4b da 4a 87 53 f6 be 50 b8 b8 ba 12 78 75 30
e7 fd ac 95 b4 99 30 00 00 00 01 1a 93 cf e8 e7 c9 44 57 86 50 eb
b0 7e ee a9 33 27 26 6e 4e 91 11 b9 a8 df 95 91 b7 58 dc 06 d8 df
0c de d3 79 f2 aa f3 db 55 5e 7b aa ab cf de 85 1f df 42 8d d5 b5
db c9 7d 4f 47 02 80 15 45 f5 fd 32 64 dd 73 e9 92 d1 6f 16 f7 9e
a2 6c 5b 44 dd d7 d1 35 f5 d0 a4 3e a9 56 f6 f5 cf 84 4d ad e4 d4 c1
2a 44 12 c4 41 bb f0 ab 54 d7 35 4c 9b 46 82 86 26 c6 db 5b 0b 0c
34 75 60 d1 83 26 96 9b 95 02 e2 52 0b 81 15 b1 92 f9 9a 21 40 56
89 ca 09 f0 40 3a 17 81 8a 10 80 c5 12 66 b9 7e 09 48 ed 43 7c 89
34 35 1b 4c 17 d1 07 57 de 97 5b 3c 3c c7 15 4d 94 e6 3b 96 2a 86
9e 7f 21 f4 39 b8 79 7d 2a 3a 84 dc 66 09 29 a5 f1 f5 b0 2f 53 3f
ba b4 2b b3 da b5 3b 82 f5 30 aa 86 9a 07 54 60 a8 24 37 55 d3 46
7a 46 02 ea ab 6c 3c 32 41 89 9d 23 3b 46 fc 29 47 02 80 16 bf 4e
1e 38 23 52 d6 35 4d 6c 3b 55 aa 6b 5a 96 b6 1a 09 d5 31 6b 1b
2b 53 53 41 36 36 56 b5 91 93 43 46 ca d9 9b 53 65 66 d4 f1 b3 b6
56 d6 d8 99 b0 d1 f3 36 46 d6 d9 1a d9 b4 7c 6d 43 5e d4 b6 8c 34
6c ad 43 3b 56 d4 d4 d1 aa 6a d9 9a 86 a1 3c 65 66 60 d6 c9 a6
b4 95 0c 4c 9b 79 72 e2 a0 00 00 00 01 1b 8a e7 12 d1 7a 82 77 e0
1c ab 1b 9e de 39 be 76 37 a3 5b cf 30 03 58 03 38 03 58
03 59 35 4e 6c aa a8 df 91 9b 63 61 6d 6d f1 0d ed b4 7d 23 73 52
d7 34 35 36 bd 8d f0 b6 76 46 d0 df 9b 37 d4 34 28 ae a6 a5 64 14
6f 1f 93 a8 49 f0 47 02 80 17 49 09 02 50 34 6d 74 14 3f 1d da dc
e9 5b 42 34 46 8d bc f3 00 2f 80 2f 80 33 80 33 80 a4 4d a9 82
50 34 04 90 51 34 2e 05 fc a0 1d 0f 22 02 90 15 ad 41 80 6c 0c 83
26 06 41 90 64 19 22 85 1a 06 68 19 1c 00 4d e0 04 10 02 4f 00
20 80 43 01 4b c3 e0 33 f3 00 1e 40 0f 20 07 90 03 c8 87 15 47 ee
b4 65 68 df 87 37 ea 4d 1a 31 60 c5 93 4b 46 ad a3 06 8c da da 39
64 6b d1 b9 99 38 7b cc cc da b6 4d 18 b5 6d 73 54 d4 c3 47 46
0d 19 98 6a 68 37 67 68 c5 83 0d 6d 04 d0 c1 a3 56 d0 d2 d7 39 f7
47 a1 91 99 83 7e 18 df a8 b4 28 ef f7 26 4d 0a 37 e9 47 02 80 18
8b 8b 8b a0 9d db 3b 5e db d9 df f1 46 fd 3d a0 9b 9d b0 b6 a6 ca
c9 86 8d 99 b3 b5 ed 85 b5 66 d7 75 10 e0 2b 10 e0 2b 54 6f 84 35
2d 6b 52 df 8e b7 ea 4d 1e 86 56 76 56 6c da 6c da 08 58 1c 01 71 07
61 e8 7b 91 5c d0 54 19 13 20 cc 7c 19 00 00 00 01 1c 72 e7 e9 aa
b5 6a 06 90 58 03 40 2e 32 3b c5 c4 b9 a2 a4 d2 b6 73 2b ca ad 41
f8 6e 0f c2 b5 f6 b0 5d 26 37 e7 16 d4 d5 33 b7 e7 cd ee b4 7d 0d
73 50 d7 32 6d 5a 14 57 0f 6b 18 aa 57 57 af 95 b0 a3 b8 97 a0 dd
80 15 0d 89 b1 71 51 74 1b f8 55 bb d5 b8 55 bb f5 bf 1f 8d fa 35
