

High-bandwidth Digital Content Protection System

Revision 1.3

21 December, 2006

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.

Copyright © 1999-2006 by Intel Corporation. Third-party brands and names are the property of their respective owners.

Acknowledgement

Silicon Image Inc. has contributed to the development of this specification.

Intellectual Property

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

Contact Information

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

Email: info@digital-cp.com

Web: www.digital-cp.com

Revision History

1 September 99 – 0.80 Revision. Initial publication at Intel Developer Forum
13 October 99 – 0.89 Revision. Publication at Copy Protection Technical Working Group
11 November 99 – 0.90 Revision. Publication at Copy Protection Technical Working Group
11 January 00 – 0.95 Revision. Publication at Copy Protection Technical Working Group
17 February 00 – 1.00 Revision. Publication at Intel Developer Forum
9 June 03 – 1.1 Revision. Publication on DCP LLC web site
13 June 06 – 1.2 Revision. Publication on DCP LLC web site
21 December 06 – 1.3 Revision. Publication on DCP LLC web site

| | | |
|--------------------|---|-----------|
| 1 | Introduction | 4 |
| 1.1 | Scope | 4 |
| 1.2 | Definitions | 5 |
| 1.3 | Overview | 7 |
| 1.4 | Terminology | 8 |
| 1.5 | References | 9 |
| 2 | Authentication | 10 |
| 2.1 | Overview | 10 |
| 2.2 | Protocol | 10 |
| 2.2.1 | First Part of Authentication Protocol | 10 |
| 2.2.2 | Second Part of Authentication Protocol | 12 |
| 2.2.3 | Third Part of Authentication Protocol | 14 |
| 2.3 | HDCP Transmitter State Diagram | 15 |
| 2.4 | HDCP Receiver State Diagram | 20 |
| 2.5 | HDCP Repeater State Diagrams | 21 |
| 2.6 | HDCP Port | 28 |
| 2.7 | Encryption Status Signaling | 33 |
| 3 | Data Encryption | 36 |
| 3.1 | Encryption/Decryption State Diagrams | 37 |
| 3.1.1 | OESS | 37 |
| 3.1.2 | EESS | 40 |
| 4 | HDCP Cipher | 43 |
| 4.1 | Overview | 43 |
| 4.2 | Linear Feedback Shift Register Module | 44 |
| 4.3 | Block Module | 46 |
| 4.4 | Output Function | 49 |
| 4.5 | Operation | 50 |
| 5 | Renewability | 55 |
| 5.1 | SRM Size and Scalability | 55 |
| 5.2 | Updating SRMs | 57 |
| Appendix A. | Test Vectors | 58 |
| Appendix B. | Confidentiality and Integrity of Values | 84 |
| Appendix C. | Sample Algorithm for Ri Verification | 86 |
| Appendix D. | Timing Diagram | 88 |
| Appendix E. | Recommended Method for Synchronous Ri Checking | 89 |

1 Introduction

1.1 Scope

This specification describes the High-bandwidth Digital Content Protection (HDCP) system, Revision 1.30, referred to as HDCP 1.3. HDCP 1.3 is a revision update to HDCP, Revision 1.20 and its errata, referred to collectively as HDCP 1.2.

HDCP 1.3 is designed for protecting Audiovisual content over certain high-bandwidth interfaces, referred to as HDCP-protected Interfaces, from being copied. In HDCP 1.3, the HDCP-protected Interfaces are Digital Visual Interface (DVI), High Definition Multimedia Interface (HDMI), Unified Display Interface (UDI), Giga-bit Video Interface (GVIF) and DisplayPort. This document explains in detail HDCP 1.3 applied to HDMI and DVI. Please refer to HDCP on DisplayPort Specification Rev 1_0 for the HDCP Amendment for DisplayPort. Use of HDCP 1.3 and the HDCP Amendment for DisplayPort are licensed under the terms of the HDCP Adopter Agreement available through DCP, LLC. No other use, right or license to HDCP 1.3 Specification or the HDCP Amendment for DisplayPort is granted by implication, estoppel or otherwise, under any patents, trade secrets or other intellectual property rights now or hereafter owned or controlled by DCP, LLC or its Adopters. In an HDCP System, two or more HDCP Devices are interconnected through an HDCP-protected Interface. The Audiovisual Content protected by HDCP, referred to as HDCP Content, flows from the Upstream Content Control Function into the HDCP System at the most upstream HDCP Transmitter. From there, the HDCP Content, encrypted by the HDCP System, flows through a tree-shaped topology of HDCP Receivers over HDCP-protected Interfaces. This specification describes a content protection mechanism for: (1) authentication of HDCP Receivers to their immediate upstream connection (to an HDCP Transmitter), (2) revocation of HDCP Receivers that are determined by the Digital Content Protection, LLC, to be invalid, and (3) HDCP Encryption of Audiovisual Content over the HDCP-protected Interfaces between HDCP Transmitters and their downstream HDCP Receivers. HDCP Receivers may render the HDCP Content in audio and visual form for human consumption. HDCP Receivers may be HDCP Repeaters that serve as downstream HDCP Transmitters emitting the HDCP Content further downstream to one or more additional HDCP Receivers.

Except when specified otherwise, HDCP 1.3-compliant HDMI Devices must interoperate with other HDCP 1.3, HDCP 1.2 and HDCP 1.1-compliant HDMI Devices attached to their HDCP-protected Interface Ports using the same protocol. Additionally, HDCP 1.3-compliant Devices of which one or more of their HDCP-protected Interface Ports are using the DVI protocol must interoperate with HDCP 1.3, HDCP 1.2, HDCP 1.1 and HDCP 1.0-compliant Devices attached to such HDCP-protected Interface Ports using the DVI protocol. HDCP Transmitters must support HDCP Repeaters.

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

Implementations must include all elements of the content protection system described herein, unless the element is specifically identified as informative or optional. Adopters must also ensure that implementations satisfy the robustness and compliance rules described in the technology license. Additionally, HDCP Transmitters may be subject to additional robustness and compliance rules associated with other content protection technologies.

1.2 Definitions

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

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

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

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

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

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

Enhanced Encryption Status Signaling (EESS). *EESS*, further described in Section 2.7, is a protocol for signaling whether encryption is enabled or disabled for a frame. *EESS* is always used with the HDMI protocol, but is an optional feature with the DVI protocol. See also, *Original Encryption Status Signaling (OESS)*.

frame. For purposes of the HDCP specification, a frame consists of the pixel data between vertical synchronization signals. HDCP may be used with both progressive and interlaced video formats. For interlaced video, every field is an HDCP frame.

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

HDCP 1.0. *HDCP 1.0* refers to, specifically, the variant of HDCP described by Revision 1.00 of this specification along with its associated errata.

HDCP 1.1. *HDCP 1.1* refers to, specifically, the variant of HDCP described by Revision 1.10 of this specification along with its associated errata, if applicable.

HDCP 1.2. *HDCP 1.2* refers to, specifically, the variant of HDCP described by Revision 1.20 of this specification along with its associated errata, if applicable.

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

HDCP 1.0-compliant Device. An HDCP Device that is designed in adherence to HDCP 1.0 is referred to as an *HDCP 1.0-compliant Device*.

HDCP 1.1-compliant Device. An HDCP Device that is designed in adherence to HDCP 1.1 is referred to as an *HDCP 1.1-compliant Device*.

HDCP 1.2-compliant Device. An HDCP Device that is designed in adherence to HDCP 1.2 is referred to as an *HDCP 1.2-compliant Device*.

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

HDCP Content. *HDCP Content* consists of Audiovisual Content that is protected by the HDCP System. *HDCP Content* includes the Audiovisual Content in encrypted form as it is transferred from an HDCP Transmitter to an HDCP Receiver over and 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 Ports 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-protected Interface. An interface for which HDCP applies is described as an *HDCP-protected Interface*. For HDCP 1.0, the only *HDCP-protected Interface* is the Digital Visual Interface (DVI). For HDCP 1.1 and HDCP 1.2, in addition to DVI, the High Definition Multimedia Interface (HDMI) is also an *HDCP-protected Interface*. For HDCP 1.3, DVI, HDMI, Unified Display Interface (UDI), Giga-bit Video Interface (GVIF) and DisplayPort are all HDCP-protected Interfaces. See the References section for further information regarding these *HDCP-protected Interfaces*.

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

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

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

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

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

I²C Bus. A bus linking the HDCP Transmitter and HDCP Receiver compliant with the Video Electronics Standards Association (VESA) Digital Display Channel (DDC) specification subset DDC2B, as required by the Digital Display Working Group (DDWG) Digital Video Interface (DVI) specification.

Key Selection Vector (KSV). Each HDCP Device contains a set of Device Private Keys. A set of Device Private Keys is associated with a *Key Selection Vector (KSV)*. Each HDCP Transmitter has assigned to it a unique *KSV* from all other HDCP Transmitters. Also, each HDCP Receiver has assigned to it a unique *KSV* from all other HDCP Receivers.

Original Encryption Status Signaling (OESS). *OESS*, further described in Section 2.7, is a protocol for signaling whether encryption is enabled or disabled for a frame. *OESS* is only used with the DVI protocol. See also, *Enhanced Encryption Status Signaling (EESS)*.

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

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

In addition, terms such as *Data Island*, *Data Island Period*, *Guard Band*, *Leading Guard Band*, *Trailing Guard Band*, *Video Data*, *Video Data Period*, and *AVMUTE*, are further explained in the HDMI Specification (see references).

1.3 Overview

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

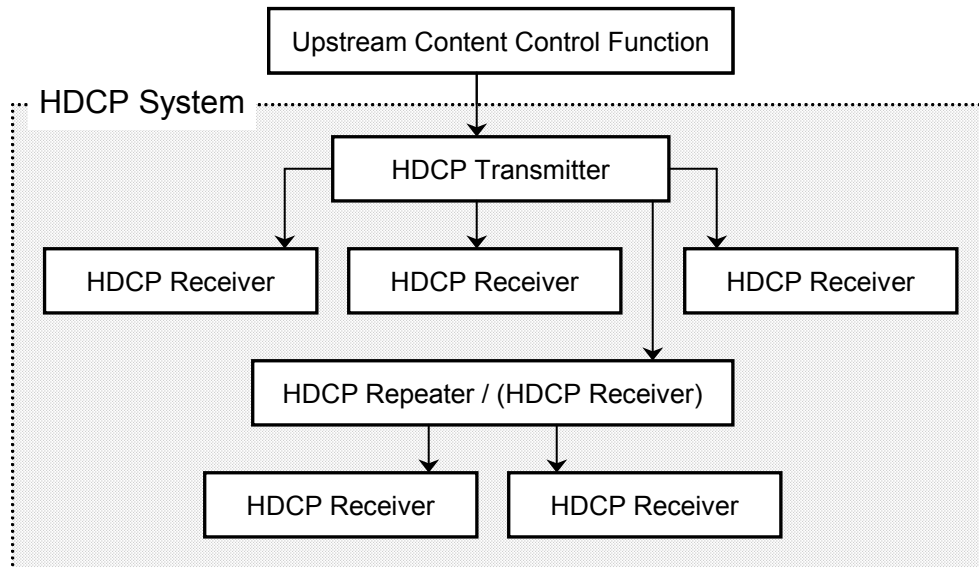


Figure 1-1. Sample Connection Topology of an HDCP System

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

This document contains chapters describing in detail the requirements of each of these elements. In addition, a chapter is devoted describing the cipher that is used in both the authentication protocol and in the encryption of the HDCP Content. All aspects of HDCP map easily onto the existing DVI and HDMI specifications.

1.4 Terminology

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

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

1.5 References

Digital Display Working Group (DDWG), *Digital Visual Interface (DVI) Revision 1.0*, April 2, 1999.

Video Electronics Standards Association (VESA), *Enhanced Display Data Channel (DDC) Standard*, September 2, 1999.

HDMI, LLC., *High-Definition Multimedia Interfaces (HDMI) Revision 1.0*, December XX, 2002.

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

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

Philips Semiconductors, *The I²C-Bus Specification*, Version 2.0, December 1998.

2 Authentication

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

2.1 Overview

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

The HDCP Authentication Protocol can be considered in three parts. The first part establishes shared values between the two HDCP Devices if both devices have a valid Device Key Set from the Digital Content Protection LLC. The second part allows an HDCP Repeater to report the KSVs of attached HDCP Receivers. The third part occurs during the vertical blanking interval preceding each frame for which encryption is enabled, and provides an initialization state for the HDCP Cipher for encrypting the HDCP Content within that frame.

2.2 Protocol

2.2.1 First Part of Authentication Protocol

Figure 2-1 illustrates the first part of the authentication exchange. The HDCP Transmitter (*Device A*) can initiate authentication at any time, even before a previous authentication exchange has completed. Authentication is initiated by the HDCP Transmitter by sending an initiation message containing its KSV (A_{ksv}) and a 64-bit pseudo-random value (A_n) generated by the HDCP Cipher function `hdcpRngCipher` (Section 4.5) to the HDCP Receiver (*Device B*). The HDCP Receiver responds by sending a response message containing the receiver's KSV (B_{ksv}) and the REPEATER bit, which indicates if the receiver is an HDCP Repeater. The HDCP Transmitter verifies that the HDCP Receiver's KSV has not been revoked (section 5), and that the received KSV contains 20 ones and 20 zeros.

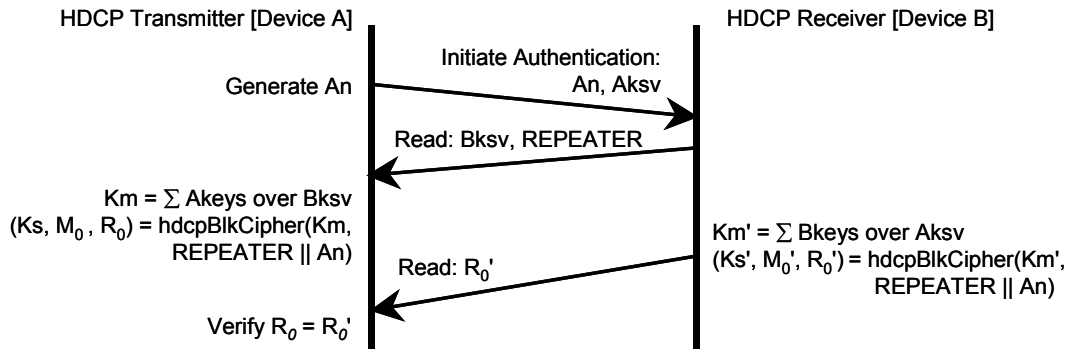


Figure 2-1. First Part of Authentication Protocol

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

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

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

The HDCP Cipher function `hdcpBlockCipher` (Section 4.5) is then used to calculate three values, K_s , M_0 , and R_0 . The cipher initialization values for this calculation are K_m (or K_m'), and the 65-bit concatenation of REPEATER with A_n . The HDCP Receiver's status bit REPEATER indicates that the HDCP Receiver supports retransmission of HDCP Content to additional HDCP Receivers. The session key K_s is a 56-bit secret key for the HDCP Cipher. M_0 is a 64-bit secret value used in the second part of the authentication protocol, and as a supplemental HDCP Cipher initialization value. R_0' is a 16-bit response value that the video receiver returns to the HDCP Transmitter to provide an indication as to the success of the authentication exchange. R_0' must be available for the HDCP Transmitter to read within 100 milliseconds from the time that the HDCP Transmitter finishes writing A_{ksv} to the video receiver. The HDCP Transmitter must not read the R_0' value sooner than 100ms after writing A_{ksv} .

If authentication was successful, then R_0' will be equal to R_0 . If authentication was unsuccessful, then R_0' and R_0 will, in most cases, differ. Future R_i' values, produced during the third part of the authentication protocol, will reveal that authentication has failed in the event that the R_0 values erroneously indicate that authentication was successful.

The HDCP Transmitter enables HDCP Encryption when the first part of the authentication protocol successfully completes.

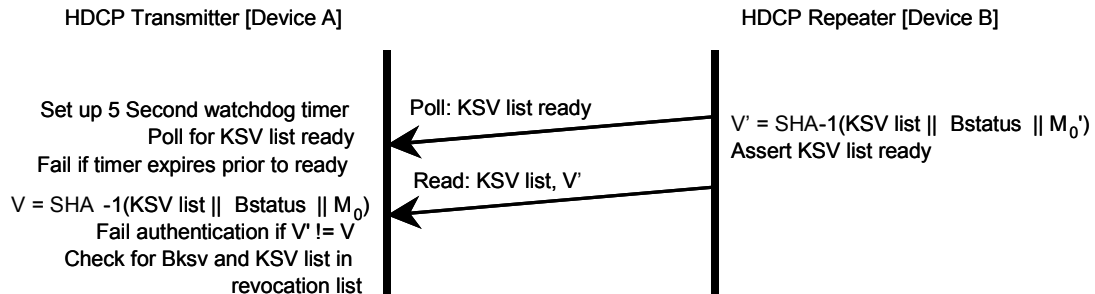


Figure 2-2. Second Part of Authentication Protocol

2.2.2 Second Part of Authentication Protocol

The second part of the authentication protocol (Figure 2-2) is required if the HDCP Receiver is an HDCP Repeater. The HDCP Transmitter executes the second part of the protocol only when the REPEATER bit is set, indicating that the attached HDCP Receiver is an HDCP Repeater. This part of the protocol assembles a list of all downstream KSVs attached to the HDCP Repeater through a permitted connection tree, enabling revocation support upstream.

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

When the HDCP Repeater has assembled the complete list of attached HDCP Devices' KSVs, it computes and appends to the list the verification value V . This value is the SHA-1 hash of the concatenation of the KSV list, $Bstatus$, and the secret value M_0 . When constructing the byte stream for SHA-1 input, the KSV list is in the same little-endian byte order in which it is transmitted over the link, $Bstatus$ is appended in little-endian order, and M_0 is also appended in little-endian order. (See tables A-24 and A-25). When both the KSV list and V are available, the HDCP Repeater asserts its READY status indicator.

The HDCP Transmitter, having determined that the REPEATER bit read earlier in the protocol is set, sets a five-second watchdog timer and polls the HDCP Repeater's READY status bit. When READY is set, the HDCP Transmitter reads the KSV list and V from the HDCP Repeater. If the size of the KSV list exceeds the capacity of the HDCP transmitter, the authentication protocol is aborted. The HDCP Transmitter verifies the integrity of the KSV list by computing the SHA-1 hash value V and comparing this value to V' . If V is not equal to V' , then the authentication protocol is aborted.

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

In addition to assembling the KSV list, an HDCP Repeater propagates topology information upward through the connection tree to the HDCP Transmitter. An HDCP Repeater reports the topology status variables DEVICE_COUNT and DEPTH. The DEVICE_COUNT for an HDCP Repeater is equal to the total number of attached downstream HDCP Receivers and HDCP Repeaters. The value is calculated as the sum of the number of attached downstream HDCP Receivers and HDCP Repeaters plus the sum of the DEVICE_COUNT read from all attached HDCP Repeaters. The DEPTH status for an HDCP Repeater is equal to the maximum number of connection levels below any of the downstream HDCP-protected Interface Ports. The value is calculated as the maximum DEPTH reported from downstream HDCP Repeaters plus one (accounting for the attached downstream HDCP Repeater). For example, an HDCP Repeater with zero downstream HDCP Devices reports a value of zero for both the DEPTH and the DEVICE_COUNT. An HDCP Repeater with four downstream HDCP Receivers that are not HDCP Repeaters reports a DEPTH of one and a DEVICE_COUNT of four. If the computed DEVICE_COUNT for an HDCP Repeater exceeds 127 or the maximum number of devices supported by the size of the KSV FIFO, the HDCP Repeater must assert the MAX_DEVS_EXCEEDED status bit. If the computed DEPTH for an HDCP Repeater exceeds seven, the HDCP Repeater must assert the MAX_CASCADE_EXCEEDED status bit. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED status from a downstream HDCP Repeater, it is required to assert the corresponding status bits to the upstream HDCP Transmitter. If either MAX_CASCADE_EXCEEDED or MAX_DEVS_EXCEEDED status bits are set, the READY bit may be set by the repeater, or it may not set the READY bit and simply let the timeout occur in the HDCP transmitter.

For dual link repeaters, the repeater combines the topology information for both links into a single KSV list that is read from the primary link KSV FIFO. It may at its discretion remove duplicate KSV information in the list. Duplicate KSV values may result from downstream dual link HDCP devices having the same KSV on both links.

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

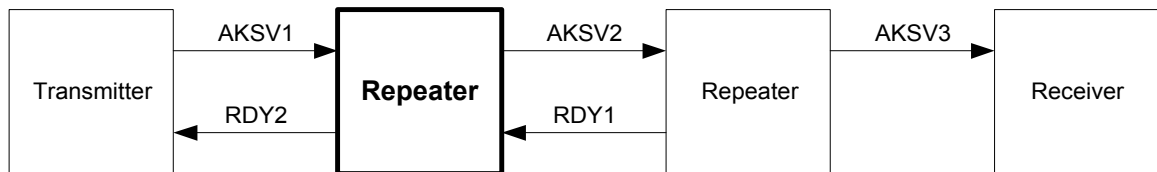


Figure 2-3. Multi-level Repeater Protocol Signals.

| From | To | Max Delay | Conditions and Comments |
|---|---|-----------|---|
| AKSV1 Upstream HDCP Transmitter <i>Aksv</i> received | AKSV2 HDCP Repeater's <i>Aksv</i> transmitted downstream | 100 ms | Downstream propagation time. To latest <i>Aksv</i> transmission when more than one HDCP Receiver is attached. |

| | | | |
|---|--|-------------|--|
| AKSV3 <i>Aksv</i> transmitted to all downstream HDCP-protected Interface Ports | RDY1 Upstream READY asserted | 500 ms | Upstream propagation time when no downstream HDCP Repeaters are attached. (no downstream KSV lists to process) |
| RDY1 Downstream READY asserted | RDY2 Upstream READY asserted | 500 ms | Upstream propagation time when one or more HDCP Repeaters are attached. From latest downstream READY. (downstream KSV lists must be processed) |
| AKSV1 Upstream HDCP Transmitter transmits <i>Aksv</i> | RDY2 Upstream HDCP Transmitter polls asserted READY | 4.2 seconds | For the Maximum of seven repeater levels, 7 * (100 ms + 500 ms) |

Table 2-1. HDCP Repeater Protocol Timing Requirements

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

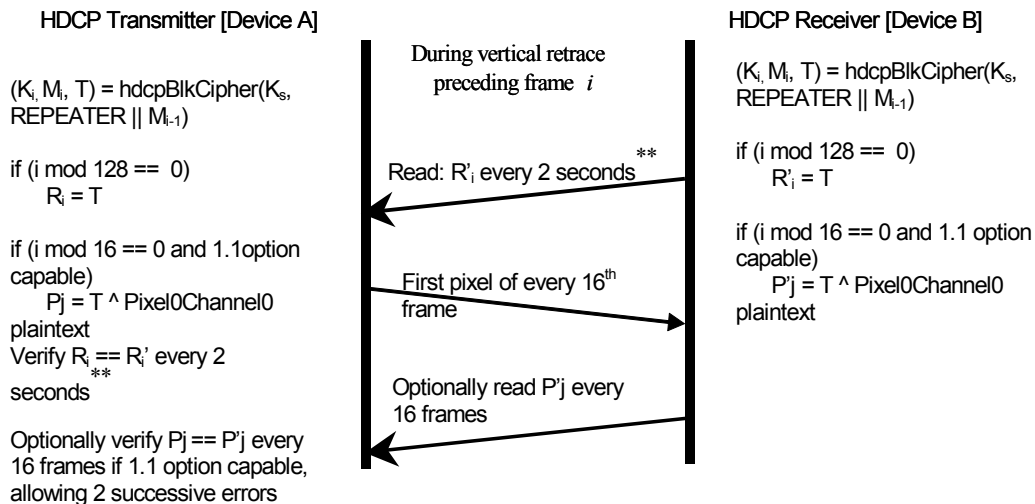


Figure 2-4. Third Part of Authentication Protocol **

2.2.3 Third Part of Authentication Protocol

The third part of the authentication protocol, illustrated in Fig. 2-4, occurs during the vertical blanking interval preceding the frame for which it applies. Each of the two HDCP Devices calculates new cipher initialization values, K_i and M_i , and a third value R_i . The index, i , represents the frame number, starting with the value of one for the first video frame for which encryption is enabled after the completion of the first part of the authentication protocol, and incrementing either on encrypted frames or on every frame,

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

depending upon whether ADVANCE_CIPHER mode is enabled (see below). However, the frame counter does not advance while the HDCP device is in the HDMI AVMUTE state, and does not resume advancing after HDMI AVMUTE state until the first encrypted frame. K_i is a 56-bit key used to initialize the HDCP cipher for encryption or decryption of the HDCP Content. M_i is a new 64-bit initialization value for the HDCP cipher. R_i is a 16-bit value used for link integrity verification, and is updated for every 128th frame counter increment, starting with the 128th. The HDCP Transmitter verifies R_i' against its own calculations to insure that the video receiver is still able to correctly decrypt the information. This verification is made at a minimum rate of once every two seconds. Synchronous reading of R_i every time it changes (every 128th frame) is also acceptable in lieu of asynchronous polling. (Synchronous reading in the frame prior to R_i update and shortly after 1 millisecond of the R_i update also provides a method of detecting frame counter mismatch between HDCP transmitter and HDCP receiver when either device does not support Enhanced Link Verification). It is required that the R_i' read operation complete within 1 millisecond from the time that it is initiated by the HDCP Transmitter. Failure for any reason causes the HDCP Transmitter to consider the HDCP Receiver to be unauthenticated.