68 55 b9 80 0c a0 06 10 03 28 01 84 01 ac 03 0a d4 0c 52 20 62 90
2e 19 23 4b a9 30 9b 18 27 6e 68 7b 74 65 6e 5b 7e 12 df ac 34 74
68 c5 18 b4 34 1b 43 16 8c 5a 98 68 37 50 d1 a3 33 53 0d 1a 96
46 2d 1a 99 34 28 de 5d d1 f0 32 35 8d 4b 7e 18 df a9 b4 13 83 06
2d 5b 68 cd a0 9a ad 4b 5e d4 35 32 68 ca d9 da db b5 1b 43 9b
36 86 e2 03 38 08 19 c0 7e 7e b2 ee a9 de 02 02 23 03 38 2d 81 9c
16 98 52 a2 9b 44 c6 f1 32 0c 93 41 3f f4 39 91 ac b9 1b d7 96 d4
36 cd f6 68 1a d0 d4 07 41 80 47 02 80 16 e8 30 96 32 08 a1 46 e6
00 34 19 00 3a 80 1a 40 0e a0 13 4f 41 1b e9 ad bd 1b 03 6e 69 6f
bd 8d 1e d6 b9 a9 6b 9b 76 f1 23 46 11 b3 35 4d 53 5b 73 b5 cd 48
ad 20 65 0c f0 e8 00 00 01 23 73 cf 0b da e2 ca 9d d4 53 54 55 55
3d e7 a0 0a c4 38 0a c4 39 37 9e 82 1c 43 88 71 0e 4d 1b f0 73 16
cc ce df 1c de cb 6a 1b 3b 64 d4 d7 35 06 c5 f8 36 2f c9 a6 81 08 d0 21 1a 5d 41 43 f3 4a b6 85 6d ad 6d 7f 3e 6f 41 a9 01
81 81 8b b6 0e 7a 6d 0b 2c 5c d4 24 f0 5a 68 51 ca ee a6 95 41 3b f2
d3 16 df 2b 71 a3 7e da 52 ad f6 68 d0 d3 c0 ba 0c 00 ed 16 37 89 94
47 02 80 17 38 e0 59 02 89 80 0f 60 07 70 03 d8 01 dd 43 3b db 9c
c0 00 77 00 39 80 1d c0 0e e4 40 3b a1 26 c1 20 16 41 24 17 2d e8
4a 03 20 dc c8 56 6d 74 1b 6f 30 b6 96 ce da 19 36 ad 04 f0 ca ca
c1 9b 26 83 75 6d 53 58 d5 b0 d4 d0 dd 00 67 c9 80 cf 93 30 9b 06
c4 d8 36 0c 8b a1 c6 81 9f 1f 10 e2 fc 8d 26 8d b5 76 56 e6 36 a6
cfc 41 bf 5f ad 14 77 98 03 a8 06 11 0a 01 d4 05 62 15 04 df 68 c1
b1 32 a3 7e 42 48 6f db 5a 09 d4 46 e5 37 49 b9 cd e6 b7 13 46 56
56 2d 6b 64 dc 9a 34 62 d4 c9 a0 dd 53 33 2b 16 4c 9a 35
2d 99 b5 b2 30 d4 d0 6f a9 ab 67 45 47 02 80 18 66 db 4b 1b 6e 60
c6 30 6b 64 d1 eb 6a 65 f6 a1 9b 26 8d 4b 06 2d 5b 53 5b 43 8f cc
00 7f a0 0c d3 00 5f 00 6a 05 87 c0 57 dd 3b 9b 03 b5 c0 55 70 0e 02 2b
17 3d 1b ee a6 d6 eb 9b 5c cf 23 76 51 b1 6f d1 1a 2b 0f 40 14 5c
05 20 29 01 48 32 85 1b 98 00 e3 40 0e a0 07 50 03 c6 80 61 55 cd
00 0d a0 07 10 03 68 01 c4 03 0a 28 ef 30 a0 40 30 80 a4 05 22 aa
8d f5 66 b9 aa 8d 73 68 df 49 1a 36 48 d4 b6 a8 d9 1a 9b 9d bc f3
03 1e 0c c6 41 59 a4 c0 00 00 00 01 24 72 77 09 a2 a2 a4 7a 37
e1 36 2a d7 35 ed f2 8d f4 32 b4 7c a8 c5 5b c8 5b 6c de 34 68
d4 ab 47 02 80 19 66 56 b9 1a 84 6c d9 34 6d 2a da d5 85 8d d4 6c
5b a9 ae 7c 9b ee 0a df 75 33 07 8d 19 0f 0f cc 15 bb 6d 5a b6 e6
f8 26 dd a0 9c 5b aa df 59 0a 9c 8a 34 6a d8 35 6d 8b 68 d1 af 6d
4d 89 ba a3 53 6c ad 1a 32 ab 77 eb 73 a5 95 b2 6d 16 34 14
3c ec ad e4 25 6f 7a 8d c8 68 6c da e6 a0 2a 0f 0c 5e 26 e8 0a 0d