In order to enhance the detection of the loss of encryption synchronization, the HDCP Transmitter and Receiver may optionally support Enhanced Link Verification, in which a computation to aid verification of cipher synchronization is performed when a specific video pixel is processed. For every 16th frame counter increment, the decrypted value of channel zero of the first pixel is combined with the least significant byte R_j using the XOR operation, and the result is made available on the P_j port. If this feature is supported by an HDCP Receiver, the *Bcaps* bit 1.1_FEATURES is set and the P_j' port is always updated. The HDCP Transmitter may optionally support reading and verifying the P_j' value against an internally generated P_j value. However, unless a minimum of three successive mismatches of stable values occur, this is considered to be a pixel transmission error and not an authentication or synchronization error. In addition, the mismatched P_j values must be sampled more than once in the same manner as the R_i value. (see Appendix C). Note that the frame counter may advance on unencrypted frames if ADVANCE_CIPHER mode (see below) is enabled, in which case the least significant byte R_j and the pixel data is captured every 16th frame. However, if ADVANCE_CIPHER is not enabled, these values are updated on every 16th encrypted frame.

ADVANCE_CIPHER mode is an optional mode in which the cipher state and frame counter is advanced for every frame in DVI mode, or for every frame when not in AVMUTE state for HDMI mode, regardless of whether encryption is enabled or disabled. The HDCP Receiver indicates this capability by setting the 1.1_FEATURES *Bcaps* bit, and the HDCP Transmitters enables it by setting the ENABLE_1.1_FEATURES bit in the *Ainfo* byte. The frame counter is first updated to a value of 1 when the first ENC_EN is sent or received after authentication, and thereafter incremented every frame, until SET_AVMUTE is set (in HDMI mode). The cipher state is not advanced if HDMI mode AV_MUTE is active, and resumes advancing with the first ENC_EN frame after HDMI CLEAR_AVMUTE.

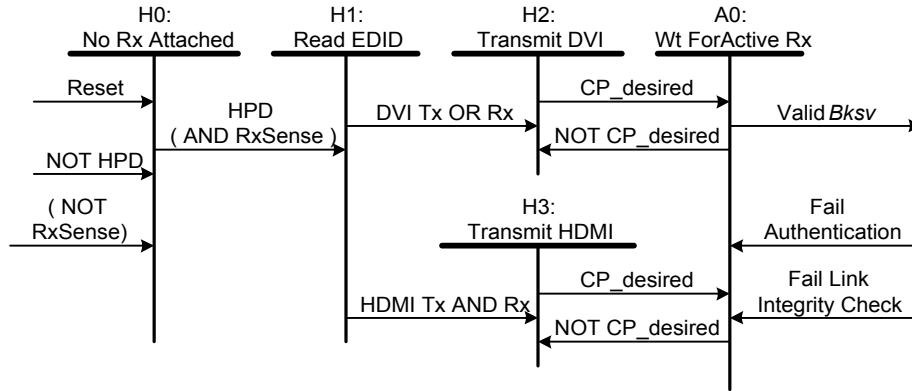
Note: An HDMI-capable HDCP Transmitter that has enabled AC (by writing 1 to ENABLE_1.1_FEATURES bit of AInfo) may need to re-authenticate after sending an AVMUTE to the HDCP Receiver, since the HDCP Receiver may have ignored the HDMI General Control Packet that contained the Set_AVMUTE command, causing the loss of HDCP cipher synchronization.

2.3 HDCP Transmitter State Diagram

The HDCP Transmitter Link State Diagram and HDCP Transmitter Authentication Protocol State Diagram (Figures 2-5 and 2-6) illustrate the operation states of the authentication protocol for an HDCP Transmitter that is not an HDCP Repeater. For HDCP Repeaters, the downstream (HDCP Transmitter) side is covered in Section 2.5.

A variety of events are involved in the transmitter's decision to begin authentication or to transmit video. Some examples of such events may include hot plug detection of an attached HDCP Receiver, completion of certain phases of the operating system, a software request, and mode settings. HDCP receivers are not required to authenticate until presented with a video signal. When an HDCP Receiver acknowledges an I²C register read, it must be ready to

authenticate, and, in the event of authentication failure, must be prepared to process subsequent authentication attempts. The HDCP transmitter should not attempt to authenticate until it has successfully obtained an acknowledged read of an HDCP I²C register. Should the I²C register read or the authentication fail, the HDCP Transmitter must retry periodically, with a period of no more than 2 seconds (preferably much more often). It may cease to attempt authentication only if the HDCP Receiver is clearly disconnected, as with a hot-plug detach.



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

Fig. 2-5 HDCP Transmitter Link State Diagram

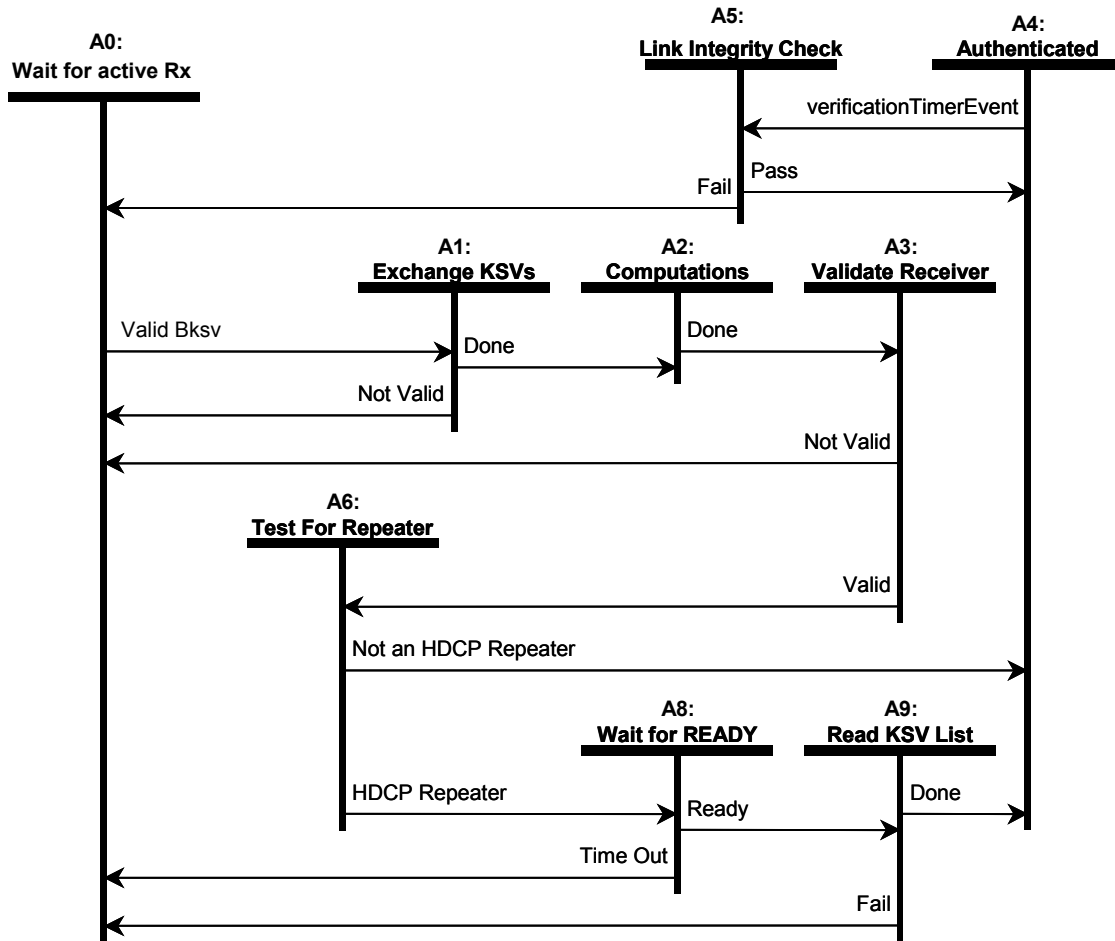


Figure 2-6. HDCP Transmitter Authentication Protocol State Diagram

Transition Any State:H0. Reset conditions at the HDCP Transmitter or loss of Hot Plug Detect (HPD) cause the HDCP Transmitter to enter the No Receiver Attached state. Some TMDS transmitters have the ability to directly sense the presence of a TMDS receiver (Receiver Sense). If this signal is present and indicates the absence of an HDCP Receiver, it should cause a transition to State H0.

Transition H0:H1. The detection of Hot Plug Detect indicates that a sink device is attached and that the EDID ROM is available for reading.

State H1: Read EDID. Whenever HPD is active, the HDCP Transmitter must assume that the receiver is available and displaying an image to the user. Consequently, the transmitter should transmit an image to the screen as soon as feasible. Upon detection of HPD, the HDCP Transmitter reads the EDID to determine the capabilities of the sink device, including whether it is HDMI capable (refer to the HDMI Specification 1.0 for details). If both transmitter and receiver are HDMI-capable, then the TMDS transmitter enters HDMI mode, otherwise, it enters DVI mode.

Transition H1:H2. Immediately after transitioning to DVI mode, the transmitter should begin sending an unencrypted video signal. This signal may be required before the registers of the HDCP Receiver are visible to the HDCP Transmitter.

Transition H1:H3. Immediately after transitioning to HDMI mode, the transmitter should begin sending a video signal. This signal may be required before the registers of the HDCP Receiver are visible to the HDCP Transmitter.

Transition H2:A0, H3:A0. If content protection is desired by the Upstream Content Control Function, then the HDCP Transmitter waits for the availability of an Active HDCP Receiver.

State A0: Wait For Active Receiver. In this state the HDCP Transmitter enables video to a non-HDCP receiver with HDCP Encryption disabled. If no low value content is available or appropriate, a “blue screen” or informative on-screen display should be transmitted. The transmitter must repeatedly attempt to read an HDCP register, at least once every 2 seconds and preferably much more often. Note that in most cases, if the user is watching video from a different input, the Hot Plug Detect signal will still be asserted but the HDCP Receiver’s registers will not be readable. Upon switching to the HDCP input, the user will wish to see valid video within a fraction of a second or they may conclude that the HDCP input is not connected and immediately switch to another input. For this reason, a valid video screen should at all times be transmitted whenever HPD is asserted and authentication should be started immediately after detecting a valid *Bksv* (Transition A0: A1).

Transition A0:A1. The successful read of an I2C HDCP register initiates the authentication protocol.

State A1: Exchange KSVs. In this state, the HDCP Transmitter generates a 64-bit pseudo-random value (*An*) and writes that value to the HDCP Receiver. The transmitter reads the Bcaps register and determines the transport operating mode (DVI or HDMI). If necessary, it sets *Ainfo* in the HDCP Receiver to determine the options that will be in effect prior to writing its KSV (*Aksv*) to the HDCP Receiver. The HDCP Transmitter also reads the HDCP Receiver’s KSV (*Bksv*) and the REPEATER status bit necessary for cipher initialization. Generation of *An* using the HDCP Cipher is described in section 4.5.

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

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

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

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

State A3: Validate Receiver. The HDCP Transmitter reads *R₀'* from the HDCP Receiver and compares it with the corresponding *R₀* produced by the HDCP Transmitter during the computations of State A2. If *R₀* is equal to *R₀'*, then HDCP Encryption is immediately enabled. The verification timer is set up to generate timer events at the nominal rate of once every two seconds, plus or minus one-half second. The HDCP Transmitter must allow the HDCP Receiver up to 100 ms to make *R₀'* available from the time that *Aksv* is written. The HDCP Transmitter also checks the current revocation list for the HDCP Receiver’s KSV *Bksv*. If *Bksv* is in the revocation list, then the HDCP Receiver is considered to have failed the

authentication. Note: checking the revocation list for $Bksv$ may begin as soon as the $Bksv$ has been read in State A1, asynchronously to the other portions of the protocol, but it must complete prior to the transition into the authenticated state (State A4).

The integrity of the current revocation list must be verified by checking the signature of the SRM using the Digital Content Protection LLC public key, as specified in Section 5.

Transition A3:A0. The link integrity message R_0 received from the HDCP Receiver does not match the value calculated by the HDCP Transmitter, or $Bksv$ is in the current revocation list.

Transition A3:A6. The link integrity message R_0 received from the HDCP Receiver matches the expected value calculated by the HDCP Transmitter and $Bksv$ is not in the current revocation list.

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

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

State A5: Link Integrity Check. In this state, the HDCP Transmitter reads R'_i from the HDCP Receiver and compares that value against its value R_i . If the values are not equal, then the HDCP Receiver is incorrectly decrypting the transmitted stream. The R'_i value may be re-read to allow for synchronization and I²C bus errors. The HDCP Transmitter may also check for a loss of synchronization with the HDCP Receiver (see Appendix C). If both the HDCP Transmitter and HDCP Receiver are capable of the Enhanced Link Verification option, the P_j values may be used to check for loss of synchronization more frequently than the R_i check.

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

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

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

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

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

State A8: Wait for Ready. The HDCP Transmitter sets up a five-second watchdog timer and polls the HDCP Receiver's READY bit.

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

Transition A8:A9. The asserted READY signal is received.

State A9: Read KSV List. The watchdog timer is cleared. The HDCP Transmitter reads the list of attached KSVs from the KSV FIFO, reads V' , computes V , and verifies $V == V'$, and the

KSVs from the list are compared against the current revocation list. If the size of the KSV list exceeds the capacity of the HDCP Transmitter, the authentication protocol is aborted.

The integrity of the current revocation list must be verified by checking the signature of the SRM using the Digital Content Protection LLC public key, as specified in Section 5.

Transition A9:A0. This transition is made if $V \neq V'$, [verification of the SRM fails,] or if any of the KSVs in the list are found in the current revocation list. A retry of the entire KSV FIFO read operation may be implemented if $V \neq V'$. Two additional status bits cause this transition when asserted. These are MAX_CASCADE_EXCEEDED and MAX_DEVS_EXCEEDED.

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

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

2.4 HDCP Receiver State Diagram

The operation states of the authentication protocol for an HDCP Receiver that is not an HDCP Repeater are illustrated in Figure 2-7. For HDCP Repeaters, the upstream (HDCP Receiver) side is covered in Section 2.5.

The Fast Re-authentication capability indication alerts the HDCP Transmitter that it may re-authenticate by writing a new Aksv and An without first resetting the HDCP Receiver by powering off the TMDS buffers for 100 ms. When set to 1, or the receiver is HDMI capable, no reset is necessary, and the receiver is capable of starting a new authentication if it receives a new Aksv in the B0:B1, B1:B2 and B2:B1 transitions as well as the B0, B1 and B2 states. If set to zero and the receiver is not HDMI capable, it may be necessary to reset the receiver prior to writing Aksv, which, in TMDS applications, may be effected by the HDCP Transmitter by powering off the TMDS buffers for a period of 100 ms.

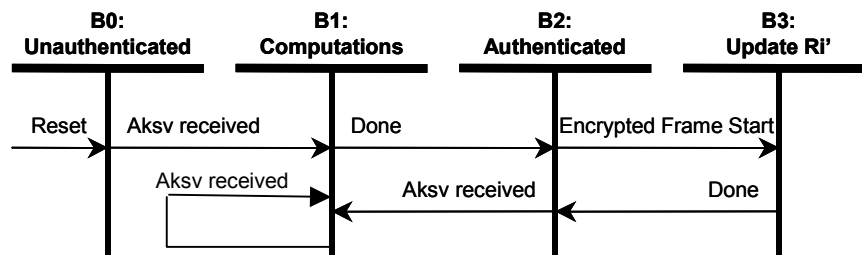


Figure 2-7. HDCP Receiver Authentication State Diagram

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

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

Transition B0:B1. The final byte of *Aksv* is received from the HDCP Transmitter. *Ainfo* is copied from the port value and takes effect, and the port value is cleared to its default value of zero.

State B1: Computations. In this state, the HDCP Receiver calculates the values Km' , Ks' , M_0' , and R_0' using the HDCP Receiver's Device Private Keys and the received values of *An* and *Aksv*. The HDCP Receiver must complete the computations within 100 milliseconds and make R_0' available to the HDCP Transmitter.

Transition B1: B1. Should the HDCP Transmitter write the *Aksv* while the HDCP Receiver is in State B1, the HDCP Receiver abandons intermediate results and restarts the computations. *Ainfo* is copied from the port value and takes effect, and the port value is cleared to its default value of zero.

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

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

Transition B2:B1. A new authentication is forced any time the *Aksv* is written by the attached HDCP Transmitter. *Ainfo* is copied from the port value and takes effect, and the port value is cleared to its default value of zero.

Transition B2:B3. This transition is made during the vertical blank interval preceding encrypted frames. The third part of the authentication protocol requires periodic updates to the R_i' value.

State B3: Update R_i' . During the vertical blank interval preceding each encrypted frame the HDCP Receiver determines whether or not to update the response value R_i' with HDCP Cipher output value available during the frame key calculation. The R_i' value is updated when $(i \bmod 128 == 0)$. The updated R_i' value must be available through the HDCP-protected Interface Port no more than 128 pixel clocks from the time that encryption enable is indicated for the next frame over the CTLx signals. Section 2.7 specifies encryption enable signaling. Also, if the HDCP Receiver indicates it is capable of the Enhanced Link Verification option, it will similarly make a new P_j available if $(j \bmod 16 == 0)$ within 128 pixel clocks after it receives the first pixel of the frame.

Transition B3:B2. Once R_i' has been updated, return to the authenticated state.

2.5 HDCP Repeater State Diagrams

The HDCP Repeater has one HDCP-protected Interface connection to an upstream HDCP Transmitter and one or more HDCP-protected Interface connections to downstream HDCP Receivers as permitted in the Digital Content Protection LLC license. The state diagram for each downstream connection (Figures 2-8 and 2-9) is substantially the same as that for the host HDCP Transmitter (Section 2.3), with two exceptions. First, the HDCP Repeater is not required to check for downstream KSVs in a revocation list. Second, the HDCP Repeater initiates authentication downstream when it receives an authentication request from upstream, rather than at detection of an HDCP Receiver on the downstream HDCP-protected Interface Port. The HDCP Repeater signals the detection of an active downstream HDCP Receiver to the upstream HDCP Transmitter by pulsing the Hot Plug Detect signal of the upstream HDCP-protected Interface Port. The pulse width must be greater than 100 ms. In this state diagram and its following description, the downstream (HDCP Transmitter) side refers to the

HDCP Transmitter functionality within the HDCP Repeater for its corresponding downstream HDCP-protected Interface Port.

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

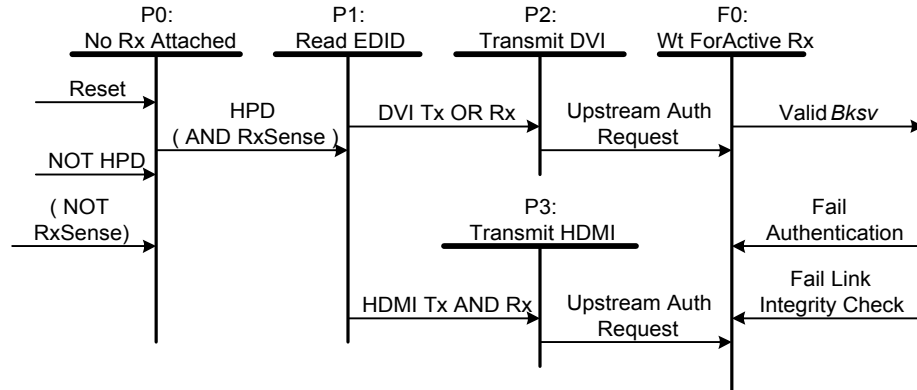


Figure 2-8 HDCP Repeater Downstream Link State Diagram

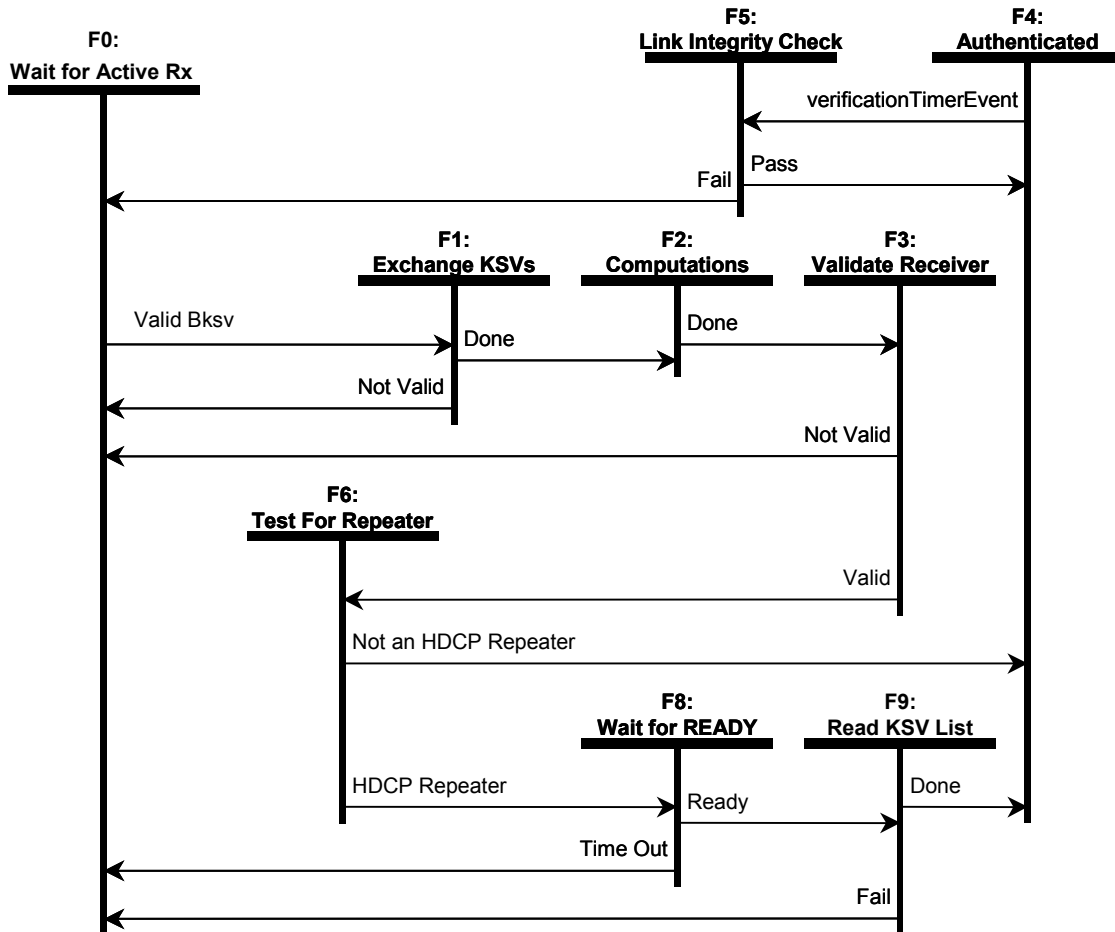


Figure 2-9. HDCP Repeater Downstream Authentication Protocol State Diagram

Transition Any State:P0. Reset conditions at the HDCP Repeater or loss of Hot Plug Detect (HPD) cause the HDCP Repeater to enter the No Receiver Attached state for this port. Some TMDS transmitters have the ability to directly sense the presence of a TMDS receiver (Receiver Sense). If this signal is present and indicates the absence of an HDCP Receiver, it should cause a transition to State P0.