04 ef c9 6a c5 5b b0 d9 db f3 b6 eb 56 83 86 c6 ce da 5b 2c 69 6e
cb 5a 98 24 81 e8 24 81 cb 48 9b 60 c6 d4 24 f8 9a 69 e7 55 4c 2c
66 f0 a4 01 7a 0b 37 de 78 20 6f 9e 2f df a1 d6 e6 17 26 2e 4d 69
34 90 1f 60 5e 46 18 4d d0 30 81 72 06 24 47 02 80 3a 35 00 ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
ff ff ff ff ff ff ff ff 14 17 55 5b 9f 26 dc 6e 58 ba aa 9c 65 4c 98
77 52 08 77 0a 34 0e a0 33 8e fa ee 85 1f 98 99 32 6e ee a4 09 00
5e 83 2c 28 d0 4c 01 86 1d d7 3b eb 9f ee ee bb ab bb ba a2 8f 03
1c 07 3f c8 71 70 6c 5d e7 8a 26 4d 5e 78 15 11 32 a4 48 a7 33 03
64 c4 3e d0 6a a4 c5 50 e3 73 13 17 06 a0 0d 8b 91 4e e6 72 e6 e4
c8 a7 73 0b b8 bb b9 14 38 ee 35 5e 7a 83 20 c8 30 2e 0c 93 5d d4
54 89 15 26 47 42 80 1b
Second packet
PES Header
00 00 01 e0 00 00 8b c0 0a 3d 8c d5 07 1d 8c d3 b2 b7

PES Payload (Plaintext to be encrypted)
00 00 00 01 00 02 13 51 4b 80 00 00 00 01 b5 86 5f fb 98 00 00 00 00 01 b2 44 54 47 31 41 fe 00 00 01 01 5a 37 f1 26 b6 a6 cb 5b 6b 66 cd a3 5e b5 b4 ad 6a db 53

The encrypted PES stream generated at transmitter T1 for receiver R2 is provided below.

First Packet
PES Header and PES private field
Encrypted Payload

inputCtr = 0, streamCtr = 0 (sent in PES Header)
d2 11 bd d7 6b 8e 75 bf 3a bd de b2 c1 d2 0a 09 2f 33 4f 48 4b 38
75 ef ad b8 f2 7b fa 19 d4 6d a5 0f f8 91 5c 55 68 02 3c c4 76 7d
35 74 e8 35 7b 88 0b 81 45 98 0e fe d6 c7 85 14 25 18 a2 9e 22 f5
02 24 ad fd 1e 96 5c 81 14 le c1 23 0d 44 4b f7 8f c5 9d d3 d1 1c
13 12 00 40 70 14 Od dc 34 ba d4 0e 99 5a 10 61 d4 98 fc fc 9c d7
b9 bb b0 aa 2f bc 7d 19 c6 bd 4e 37 65 b9 b2 9e 6d 47 2c 93 bc a0
7d d4 78 d3 03 35 0d 9a 09 2f 33 4f 48 4b 38
84 49 39 20 1f b6 4a 66 b6 82 aa 2e 2e 8b af 4d 61 00 40 18 35 5f
e9 62 d6 32 90 94 c9 95 a6 0b 89 e3 c0 4f 6d 2b 18 f6 58 7e 67 c5 7e eb
f9 eb d0 18 35 70 03 0a d2 60 76 3b 94 10 fc 61 92 21 4b 26 54 ff
f6 b8 87 2e 56 le c9 51 04 fe d8 ce 7a d8 03 07 4f 00 8e 0b 3e 11
fc 51 00 8f ee ce 5c 7b 25 0f 60 ac 4c 45 5e e3 ed 5d 4d ce 5c 31
4c 5e 23 e0 b0 f0 05 56 fa 3c f6 30 2d f6 e4 98 9d 1b 49 84 ca 39
0e 86 0b e3 8f fc 72 e7 0e 2e b6 6b eb 6d c1 17 5f c4 b4 f9 36 c6
d3 d3 4f 7e 5f f7 ef 57 37 75 cb 4a 4c 22 5a de 67 f7 01 8e 09 e9
28 8a 44 04 56 1b 15 53 e5 46 ac 40 e8 81 92 79 52 f2 d2 79 03 c2
f4 d6 2f dc da ed 7e ee 42 d6 38 7c d8 11 8e cb e4 a9 25 0c fd d6
4f da 3c 65 5c 6c 89 b5 98 01 ed c0 6e 83 d3 3d 3f f3 78 f1 84 05
65 eb f3 dc e6 8d df db 68 a9 f0 00 87 4a 80 85 a1 f9 22 b5 fa