Transition P0:P1. The detection of Hot Plug Detect indicates that a sink device is attached and that the EDID ROM is available for reading.

State P1: Read EDID. Whenever HPD is active, the HDCP Repeater must assume that the receiver is available and displaying an image to the user. Consequently, the transmitter should transmit an image to the screen as soon as feasible. Upon detection of HPD, the HDCP Transmitter reads the EDID to determine the capabilities of the sink device, including whether it is HDMI capable (refer to the HDMI Specification 1.0 for details). If both transmitter and receiver are HDMI-capable, then the TMDS transmitter enters HDMI mode, otherwise, it enters DVI mode.

Transition P1:P2. Immediately after transitioning to DVI mode, the transmitter should begin sending an unencrypted video signal. This signal may be required before the registers of the HDCP Receiver are visible to the HDCP Repeater.

Transition P1:P3. Immediately after transitioning to HDMI mode, the transmitter should begin sending a video signal. This signal may be required before the registers of the HDCP Receiver are visible to the HDCP Repeater.

Transition P2:F0, P3:F0. Upon an Upstream Authentication Request, the HDCP Repeater waits for the availability of an Active HDCP Receiver on this port.

State F0: Wait For Active Receiver. In this state the HDCP Repeater must not decrypt or encrypt video. If no low value content is available or appropriate, a “blue screen” or informative on-screen display should be transmitted. The HDCP Repeater must repeatedly attempt to read an HDCP register, at least once every 2 seconds and preferably much more often. Note that in most cases, if the user is watching video from a different input, the Hot Plug Detect signal will still be asserted but the HDCP Receiver’s registers will not be readable. Upon switching to the HDCP input, the user will wish to see valid video within a fraction of a second or they may conclude that the HDCP input is not connected and immediately switch to another input. For this reason, a valid video screen should at all times be transmitted whenever Hot Plug Detect is asserted, and downstream authentication should be started immediately after detecting a valid $Bksv$ (Transition F0: F1).

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

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

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

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

State F3: Validate Receiver. The downstream (HDCP Transmitter) side reads R_0' from the HDCP Receiver and compares it with the corresponding R_0 produced by itself during the computations of State F2, then immediately enables data encryption if R_0' is equal to R_0 . The HDCP Receiver must make R_0' available less than 100 ms from the time that $Aksv$ is written. The HDCP Transmitter must not attempt to read R_0' sooner than this 100 ms. The HDCP Receiver’s $Bksv$ is added to the KSV list for this HDCP Repeater.

Transition F3:F0. The link integrity message R_0' received from the HDCP Receiver does not match the value calculated by the downstream (HDCP Transmitter) side.

Transition F3:F6. The link integrity message R_0' received from the HDCP Receiver matches the expected value calculated by the downstream (HDCP Transmitter) side.

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

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

State F5: Link Integrity Check. In this state, the downstream (HDCP Transmitter) side reads R_i' from the HDCP Receiver and compares that value against its value R_i . If the values are equal, then the HDCP Receiver is correctly decrypting the transmitted stream. The R_i' value may be re-read to allow for synchronization and I²C bus errors. The HDCP Transmitter may also check for a loss of synchronization with the HDCP Receiver (see Appendix C). If both the HDCP Transmitter and HDCP Receiver are capable of the Enhanced Link Verification option, the P_j values may be used to check for loss of synchronization more frequently than the R_i check.

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

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

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

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

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

State F8: Wait for Ready. The downstream (HDCP Transmitter) side sets up a five-second watchdog timer and polls the HDCP Receiver's READY bit.

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

Transition F8:F9. The asserted READY signal is received.

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

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

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

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

The HDCP Repeater upstream state diagram, illustrated in Figure 2-10, makes reference to states of the HDCP Repeater downstream state diagram.

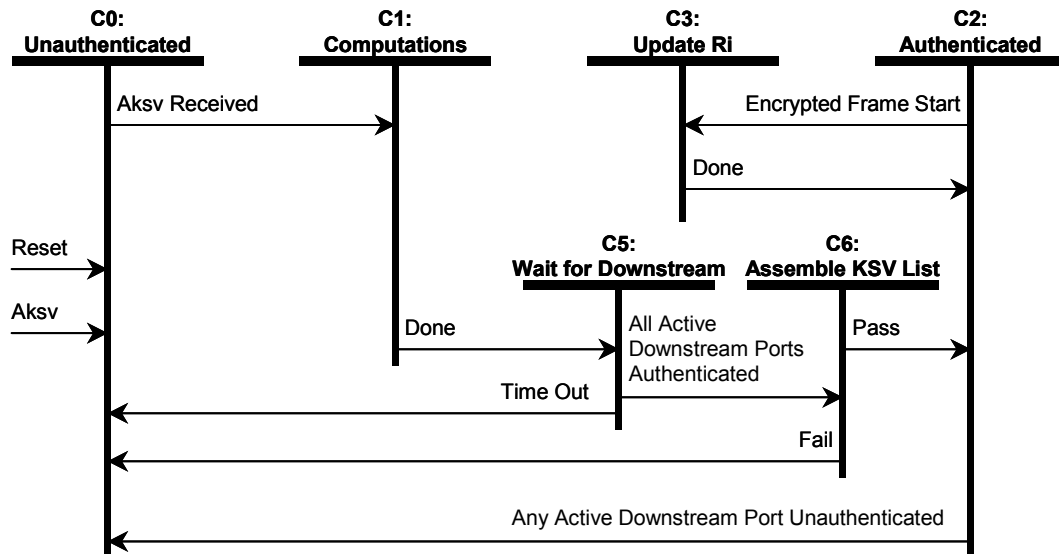


Figure 2-10. HDCP Repeater Upstream Authentication Protocol State Diagram

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

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

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

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

Transition C1:C5. The computations are complete and the results are available for reading by the HDCP Transmitter.

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

Transition C2:C0. The upstream (HDCP Receiver) connection becomes unauthenticated if any downstream HDCP Port enters the unauthenticated state OR if a downstream HDCP-protected Interface Port that previously had no active downstream HDCP Receiver attached senses an attached active HDCP Receiver.

NOTE: Optionally, the upstream connection need not enter into an unauthenticated state, if an attached HDCP receiver is disconnected from the downstream HDCP-protected Interface Port. Also, if an authenticated HDCP receiver attached to the downstream HDCP-protected Interface Port is disconnected and reconnected (i.e. the downstream HDCP Port sees the same KSV stored in its KSV list at the HDCP protected interface port), the upstream connection need not enter into an unauthenticated state. The downstream HDCP Port only needs to re-authenticate the attached HDCP receiver.

Transition C2:C3. This transition is made during the vertical blank interval preceding encrypted frames. The third part of the authentication protocol requires periodic updates to the R_i' value.

State C3: Update R_i' . During the vertical blank interval preceding each encrypted frame the HDCP Repeater determines whether or not to update the response value R_i' with HDCP Cipher output value available during the frame key calculation. The R_i' value is updated when $(i \bmod 128) = 0$. The updated R_i' value must be available through the HDCP-protected Interface Port no more than 128 pixel clocks from the time that encryption enable is indicated for the next frame over the CTLx signals. Section 2.7 specifies encryption enable signaling.

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

State C5: Wait for Downstream. The upstream (HDCP Receiver) state machine waits for all downstream HDCP-protected Interface Ports of the HDCP Repeater to enter either the unconnected (State P0), inactive (State F0), or the authenticated state (State F4).

Transition C5:C0. The watchdog timer expires before all downstream HDCP-protected Interface Ports enter the authenticated or unconnected state.

Transition C5:C6. All downstream HDCP-protected Interface Ports with attached HDCP Receivers have reached the state of authenticated or unconnected.

State C6: Assemble KSV List. The HDCP Repeater assembles the list of all attached downstream topology HDCP Devices as the downstream HDCP-protected Interface Ports reach terminal states of the authentication protocol. An HDCP-protected Interface Port that advances to State P0, the unconnected state, or F0, the inactive state, does not add to the list. A downstream HDCP-protected Interface Port that arrives in State F4 that has an HDCP Receiver that is not an HDCP Repeater attached, adds the B_{ksv} of the attached HDCP Receiver to the list. Downstream HDCP-protected Interface Ports that arrive in State F4 that have an HDCP Repeater attached will cause the KSV list read from the attached HDCP Repeater, plus the B_{ksv} of the attached HDCP Repeater itself, to be added to the list. The HDCP Repeater must verify the integrity of the downstream HDCP Repeater's list by computing V and checking this value against V' received from the attached HDCP Repeater. If V does not equal V' , the downstream KSV list integrity check fails. A retry of the entire KSV FIFO read operation should be performed if $V \neq V'$. When the KSV list for all downstream HDCP Receivers has been assembled, the HDCP Repeater computes the upstream V' .

The DEVICE_COUNT for an HDCP Repeater is equal to the total number of attached downstream HDCP Receivers (including HDCP Repeaters). The value is calculated as the sum of the number of attached active HDCP Receivers (including HDCP Repeaters) plus the sum of the DEVICE_COUNT of all attached HDCP Repeaters. The DEPTH for an HDCP Repeater is equal to the maximum number of connection levels below any of the downstream HDCP-protected Interface Ports. The value is calculated as the maximum DEPTH reported from downstream HDCP Repeaters plus one (accounting for the attached downstream HDCP Repeater). If the computed DEVICE_COUNT for an HDCP Repeater exceeds 127 or the size of the KSV_FIFO, the HDCP Repeater must assert the MAX_DEVS_EXCEEDED status bit. If the computed DEPTH for an HDCP Repeater exceeds seven, the HDCP Repeater must

assert the MAX_CASCADE_EXCEEDED status bit. When an HDCP Repeater receives a MAX_DEVS_EXCEEDED or a MAX_CASCADE_EXCEEDED status from a downstream HDCP Repeater, it is required to assert its corresponding upstream status bit.

Transition C6:C0. If any downstream HDCP-protected Interface Port should transition to the unauthenticated state, the upstream connection transitions to the unauthenticated state. This transition is also made when any downstream HDCP-protected Interface Ports becomes unauthenticated, or when the KSV list integrity check for a downstream HDCP Repeater fails.

NOTE: Optionally, the upstream connection need not enter into an unauthenticated state, if an attached HDCP receiver is disconnected from the downstream HDCP-protected Interface Port. Also, if an authenticated HDCP receiver attached to the downstream HDCP-protected Interface Port is disconnected and reconnected (i.e. the downstream HDCP Port sees the same KSV stored in its KSV list at the HDCP protected interface port), the upstream connection need not enter into an unauthenticated state. The downstream HDCP Port only needs to re-authenticate the attached HDCP receiver.

Transition C6:C2. The KSV list and V' , as well as DEVICE_COUNT and DEPTH, are ready for reading by the upstream HDCP Transmitter.

2.6 HDCP Port

The values that must be exchanged between the HDCP Transmitter and the HDCP Receiver are communicated over the I²C serial interface of the HDCP-protected Interface. The HDCP Receiver must present a logical device on the I²C bus for each link that it supports. No equivalent interface to HDCP Transmitters is specified. The eight-bit I²C device address (including the read/write bit, “x”) for the primary link is 0111010x binary, or 0x74 in the usual hexadecimal representation of I²C device addresses where the read/write bit is set to zero. The device address for the secondary link is 0x76. Table 2–2 and Table 2–3 specify the address space for these devices, which act only as slaves on the I²C bus. Multi-byte values are stored in little-endian format.

Read and write operations must complete within 100 ms per byte transferred. Master devices may elect to repeat any transfers believed to have previously completed with errors.

| Offset (hex) | Name | Size in Bytes | Rd/Wr | Function |
|--------------|------------------------|---------------|-------|---|
| 0x00 | <i>Bksv</i> | 5 | Rd | HDCP Receiver KSV. This value may be used to determine that the receiver is HDCP capable. Valid KSVs contain 20 ones and 20 zeros, a characteristic that must be verified by HDCP Transmitters before encryption is enabled. This value must be available any time the HDCP Receiver's HDCP hardware is ready to operate. |
| 0x05 | Rsvd | 3 | Rd | All bytes read as 0x00 |
| 0x08 | <i>R_i'</i> | 2 | Rd | Link verification response. Upon completion of the authentication computations, this register contains the <i>R₀'</i> value. Following that, it is updated upon completion of HDCPBlockCipher if $(i \bmod 128) == 0$. It is recommended that HDCP Transmitters protect against errors in the I ² C transmission by re-reading this value when unexpected values are received, though care must be taken to avoid missing legitimate mis-match conditions. This value must be available at all times between updates. <i>R₀'</i> must be available less than 100 ms after <i>Ak_{sv}</i> is received. Subsequent <i>R_i'</i> values must be available a maximum of 128 pixel clocks following the Encryption Enable detection (ENC_EN). |
| 0x0A | <i>P_j'</i> | 1 | Rd | Enhanced Link Verification Response. Updated upon receipt of first video pixel received when frame counter value $(j \bmod 16) == 0$. The value is the XOR of the decrypted byte on channel zero of the first video pixel with the least significant byte of <i>R_j</i> . <i>R_j</i> is derived from the output function in the same manner as <i>R_i</i> , but is captured every 16 th counted frame (rather than every 128 th counted frame). |
| 0x0B | Rsvd | 5 | Rd | All bytes read as 0x00 |
| 0x10 | <i>Ak_{sv}</i> | 5 | Wr | HDCP Transmitter KSV. Writes to this multi-byte value are written least significant byte first. The final write to 0x14 triggers the authentication sequence in the HDCP Receiver, and the current <i>Ainfo</i> value is copied from the port, takes effect, and the port is reset to the default value of zero. |
| 0x15 | <i>Ainfo</i> | 1 | Wr | Bits 7-2: Reserved zeros. Bit 1: ENABLE_1.1_FEATURES. This bit enables the Advance Cipher option. If in DVI mode, it also enables the Enhanced Encryption Status Signaling (EESS) (in HDMI mode, EESS is enabled regardless of this bit setting). This bit resets to default zero when the HDCP Receiver becomes attached or active, or is reset, or the last byte of <i>Ak_{sv}</i> is written. A write to the last byte of <i>Ak_{sv}</i> copies the port value and causes it to take effect, and then resets the port value to the default value of zero. Thus the options must be explicitly enabled prior to each authentication. Bit 0: Reserved (must be zero). |
| 0x16 | Rsvd | 2 | Rd | All bytes read as 0x00 |
| 0x18 | <i>An</i> | 8 | Wr | Session random number. This multi-byte value must be written by the HDCP Transmitter before the KSV is written. |
| 0x20 | <i>V'.H0</i> | 4 | Rd | H0 part of SHA-1 hash value used in the second part of the authentication protocol for HDCP Repeaters. (NOTE: port 0x20 is the least significant byte of the H0 value, as all ports are little-endian byte order). |
| 0x24 | <i>V'.H1</i> | 4 | Rd | H1 part of SHA-1 hash value <i>V'</i> . |
| 0x28 | <i>V'.H2</i> | 4 | Rd | H2 part of SHA-1 hash value <i>V'</i> . |
| 0x2c | <i>V'.H3</i> | 4 | Rd | H3 part of SHA-1 hash value <i>V'</i> . |

| | | | | |
|------|----------------|-----|-----------|--|
| 0x30 | <i>V'.H4</i> | 4 | Rd | H4 part of SHA-1 hash value <i>V'</i> . |
| 0x34 | Rsvd | 12 | Rd | All bytes read as 0x00 |
| 0x40 | <i>Bcaps</i> | 1 | Rd | <p>Bit 7: HDMI_RESERVED Use of this bit is reserved. HDCP Receivers not capable of supporting HDMI must clear this bit to 0.</p> <p>Bit 6: REPEATER, HDCP Repeater capability. When set to one, this HDCP Receiver supports downstream connections as permitted by the Digital Content Protection LLC license. This bit does not change while the HDCP Receiver is active.</p> <p>Bit 5: READY, KSV FIFO ready. When set to one, this HDCP Repeater has built the list of attached KSVs and computed the verification value <i>V'</i>. This value is always zero during the computation of <i>V'</i>. See states C0 and C2.</p> <p>Bit 4: FAST. When set to one, this device supports 400 KHz transfers. When zero, 100 KHz is the maximum transfer rate supported. Note that 400KHz transfers are not permitted to any device unless all devices on the I²C bus are capable of 400KHz transfer. The transmitter may not be able to determine if the EDID ROM, present on the HDCP Receiver, is capable of 400KHz operation. This bit does not change while the HDCP Receiver is active.</p> <p>Bits 3-2: Reserved (must be zero).</p> <p>Bit 1: 1.1_FEATURES. When set to one, this HDCP Receiver supports Enhanced Encryption Status Signaling (EESS), Advance Cipher, and Enhanced Link Verification options. For the HDMI protocol, Enhanced Encryption Status Signaling (EESS) capability is assumed regardless of this bit setting. This bit does not change while the HDCP Receiver is active.</p> <p>Bit 0: FAST_REAUTHENTICATION. When set to 1, the receiver is capable of receiving (unencrypted) video signal during the session re-authentication. All HDMI-capable receivers shall be capable of performing the fast re-authentication even if this bit is not set. This bit does not change while the HDCP Receiver is active.</p> |
| 0x41 | <i>Bstatus</i> | 2 | Rd | Refer to Table 2–4 for definitions. |
| 0x43 | KSV FIFO | 1 | Rd | <p>Key selection vector FIFO. Used to pull downstream KSVs from HDCP Repeaters. All bytes (DEVICE_COUNT * 5) must be read in a single, auto-incrementing access.</p> <p>All bytes read as 0x00 for HDCP Receivers that are not HDCP Repeaters (REPEATER == 0).</p> |
| 0x44 | Rsvd | 124 | Rd | All bytes read as 0x00 |
| 0xC0 | dbg | 64 | Rd/ Wr | Implementation-specific debug registers. Confidential values must not be exposed through these registers. |

Table 2–2. Primary Link HDCP Port (I²C device address 0x74)

| Offset (hex) | Name | Size (Bytes) | Rd/Wr | Function |
|--------------|--------------|--------------|-------|--|
| 0x00 | <i>Bksv</i> | 5 | Rd | HDCP Receiver KSV. See primary link comments. This value may match the value of <i>Bksv</i> for the primary link. |
| 0x05 | Rsvd | 3 | Rd | All bytes read as 0x00 |
| 0x08 | <i>Ri'</i> | 2 | Rd | Link verification response. See primary link comments. This value will usually differ from the value of <i>Ri'</i> for the primary link. |
| 0x0A | <i>Pj'</i> | 1 | Rd | Enhanced Link Verification Response. See primary link comments. This value will usually differ from the value of <i>Ri'</i> for the primary link. |
| 0x0B | Rsvd | 5 | Rd | All bytes read as 0x00 |
| 0x10 | <i>Aksv</i> | 5 | Wr | HDCP Transmitter KSV. See primary link comments. This value may be programmed to the same value of <i>Aksv</i> for the primary link. |
| 0x15 | <i>Ainfo</i> | 1 | Wr | Bits 7-2: Reserved zeros. Bit 1: ENABLE_1.1_FEATURES. This bit enables the Advance Cipher option. If in DVI mode, it also enables the Enhanced Encryption Status Signaling (EESS) (in HDMI mode, EESS is enabled regardless of this bit setting). This bit resets to default zero when the HDCP Receiver becomes attached or active, or is reset, or the last byte of <i>Aksv</i> is written. A write to the last byte of <i>Aksv</i> copies the port value and causes it to take effect, and then resets the port value to the default value of zero. Thus the options must be explicitly enabled prior to each authentication. Bit 0: Reserved (must be zero). |
| 0x16 | Rsvd | 2 | Rd | All bytes read as 0x00 |
| 0x18 | <i>An</i> | 8 | Wr | Session random number. See primary link comments. This value must differ from the programmed value of <i>An</i> for the primary link. |
| 0x20 | Undef | 160 | Rd | These bytes may assume any values as an artifact of implementation, but must not be used by implementations. Confidential values must not be exposed through these registers. |
| 0xC0 | dbg | 64 | Rd/Wr | Implementation-specific debug registers. Confidential values must not be exposed through these registers. |

Table 2–3. Secondary Link HDCP Port (I²C device address 0x76)

Note: There is no separate KSV FIFO for the secondary link. Dual link repeaters combine their downstream KSV lists into a single list that is read on the primary link. They may remove duplicate KSV values that result from downstream HDCP devices using the same KSV on both links.

| Name | Bit Field | Rd/Wr | Description |
|-----------------------|-----------|-------|---|
| Rsvd | 15:14 | Rd | Reserved. Read as zero. |
| HDMI_RESERVED_2 | 13 | Rd | Reserved for future possible HDMI use. |
| HDMI_MODE | 12 | Rd | HDMI Mode. When set to one, the HDCP Receiver has transitioned from DVI Mode to HDMI Mode. This has occurred because the HDCP Receiver has detected HDMI bus conditions on the link. This bit must not be cleared when the HDCP Transmitter and HDCP Receiver are connected and both are operating in an active HDMI mode. This bit must be cleared upon power-up, reset, unplug or plug of an HDCP Transmitter or anytime that the HDCP Receiver has not seen at least one Data Island within 30 video frames. |
| MAX_CASCADE_EXCEEDED. | 11 | Rd | Topology error indicator. When set to one, more than seven levels of video repeater have been cascaded together. |
| DEPTH | 10:8 | Rd | Three-bit repeater cascade depth. This value gives the number of attached levels through the connection topology. |
| MAX_DEVS_EXCEEDED | 7 | Rd | Topology error indicator. When set to one, more than 127 downstream devices, or the capacity of the KSV Fifo, are attached. |
| DEVICE_COUNT | 6:0 | Rd | Total number of attached downstream devices. Always zero for HDCP Receivers. This count does not include the HDCP Repeater itself, but only downstream devices downstream from the HDCP Repeater. |

Table 2–4. Bstatus Register Bit Field Definitions

The HDCP Receivers at these slave addresses respond to I²C accesses as diagrammed in Figure 2-11, Figure 2-12, and Figure 2-13. The nomenclature within these diagrams, and used to describe them, is the same as found in *The I²C Bus Specification Version 2.0*.

Figure 2-11 illustrates a combined-format byte read, in which the master writes a one-byte address to the slave, followed by a repeated start condition (Sr) and the data read. HDCP Devices must support multi-byte reads with auto-increment. For reads beginning with the KSV FIFO address, the bytes will increment through the KSV FIFO data. For all other reads, the bytes will increment through the port addresses, reading one byte for each offset. Auto-incremented sequential accesses that start before the KSV FIFO address and cross through the KSV FIFO address read only the first byte of the KSV FIFO and then continue incrementing through the HDCP port address space.

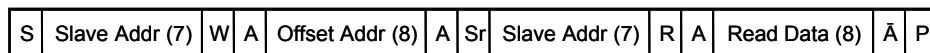


Figure 2-11. HDCP Port Combined-Format Byte Read

Figure 2-12 illustrates a byte write access. As for combined-format read accesses, the HDCP port must support multi-byte writes with auto-increment, again with an exception for KSV FIFO writes where the implicit address increment moves through the KSV FIFO data structure rather than through the HDCP port address space. Auto-incremented sequential accesses that start before the KSV FIFO address and cross through the KSV FIFO address access only the first byte of the KSV FIFO and then continue incrementing through the HDCP port address space.