43 6d 68 5b 42 71 d6 f3 85 ea b9 79 78 74 cb e6 11 72 1a db 82 c6
be 70 18 be 6c 76 f2 3c 40 e6 f4 9f 6d 65 69 c8 10 73 07 45 4d f4
bc 13 8d b2 f0 5f 5b 7c 57 c5 37 63 ea 97 67 cb f5 21 f5 85 07 5c ed
2a be 8e 25 1d 11 a6 44 33 8e 22 54 07 11 f2 3f 68 54 33 82 05 0e
25 be b0 75 7a 4e 12 0d 2e ac 8c a0 6b 15 a0 a0 f6 9f fe 17 b2
95 3f 6e e8 72 29 20 26 a7 24 21 03 bd 68 58 6d 31 74 64 db ba 4c
eb db 36 9f c2 d5 92 9d 9e 2a a6 ea 4e 16 2d 81 55 2b 20 72 5a 65
8e 40 e1 02 5a 0e 62 6c le d7 e8 0e c9 69 b0 86 ea 0a 19 4b 37 35
41 fc b2 4b 5e de 4c fc f9 ad 93 01 74 8c 04 4d e2 d6 37 04 58 0e f3
17 51 94 b1 56 f5 8b f3 4d a9 95 7b ec 79 29 e6 38 12 10 19 cf 8d
98 54 0d da 69 03 2d f0 f3 31 b5 fb 0e bd a7 b2 e7 f0 bd 98 05 6b
f5 c5 8f 22 ba 16 36 47 13 78 b9 b0 e2 58 cd b6 e7 e5 48 93 27 bb
c9 49 a0 af 55 2a 62 bc 1c f9 94 71 fc gb 8b d3 bc 72 88 5b 8a 87
40 c0 70 0d c5 93 85 1f a9 46 a5 fc 03 67 6f fb 64 66 b3 bc f1 2b
86 68 5f 7c 18 d0 f6 bo 07 5e a4 36 81 44 b9 04 a1 84 73 d3 f8 3a
8f a7 b1 6a cc 0f 24 d0 4e a7 10 fa 00 63 87 72 46 ce 15 99 0c 82
5d e4 41 d7 cf 3a b4 f8 5d 7f fb 0b ec af 05 23 a3 2c d6 79 8c 0c
11 4b 99 97 7c 3c 34 7d fd 96 0a 82 63 ac 9f b5 79 7f c6 60 3d 3c
69 37 73 07 24 3a a2 1e fa 49 e2 39 ab ba a8 70 38 36 a5 8f 03 39
4a 9b 7d d9 2e c9 f1 22 3f ab e3 a7 b9 9b ca 28 5b af e3 f6 62 1e
b3 8b la 51 11 68 fc e0 6c c2 c0 98 5b 88 c5 34 5d 89 5a ea eb be
a1 37 81 0b fe 54 8c 4c 5b 2a a3 03 76 b5 09 c0 52 c4 e3 2d df 29
ac c7 44 22 7f 80 b1 f4 07 45 13 55 bd 42 85 ba 69 4a 87 95 b9 10
86 f9 92 39 d4 e1 b0 64 6e a0 80 ec 5b 08 79 c8 f6 a8 98 6a 8e 10
3f ac 9a 43 86 d9 67 d0 a6 fd 4b b6 dx 6c b5 ec b1 83 75 32 88 30 68 43
c0 a7 b6 1b 6f e1 f9 58 8a a0 d1 a4 a7 45 e4 b5 d6 18 bd c3 1a 8c
df 3e 2d 34 b3 3a 30 f7 b1 1f 9f 76 ed 6c c4 4a c5 fc 73 3a 1e 55
2a e8 42 62 9b a3 f7 0f 94 49 b4 91 ad dc 5a ee 5d ld 63 30 1c f0 95
70 90 ab 4d aa d8 a4 25 5c 44 20 d1 fd d4 dd 59 83 08 a2 ab 68 c6
c1 b3 c8 70 2d d6 59 d6 e6 de 5c 2d 80 49 6e f3 5a a5 4f a5 8f
a3 33 35 88 3c 48 77 74 f6 88 9a 21 c9 21 ed 0c 11 32 fe b7 d1 4d
4b 2f 85 50 24 bb 2c d8 a8 77 40 51 ce 85 d1 25 a8 e5 6b 62 0e 35
84 e6 ca e2 b1 69 aa 74 43 0c 19 f8 54 22 07 6e 48 5c 87 b3 9c 40
f3 fc f3 ce cd 6e d3 48 c0 c0 ed c4 e2 d7 4b 1d 88 0d 1f 7c 37 e8
32 67 29 1a e8 4f 0c 37 40 96 8b dd 0f 14 fc ce e8 b2 29 28 bc c9
54 cd 5a 81 e0 d7 19 88 d5 10 69 1a 27 e0 a1 6f df 6c 75 c3 1b 65
bc 70 6d 79 64 9f 76 64 69 1a 15 b2 ec 71 c4 6d 5f 54 cc 27 fb c2