Figure 2-12. HDCP Port Byte Write

In order to minimize the number of bits that must be transferred for the link integrity check, a second read format must be supported by all HDCP Receivers and by HDCP Transmitters that do not implement a hardware I²C master. This access, shown in Figure 2-13, has an implicit offset address equal to 0x08, the starting location for R_i' . The short read format may be uniquely differentiated from combined reads by tracking STOP conditions (P) on the bus. Short reads must be supported with auto-incrementing addresses.

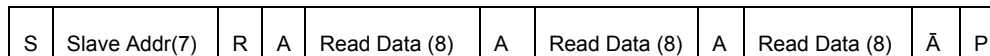


Figure 2-13. HDCP Port Link Integrity Message Read

2.7 Encryption Status Signaling

The HDCP Transmitter signals the HDCP Receiver to begin the third part of the authentication protocol through the previously reserved control signals CTL3, CTL2, CTL1, and CTL0 of the HDCP-protected Interface. Two different protocols for signaling are supported. One is termed *Original Encryption Status Signaling* (OESS) while the other is termed *Enhanced Encryption Status Signaling* (EESS). These protocols are only used when the HDCP Device is in an authenticated state. However, since an HDCP Transmitter may become unauthenticated with no immediate downstream indication, an HDCP Receiver may not be aware of this change and will continue to expect encryption signaling. Therefore it is highly recommended that the HDCP Transmitter not signal frame encryption while in the unauthenticated state. In the case of prior EESS signaling, it is recommended that the encryption-disabled signaling continue (rather than no encryption signaling), ensuring that the HDCP receiver properly displays the blue screen, informative display, or low value content which is sent while the HDCP Transmitter is in an unauthenticated state and the HDCP Receiver is still in an authenticated state. Authenticated states for HDCP Transmitters are State A4 and A5. Authenticated states for HDCP Receivers are State B2 and B3. Authenticated states for HDCP Repeaters are State C2 and C3.

The decision of which Encryption Status Signaling to use may only occur while both HDCP Devices are in an unauthenticated state, typically upon detection of a newly active or connected HDCP Receiver. Any transition to State A0 of Figure 2-4, State B0 of Figure 2-7, State F0 of Figure 2-9, or State C0 of Figure 2-10 causes the HDCP Device to assume DVI protocol and OESS will be used. Before the authentication process may start, the HDCP Transmitter must determine whether EESS may be used.

Original Encryption Status Signaling (OESS)

OESS utilizes only CTL3, and is only used during DVI protocol. This signaling is accomplished with a single high-going pulse, during the vertical blanking interval, of sufficient width that it may be distinguished from bit errors on the channel or any effects due to resynchronization events in the receiver. The transmitter must assert CTL3 for *at least* 8 pixel clocks (the transmitter is encouraged to utilize a wider pulse to enhance reliability, but no more than the entire Vsync timeframe), starting no closer than 128 pixel clocks from the end of the vertical blank interval.

The state transition signal ENC_EN is true when a CTL3 pulse of sufficient duration occurs during this period. The state transition signal ENC_DIS is true when a vertical blanking period has passed with no CTL3 pulse.

Enhanced Encryption Status Signaling (EESS)

EESS utilizes all four CTLx signals. Two possible CTLx patterns are used to indicate the encryption status of the current frame as described in Table 2-5.

| CTL3: | CTL2: | CTL1: | CTL0: | Description: |
|-------|-------|-------|-------|--|
| 1 | 0 | 0 | 1 | Encryption is enabled for this frame. |
| 0 | 0 | 0 | 1 | Encryption is disabled for this frame. |

Table 2-5. Enhanced Encryption Status Signaling (EESS)

This signaling is always used with the HDMI protocol, but is an optional feature with the DVI protocol. An HDCP Transmitter may determine if the HDCP Receiver supports EESS during DVI protocol by querying its *Bcaps* bit 1.1_FEATURES. If both HDCP Devices are capable of EESS, then the HDCP Transmitter must indicate that EESS will be used by setting the ENABLE_1.1_FEATURES bit of the Ainfo register prior to starting the authentication protocol.

HDMI-capable HDCP Transmitters must determine if the HDCP Receiver is DVI or HDMI-capable before authentication is attempted. An HDCP Receiver indicates that it is HDMI capable as prescribed by the HDMI specification (rather than the *Bcaps* HDMI_RESERVED bit). Transition to HDMI protocol must then be initiated by the HDCP Transmitter (or downstream side of an HDCP Repeater) by the transmission of a Data Island period. The reception of a Data Island preamble followed by a Data Island Guard Band will transition the HDCP Receiver to HDMI mode. The successful transition to HDMI mode by the HDCP Receiver is indicated by setting *Bstatus* bit HDMI_MODE. After this, the authentication protocol is started and EESS assumed regardless of the setting of *Bcaps* bit 1.1_FEATURES or Ainfo bit ENABLE_1.1_FEATURES.

If an HDMI-capable HDCP Transmitter detects an HDCP Receiver that is not HDMI-capable, then it must follow the sequence above for DVI HDCP Transmitters if it is capable of performing EESS while transmitting with the DVI protocol.

For an HDCP Repeater that supports HDMI protocol, it must be noted that the upstream link is in many ways independent from the downstream links. It is possible that HDMI protocol may be active on the downstream HDCP-protected Interfaces from an HDCP Repeater while the upstream HDCP-protected interface is receiving DVI protocol, or vice versa. If two or more downstream HDCP-protected Interface Ports are supported, it is possible for one or more of these to be transmitting DVI protocol while one or more others are transmitting HDMI protocol. Such an HDCP Repeater must accommodate all of these scenarios.

The CTLx signals described in Table 2-5 are only valid within a 16-clock window of opportunity starting at 512 pixel clocks following the active edge of VSYNC. When

authenticated and not in the HDMI AVMUTE state, the HDCP Transmitter must continually assert one of these CTLx patterns during this window of opportunity. The CTLx signals may be used for other purposes outside of this window of opportunity. See timing diagram D-1 in Appendix D .

The state transition signal ENC_EN is true at the end of this window of opportunity if the encryption enable value is transmitted during the window. The state transition signal ENC_DIS is true at the end of this window of opportunity if the encryption disabled value is transmitted during this window. Neither signal is activated otherwise or elsewhere.

The specific methods an HDCP Receiver uses to determine which of the two signals, ENC_EN or ENC_DIS, is presented, in consideration of environments where signaling errors may occur, are left to the implementation. The HDCP Receiver may also use heuristics based on common usage in its decision, such as assuming that the frame has the same encryption status as the previous frame.

The active edge of VSYNC is either a rising or falling edge. For the purposes of the signaling described above, the HDCP Transmitter and Receiver determine which edge is active by polling VSYNC at the start of the previous Video Data Period. If VSYNC is low at this point, then the active edge of VSYNC is defined as a rising edge. If VSYNC is high at this point, then the active edge of VSYNC is defined as a falling edge. Upon removal or connection of an HDCP Device, the active edge of VSYNC must default to the falling edge. In addition, following a VSYNC active edge, no subsequent VSYNC edge may be considered active until a Video Data Period occurs. This VSYNC polarity determination is not specified as used for any other purpose than to establish the position of the encryption signaling window position.

It is required that no Data Island or Video Data, nor any Guard Band, be transmitted during a keep-out period that starts 508 pixels past the active edge of VSYNC and ends 650 pixels past the active edge of VSYNC. See timing diagram D-1 in Appendix D.

3 Data Encryption

HDCP Encryption is applied at the input to the T.M.D.S. Encoder and decryption is applied at the output of the T.M.D.S. Decoder (Figure 3-1). HDCP Encryption consists of a bit-wise exclusive-or (XOR) of the HDCP Content with a pseudo-random data stream produced by the HDCP Cipher. In dual-link implementations the Audiovisual Content is 48-bits wide and requires two HDCP Ciphers to produce the required pseudo-random streams.

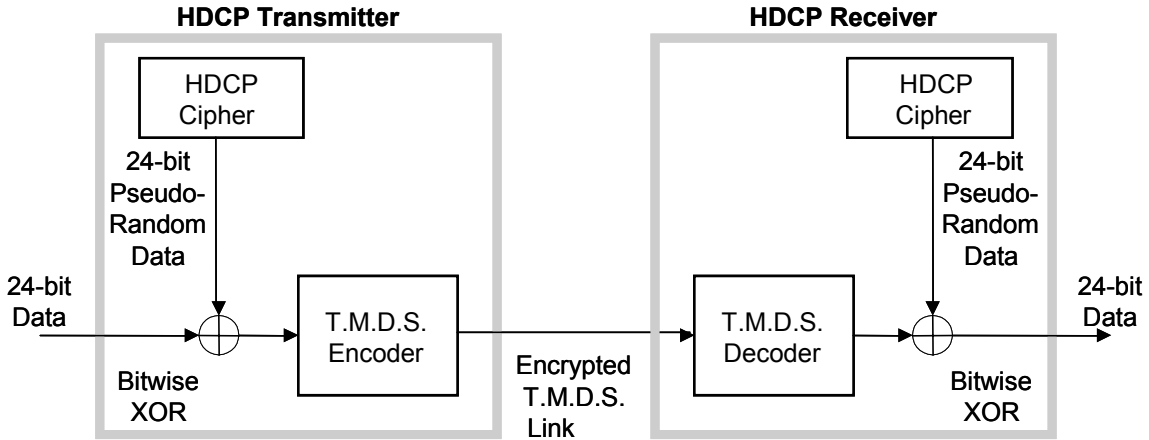


Figure 3-1. HDCP Encryption and Decryption

During the vertical-blanking interval, the `hdcpBlockCipher` function prepares the HDCP Cipher to produce the 24-bit wide key-dependent pseudo-random stream during data periods. The HDCP Cipher generates a new 24-bit word of pseudo-random data for every pixel of HDCP Content to be encrypted. The 24-bits of cipher output are applied to the T.M.D.S. channels as shown in Table 3-1. As an example, the RGB video stream data is also shown in Table 3-1.

| Cipher Output | T.M.D.S. Channel | Video Stream Bits |
|---------------|------------------|-------------------|
| 23:16 | 2 | Red[7:0] |
| 15:8 | 1 | Green[7:0] |
| 7:0 | 0 | Blue[7:0] |

Table 3-1. Encryption Stream Mapping

When transmitting auxiliary data, four TERC4 data bits on each of the two T.M.D.S. channels 1 and 2, plus one TERC4 data bit on channel 0 are encrypted. Nine bits of the existing HDCP stream cipher XOR mask are used. Table 3-2 identifies the mappings of HDCP stream cipher output bits to be XORed with TERC4 bits.

| Cipher Output: | TERC4 Bits: |
|-----------------------|--------------------|
| 23:20 | Unused |
| 19 | Channel 2 bit 3 |
| 18 | Channel 2 bit 2 |
| 17 | Channel 2 bit 1 |
| 16 | Channel 2 bit 0 |
| 15:12 | Unused |
| 11 | Channel 1 bit 3 |
| 10 | Channel 1 bit 2 |
| 9 | Channel 1 bit 1 |
| 8 | Channel 1 bit 0 |
| 7:3 | Unused |
| 2 | Channel 0 bit 2 |
| 1:0 | Unused |

Table 3-2. Encryption Stream Mapping for TERC4 Encoding

During horizontal-blanking intervals on the interface, the HDCP Cipher is re-keyed for 56 pixel clocks as described in Section 4.5. This complicates the task of breaking the encryption from line to line.

3.1 Encryption/Decryption State Diagrams

3.1.1 OESS

Figure 3-2 illustrates the encryption functions while using OESS as they relate to HSYNC, VSYNC, Video Data, and Encryption Status Signaling (ENC_EN). This diagram is applicable to both HDCP Transmitters and HDCP Receivers.