ef ab 02 08 c9 a4 6b 05 53 6a f8 53 10 d8 ab 1e 34 d8 6e ee 2b 46
c4 d6 41 3a fa 59 21 d8 56 5c b0 79 53 d0 71 6f 00 f6 c4 95 58
bb 77 6e 53 b0 30 68 08 3f b6 2f 26 83 98 fc 7c 01 99 4e 4c f6 8d
a6 41 cf 3d b3 99 7b 7a ca e3 c1 1f e7 6d 63 bc e2 58 d5 38 d3
2f 06 49 e2 51 f7 68 56 16 c4 fe ef 4a a7 d0 ec 6d 5d 86 ca af 1c
a2 72 1f 2b d3 b7 f4 c3 13 76 e7 61 fe ef 08 13 86 77 e6 f9 c1 55
da 8c 54 96 1b db 4b 0a 6b 9c 1e cc 14 99 52 85 e0 ba b4 39 20 19
c9 a0 b5 1b 11 56 be 9d f2 57 ff d9 16 9d 1a b5 54 38 83 39 2a da
e2 26 f8 b7 9d fa ee e0 c0 be 60 6f d0 0a 3b a9 f5 14 f4 3e b6 e0 a2
db 67 db 72 cf 64 50 89 c0 e1 5f 6b ee 63 ec 08 f6 e4 d0 3e 55
9d 29 5f 53 74 6a 3b 5e 44 88 d0 be 5e 89 1c c0 81 3c f0 b5
36 30 8d b8 a7 f0 91 4e 85 94 da 20 c8 0b 04 56 e4 e2 e6 2e ad f0
27 cd 19 cb fe 03 75 7d 68 2f d3 43 cf cd ed 08 4c 19 5f 6f 01 e5 11
17 82 5d f5 03 d9 2b b7 7a 81 19 70 52 b9 1a 64 40 47 f7 ec d1
a2 9f b4 eb 4e 3b 1e 8c e1 fb 62 97 4f 21 9a 8d 43 9b 4c fc bf 8f
e3 d3 89 dc 68 f8 56 e0 f0 79 39 ef 0b c1 9c 33 8a 3b 68 77 3f c8
5b 91 b0 ee b2 32 17 dd 2b 99 59 62 55 de 58 ad 69 97 1a 02 0b 80
c6 a0 d5 9a d4 42 b0 a3 0b 77 4a ea e7 bc cf bc ee dc 9e cf c2 a5
a9 96 5c 5d b3 f8 54 d0 5d f0 62 05 20 ae d3 b6 2b 2d 72 a2 f1 84
73 3e 52 14 ac 85 72 40 ed be 5b 9d 1b 9b 1b cb b2 c2 ff 0d 57 af
1c f5 54 3d 2e af 67 fc 9a f9 7b af 9e 81 6b b1 0f 02 a0 1a 62 e8
07 f9 47 65 1f c8 7b 83 16 a7 fe 56 0b e1 34 f9 b8 0c 27 55
e6 80 55 ad 56 f4 b2 d4 6c b2 c2 34 90 eb 98 5e 64 Oc 2f 2f 81 c9
b6 f5 d3 5c f7 aa 30 ca e8 e8 49 9b 90 d3 1d 58 48 d7 f6 90 ff d0
51 38 ff 24 7e 6b 29 87 ad c6 9d 6c 0a ce 67 a3 00 98 77 10 ba c4
ae 84 a6 da a2 63 eb 44 a3 0f 92 08 d7 22 d6 31 e2 dc 8a 5a 7f d7
9c ea d1 ba a0 e6 d1 e3 6b 9e a2 31 a7 aa 86 f9 3d db 5d 09 31 d0
af f0 43 7a 59 df c7 7f ef e7 62 31 ab 45 99 ec ef 3e 88 7b 5f 3f
49 9e 22 dc ec b2 69 fb d0 3b c3 74 83 72 dd 9d 86 f1 f6 1b 81
32 66 66 02 c0 e0 57 d8 34 30 57 68 7d 4c 71 fc 1c 5a c6 83 63 91
f5 ea 28 84 ab d9 45 5f e4 0f b6 0d b1 d6 14 0c df 08 98 4a 84 1d
82 dc 61 6f 62 3a 7e 6b 47 71 76 b7 0e a0 10 d2 e6 24 00 68 21
11 1f 4d 46 4d df d2 b4 4c e4 83 4e b3 82 36 32 79 43 ca 3f 50 7a
6e 17 46 34 38 3a ec 8f 15 27 31 bf 2f 01 a0 d5 b6 04 11 b2 ca 73
22 c2 fe 5d ce 3a 7b 0c 64 5b 3e 65 45 3a 30 5f ea 32 b1 9e 49 6e
f9 a6 eb 32 65 86 2a 1a 96 a8 e4 ad 00 18 b1 ab 63 c4 d5 f8 2c 73
33 3a 80 0f c9 f2 1d 3f 50 c0 40 5d a5 ac 91 89 7e fd 5a a9 b2 b2
33 7b 7c 4d 25 90 95 de d2 6a d6 56 ad 5f 61 1e 58 57 c0 34 0a
67 c3 8e af 00 c0 1b b6 42 5e 28 3d 97 a9 03 d3 fe 92 a1 b0 e0 cd
70 ac f9 f9 9b 7b 49 8a 0d 4b f8 25 50 f8 8c b6 01 53 a7 f2 38 9b
6b 8d b2 14 a0 94 ab a6 41 f1 f6 ea 98 bd 9b 43 bd a8 4d 21 b8 39
16 02 12 1c c8 50 1e 1d e1 29 97 2d b3 80 ae 9f 42 61 fe 62 9b fd
ad fc ef 6f 78 82 8c 2a 0d 65 70 44 7d 17 98 cf ce 01 bb 0d e6 d1
41 91 60 98 6f c3 19 cd 78 3f 84 ff 81 a1 6c 49 6d 6b 1f 8a 6b 5e
f7 48 04 25 c9 5c 7f 79 e2 56 8b 4a 88 be 53 fa 77 21 0e 34 1e 71
2b d9 6c 6c 9f 8f 5a 89 41 21 40 e4 f7 f0 73 1d 29 b1 d9 15 b8 bd
4d 80 b9 61 34 db bf 8d 78 13 ae c8 00 17 0f e1 49 c7 88 b2 ce 7b
b1 5a c9 15 81 51 66 17 54 13 ec b6 a1 6a 92 cd 2e 8e d3 96 7c 6c
11 2a 42 1d ab 36 67 88 3a e2 2c ee d0 4d 26 11 49 71 fb 8b f3 61
42 70 8c 2c 6a a4 ae c0 05 22 f9 94 e6 da 4f 98 db bb 25 56 92 b1
36 61 ac 8f 08 34 40 f7 49 85 c1 f4 b0 62 9f 5c 9f 9c 18 9d ba fc
3f b5 16 37 b6 51 47 c6 d3 1f d8 cb 4e 53 d8 3d 2f c8 9f 4b 53 7c
a9 4d 2b b0 0d dd ad 5d 5c 72 8e bb 5e b0 d8 22 1f e3 af 22 9a
cd 30 8e 6a 34 18 8f 49 85 c1 f4 b0 62 9f 5c 9f 9c 18 9d ba fc
3e 4c d5 0e 5f 08 ee 26 55 63 d9 cf 02 d0 80 39 74 f9 24 f9 f3 3d
0e 13 2f 6e 53 aa a9 69 47 ea ce 11 74 e1 63 b3 5c 2d 3d fb d6 a8
e3 d5 ff 62 34 03 e0 1f c4 97 1a af a7 8f e8 43 61 f6 8a 20 64 c7
dc f8 76 66 0b 86 73 c1 f4 96 22 17 d7 83 82 b4 7d 9e f4 61 e5 8e
6f 32 17 26 21 9c 45 48 3d b1 81 15 1e ae 85 f8 39 d9 86 d1
fa 03 b5 b5 3c 1b 9b 78 44 18 73 9c 6b fa 23 0a b7 67 73 cf 9c 5e
77 50 f3 95 2f e3 19 1f ef 89 2c 44 19 cc d9 00 3c 52 72 ad 59 fe
29 ea c5 1f 70 7f 0e 1a f4 81 ac bf 48 a7 f3 5a e1 b2 a5 d8 7b 42
e8 dc 46 55 49 af c3 20 da c9 6f 01 a7 bb e9 c7 20 47 ef 88 25
13 e9 95 d9 7b f3 ea 91 d1 ca 2b 8d 18 88 8e ca 16 dc 9a d6 2b 52
59 e6 c2 c6 64 01 7b 8f 22 59 08 0a 69 17 81 75 df 76 67 5f 1c c0
f1 16 ce b8 c6 06 3b e3 28 62 f1 c1 f1 f1 7c fd 6c 3d 26 61 01 04
76 1f ef 9a bf 34 72 a9 dd 73 30 4e 52 e5 cc 00 62 7d cd 92 aa fa
d9 05 b8 06 81 85 80 ae e6 58 9b f2 db e6 21 56 b1 4d 0f c1 11 74
d1 ad 68 9a 51 48 68 3d b1 42 2a 97 9c 98 1f d3 d4 56 5f 04 bf 24 28
af 60 01 1c 49 f8 72 a2 ef 55 f4 c6 4e 3f 57 2f 7e 5b 5b 1d ca 40
18 bc 78 4e 8f 11 d7 2f f2 84 e5 55 d2 df 73 79 cc 22 dc 09 47 ab
91 1e 82 45 5c e2 b0 c9 ef 05 db 05 42 cb 85 e0 0b 3e d8 1f db 1c
83 11 66 d0 cb 05 7f 89 2e 0b b7 16 4e bf 1f 44 9e 56 1c b0 dd 31
e4 88 38 40 d4 b6 d7 0a c9 6c c6 18 42 0a 62 fe f0 28 d1 7e 29 6a
bf 57 37 5f c9 08 ac 16 3c 01 b9 93 70 d6 f0 04 f1 8d 7b 47 a6 96
42 6d 10 de da c7 f3 53 3e 85 a3 25 3c 7a 98 6d 03 8c 6a 27 37 33
2b 97 d7 d7 31 b0 11 f2 fe 4a 6a 24 e3 75 e5 fe 4c 85 02 5d ff 7b
85 26 e5 7d a9 b7 19 9b 09 80 43 97 4d 67 93 17 da e0 57 8d da 07
b0 c7 e3 89 e5 c9 9c 6a 75 04 01 f8 f5 26 74 80 a7 01 39 80 1f 66
92 3f 5b bd a1 e4 a7 9f 86 de 46 f5 44 4e 98 72 84 05 14 eb c7 dd
54 d5 3b de db 57 20 98 c6 9d 48 40 27 b8 c9 a0 23 2f b3 20 02 76
bf 85 44 12 9b 0a 02 b0 56 18 2e 7f 21 3d b4 a7 38 f6 69 bb 72 99
ed 74 65 42 e2 0c b8 bf 07 e4 6e 95 39 ab 15 7c cd 77 4b 6d 9f
0f 93 74 3d 02 2d 4f c2 a8 82 52 73 d1 9e f4 6c 68 01 93 a5 09 76
42 68 6f 0e 8d 83 7c 5b c7 21 1b f1 95 b2 8c ed 85 c3 72 cf 9d df
02 97 1a a3 fb eb 43 0f cf 4a d9 17 39 b6 1a 80 9d 19 23 4f cf b5
2a 70 80 8e d8 15 3a e3 42 b8 af 7e eb 89 11 da a0 40 26 6b 32 fb
2d c1 df 7d 13 c8 5d d0 13 76 0f 19 56 0f eb 63 55 fe 0d 82 f6
b8 70 8a 86 b0 7a 61 09 07 37 a3 29 7f 3c dc dc b7 97 de 74 a5 a9
45 7d b8 65 48 2e 02 0e 96 67 b3 56 ea 7c 08 55 98 f1 fe 47 ae e1
08 c6 df 3f 52 4b 73 8d 41 f6 65 ce aa 7c 7e 10 40 a3 7e 37 fa 0b
80 9b 84 35 60 1d 24 73 dd c0 b7 13 ed 86 1d 0e 72 50 3b ae 7c a7
8d 3a 85 44 b2 e4 2a f9 a3 e1 15 63 b8 34 53 fb 8f 50 48 5a 25 69
52 44 1d a4 37 a4 92 cd 25 87 f3 81 e5 82 e4 c2 fe bd 87 00 cb ab
f7 2d 7e f0 0f 74 46 76 83 1b 93 0b 44 6d b5 9c 61 5a 43 d5 5f dd
71 dc 01 d9 1d 0b 15 1f db 22 11 d4 7c fe 79 3b 3c 47 05 b5 bc ac
98 b5 b9 c6 85 46 7e ac a8 6d 35 f9 a4 b5 0a 56 7e ce 5c e1 df 3d
54 96 52 1f 83 5d 50 d0 25 2b c6 bd 55 37 3a 97 f6 39 76 ec 1d 50
da 1e 3d 1f 50 81 18 8d 7b c1 2f a9 a6 e7 9d e4 6e b3 f3 3d 45 d7
70 c6 16 1b 0c 29 bf 1e 3d b4 c9 6f 0c c1 b6 40 1a 84 1b 49 41 e6
33 6c 56 45 a3 6b d5 d8 49 ad 59 db 89 40 f4 e9 7e ac ce e2 bf 80
c9 3d 6e 7b 0b 63 14 5e 47 1e ef 03 b4 fe 79 40 36 71 3e 9a 1f 59
45 32 f3 b1 ea 3a 8c bf be e8 ad ab 26 f0 d0 c2 d2 ad cf fc 0b
98 d0 32 3e bc 6d 98 4d 39 97 ad 6f 08 0c 13 40 a9 4e fe 52 ec 28
e8 a9 e4 f9 bc 21 74 f5 b2 c0 e1 38 ab 23 a7 6d 1d 59 7d 65 ca a4
4a 17 d1 06 cf 82 06 21 e6 ce c1 66 8e 8f 7b 30 7e 05 ba
c0 43 60 aa 80 72 52 55 82 63 76 ea 62 d9 76 b7 9f 4a 47 d1 2e ee
23 fe 10 ef 62 bc ba 93 5e 58 af 8f df 3a 9a ab 8b 5a ea a2 02 91 7c
32 a3 af e1 97 ba 3a d0 0e 18 55 80 99 5d 7e 4e d2 71 87 05 ab 07
c3 47 80 a8 f7 12 74 d1 92 66 10 34 41 dd 3d 09 7f 23 8d da 68 48
c4 ee 8a 7d 09 b4 2f a1 84 f7 78 cb 4e 36 63 79 07 dc 9b 06 be a6
1b 81 b1 0a 2b 6b 42 fa d9 9b ab c0 20 b5 14 99 bd 20 fd 46 26 b4
91 25 ff 7a ac 05 18 35 b6 21 8f 39 5b c0 76 c8 d1 eb de 28 a5 64 f4 11 05 25
fa d4 c4 3d 5b a1 a6 b9 53 c3 3a b6 b1 84 32 5c bc 1b 25 58 cc 22
2a c6 dc a5 76 9c 05 64 4e 9c 66 35 c8 60 10 87 1c dc 60 8e 4f 38 a1
e6 a1 d7 08 7d 6d e7 41 63 af 6e a4 74 65 e2 aa a9 61 a6 95 63 ba
d0 d4 c8 e6 7b eb ee b6 02 eb 15 4a 71 99 88 8c 87 b2 43 fa 2a b2
63 9b 3e 2c dd 63 30 72 2e af 2c b6 2a c4 0f dd e1 aa 9e 92 f7 d4
db 82 2d 2a f3 06 91 f6 c0 76 c8 d1 eb de 28 a5 64 f4 11 05 25
1c 1d f8 0b 8a 74 fd c7 34 bf 86 6c 83 35 dc d3 d9 da 3f 08 bd e5
11 68 c6 99 0e 8c 7b 43 92 3e df 70 e4 6c e2 69 99 35 7c d8 f4 01
3a 1f 43 47 c0 c9 08 41 07 ba 0a 16 c2 69 7e 6c 3f e2 b7 9b 9f 7b
f2 bb d8 24 21 4e 6c 2d 87 da f5 4a ad a6 31 07 11 cf 56 02 8c
c0 eb ab 5c 72 6f 39 f5 08 9d ce ff f9 f7 8c 79 0f 7f 16 5d d8 f2
29 dd 08 c7 a4 2d ac a3 e3 e4 61 d9 85 e5 0f 60 5d 5d 82 7d ff 7b
f3 4c fb 46 75 bc f6 e7 89 77 f5 30 15 dc 0a 04 51 30 20 4f 3c f8
1c 9c af 6c 9b bb a6 f1 9a bb 25 ad d8 23 de e4 53 86 81 ef 65
d0 40 99 d4 5d 8f 53 5a 9a 4e 08 82 19 17 8e 17 1a 2f f0 37 b7 cb
00 94 a5 8e 4c 12 62 c4 fc 5b 6b 59 00 35 c6 15 bd ab f3 30 1c bc
e6 7e e8 66 2f c3 65 ac 23 38 62 b2 3f bf 71 99 e7 6f 8e c7 b4 5b
13 96 57 75 e3 19 ab aa 7c 74 24 f9 60 1c 9b c3 86 d5 67 41 ff 97
96 60 b6 ca 3f 4f 64 b9 69 17 7f 6f 55 44 7f 41 46 75 f2 d3 46 2b 55
e9 18 0e 67 03 f7 c3 a3 01 17 1d 7c 05 af 03 65 4e 2d b7 1b 9f ae
9c 7a 91 77 1c 32 5b 89 e5 21 c5 a6 48 aa 53 90 f2 b9 8a 32 09 88
63 bd 49 d9 60 7a 32 0b ca 6c 8d dc 0f 88 11 73 3a 05 8e 16 5b 95
1b 25 94 4c 5c 2d c4 29 52 d7 a8 b1 c4 d2 42 64 26 0a f2 07 d1 11
22 90 10 57 d3 1d af fc 3c 29 5e f6 ef 4f 6a 4f 4d 5f de 67 86 bb
5c 92 92 38 48 73 69 17 8d 34 e2 de a1 3e d5 eb 34 b6 b4 ad ae a8
47 99 3a 44 a0 44 d3 96 13 81 92 5b bb 14 b4 08 a3 78 39 c6 7f d2
d0 33 5b 89 e5 21 c5 a6 48 aa 53 90 f2 b9 8a 32 09 88
63 bd 49 d9 60 7a 32 0b ca 6c 8d dc 0f 88 11 73 3a 05 8e 16 5b 95
1b 25 94 4c 5c 2d c4 29 52 d7 a8 b1 c4 d2 42 64 26 0a f2 07 d1 11
22 90 10 57 d3 1d af fc 3c 29 5e f6 ef 4f 6a 4f 4d 5f de 67 86 bb
5c 92 92 38 48 73 69 17 8d 34 e2 de a1 3e d5 eb 34 b6 b4 ad ae a8
47 99 3a 44 a0 44 d3 96 13 81 92 5b bb 14 b4 08 a3 78 39 c6 7f d2
Second Packet
PES Header and PES private field
00 00 01 e0 00 00 8b c1 1b 3d 8c d5 07 17 1d 8c d3 b2 b7 8e 00 01
00 01 00 01 00 00 01 00 01 00 01 00 01 06 0F

Encrypted Payload
inputCtr = 775, streamCtr = 0 (sent in PES Header)
c9 b5 25 ed 9a 83 37 3d b7 87 3f d8 3a 78 50 00 d1 5d d5 48 f7 59
2e bd e9 7e 10 c9 ff a0 68 5f 62 9b 69 08 89 ec f0 c7 31 be 36 c0
1e 83 0c d6 43 33 e6 ce 9e 13