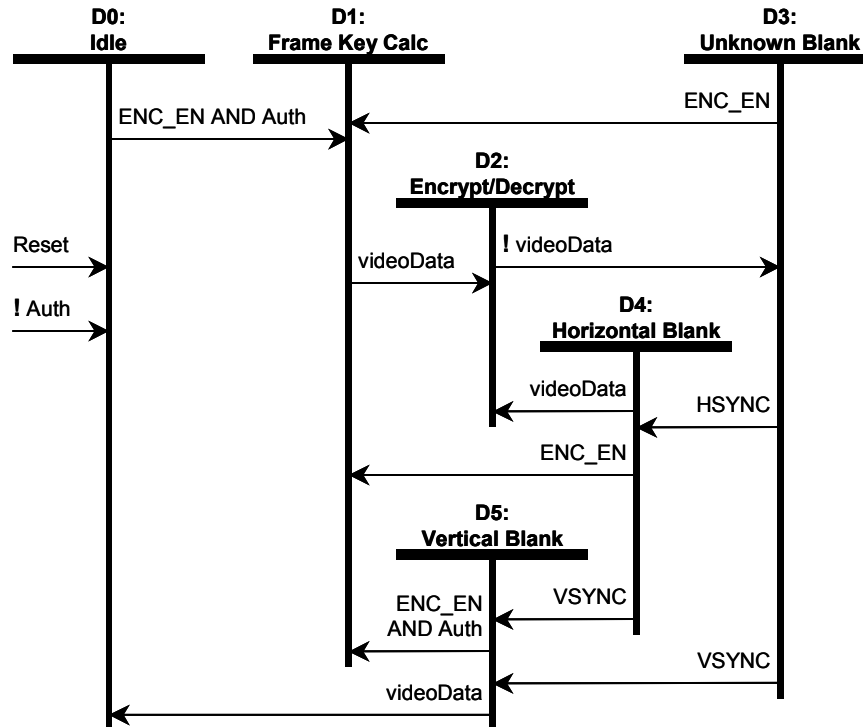


Figure 3-2. Encryption/Decryption State Diagram (OESS)

Transition Any State:D0. Reset conditions or transitions into the unauthenticated state at the HDCP Transmitter cause the encryption state machine to transition to the idle state.

State D0: Idle. The HDCP Cipher is free running and available for use as `hdcpRngCipher` (Section 4.5).

Transition D0:D1. The assertion of Encryption Enable (`ENC_EN`) when in an authenticated state (as described in Section 2.7) causes frame key calculation.

State D1: Frame Key Calculation. The frame key for the next video frame is calculated as described in section 4.5, using `hdcpBlockCipher`.

Transition D1:D2. Entering a video data period causes this transition.

State D2: Encrypt/Decrypt. HDCP Transmitters encrypt pixel data in this state, while HDCP Receivers decrypt data. Both use the `hdcpStreamCipher` as described in section 4.5.

Transition D2:D3. The end of the video data period signals the end of video pixel data.

State D3: Unknown Blank. At the end of active pixel data, it is not assumed that HDCP Devices are able to distinguish between horizontal and vertical sync. In this state, HDCP Devices must begin to rekey the HDCP cipher using `hdcpRekeyCipher` as described in section 4.5.

Transition D3:D1. The assertion of `ENC_EN` as specified for OESS (as described in Section 2.7) results in the generation of a new frame key.

Transition D3:D4. The assertion of `HSYNC` identifies the horizontal blank.

Transition D3:D5. The assertion of VSYNC identifies the vertical blank.

State D4: Horizontal Blank. The rekey operation continues if not completed during State D3.

Transition D4:D2. The assertion of videoData signals the beginning of the next line of HDCP Content to be encrypted/decrypted by the HDCP Device.

Transition D4:D1. The assertion of ENC_EN as specified for OESS (as described in Section 2.7) results in the generation of a new frame key.

Transition D4:D5. The assertion of VSYNC identifies the vertical blank.

State D5: Vertical Blank. This state waits for one of the exit conditions.

Transition D5:D1. The assertion of ENC_EN when in an authenticated state (as described in Section 2.7) results in the generation of a new frame key.

Transition D5:D0. If the Advance Cipher option is not enabled, the return to active video pixel data before the assertion of ENC_EN (as described in Section 2.7) indicates that HDCP Encryption has been disabled for this frame of active video data (i.e. indicates that ENC_DIS is true).

3.1.2 EESS

Figure 3-3 illustrates the encryption function while using EESS as they relate to HSYNC, VSYNC, Video Data, Packet Data, and Encryption Status Signaling (ENC_EN, ENC_DIS). This diagram is applicable to both HDCP Transmitters and HDCP Receivers.

HDMI transmits data during Video Data Periods and Data Island Periods. All of this data requires HDCP Encryption.

Video Data Periods are identical to DVI video data periods with the exception that they begin with a two-pixel Leading Guard Band. The state transition variable `videoData` is defined to go TRUE on the first active pixel of video data in the period (i.e. after the Guard Band) and is defined to go FALSE following the last active pixel of video data in the period. There is no Trailing Guard Band on a Video Data Period.

Data Islands begin with a two-pixel Leading Guard Band and end with a two-pixel Trailing Guard Band. Between the Guard Bands, packet data is transmitted. Unlike the 8 to 10 bit encoding used for video pixel data, each pixel of the packet data is encoded using T.M.D.S. Error Reduction Coding (TERC4) performing a 4 to 10 bit conversion of the input packet data to the 10 bits required for differential transmission on each of the three T.M.D.S. channels. The state transition variable `packetData` is defined to go TRUE for the first pixel of the Data Island containing packet data (i.e. first pixel following the Leading Guard Band) and is defined to go FALSE following the last pixel containing packet data (i.e. the first pixel of the Trailing Guard Band).

The HDMI Specification defines a facility for the HDCP Transmitter to inform the HDCP Receiver that the streams being transmitted contain no useful visual or aural information and should be muted. HDCP uses this mechanism to provide a means of temporarily disabling HDCP Encryption while remaining authenticated. During the AVMUTE state, an HDCP Transmitter is required to not assert any ENC_EN signal. Also during the AVMUTE state, the HDCP Receiver should ignore Encryption Status Signaling and operate as if ENC_DIS is asserted.

This mechanism can be applied in the case of an erratic or changing pixel clock that may result from a change from one video format to another, such as from an SDTV (27MHz) signal to an HDTV (74.25MHz) signal. If the pixel clock change were preceded by a Set_AVMUTE request and followed by a Clear_AVMUTE request, then authentication would not be affected.

The state transition signal AVMUTE is defined to be TRUE for a duration of one pixel coincident with the assertion of ENC_EN or ENC_DIS if the HDCP Device is in an AVMUTE state, as defined in the HDMI Specification.

Note that when using the DVI protocol, AVMUTE and `packetData` are always FALSE, as such features are only supported when using the HDMI protocol. Also for DVI protocol transmission, the term *Video Data Period* refers to the active DE period; the transition variable `videoData` is TRUE when DE is TRUE and FALSE when DE is FALSE; and there are no Data Islands, Data Island Periods, or Guard Bands.

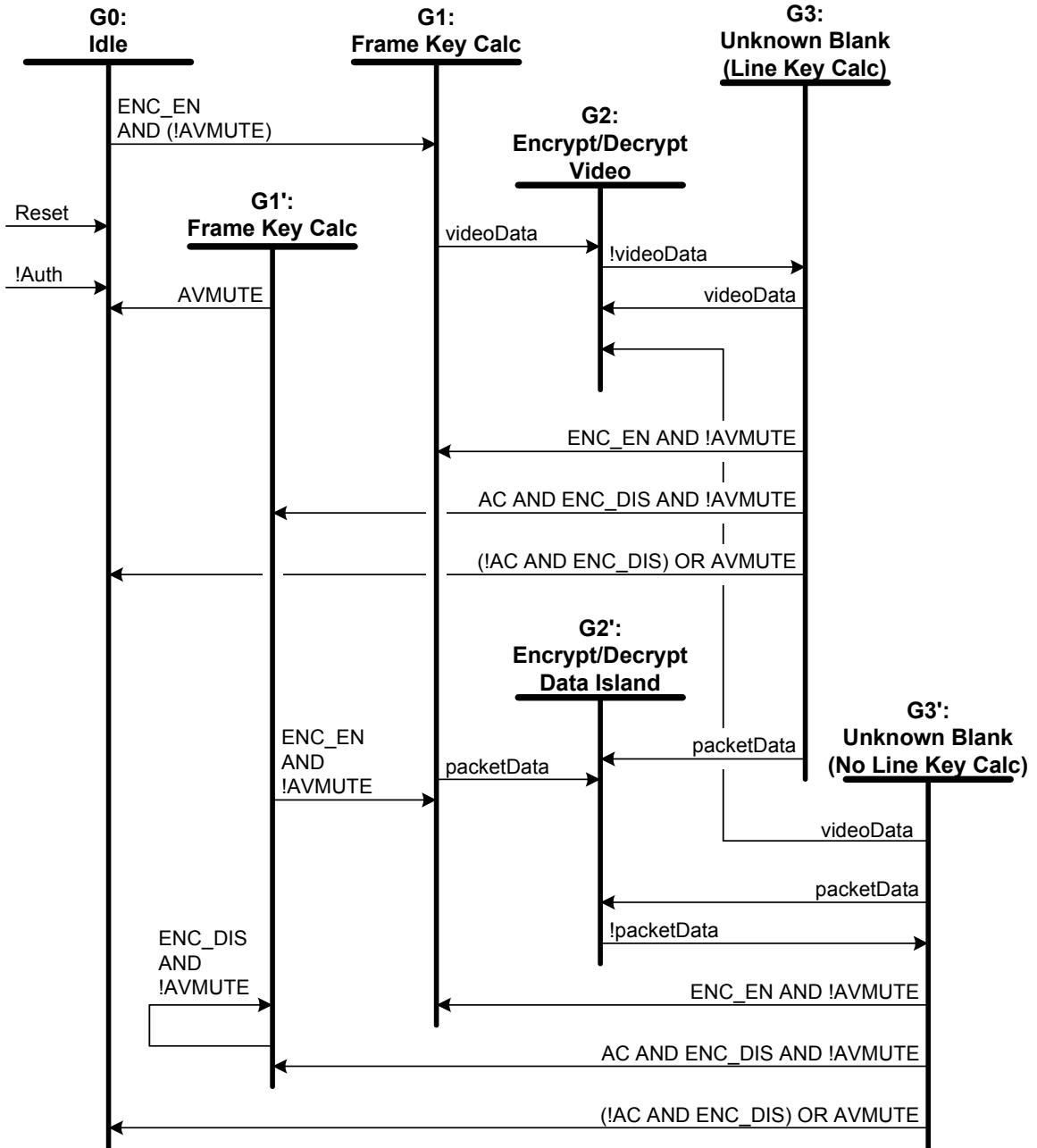


Figure 3-3. Encryption/Decryption State Diagram (EESS)

Transition Any State:G0. Reset conditions or transitions into the unauthenticated state at the HDCP Device cause the encryption state machine to transition to the idle state.

State G0: Idle. The HDCP Cipher is free running and available for use as `hdcpRngCipher`.

Transition G0:G1. The assertion/detection of Encryption Enable (`ENC_EN`) when the HDCP Device is authenticated, indicates that all of the video and auxiliary data until the next Encryption Status Signaling will be encrypted.

State G1: Frame Key Calculation. The frame key calculation using `hdcpBlockCipher` is initiated. The `hdcpBlockCipher` operation must be completed within 118 pixel clocks after the assertion of `ENC_EN`. It is required that no Data Island or Video Data, nor any Guard Band, be transmitted during a keep-out period that starts 508 pixels following `VSYNC` and ends 118 pixels past the assertion of `ENC_EN`.

State G1': Frame Key Calculation. The frame key for the next video frame is calculated as described in section 4.5, using `hdcpBlockCipher`. This state is only reached with the Advance Cipher option is enabled and an unencrypted frame is signaled.

Transition G1':G1. The assertion of Encryption Enable (`ENC_EN`) when in an authenticated state (as described in Section 2.7) causes frame key calculation.

Transition G1':G1'. The assertion of Encryption Disable (`ENC_DIS`) when in an authenticated state (as described in Section 2.7) causes frame key calculation.

Transition G1':G0. The detection of `AVMUTE` causes the encryption /decryption to enter the idle state.

Transitions G3:G1' and G3':G1'. In ADVANCE CIPHER mode the assertion of Encryption Disable (`ENC_DIS`) and not `AVMUTE` causes frame key calculation.

Transitions G1:G2, G3:G2, and G3':G2. Entering the valid data period of a video data period signals the beginning of video data encryption.

State G2: Encrypt/Decrypt Video. HDCP Devices encrypt/decrypt all 24 bits of every pixel in this state using `hdcpStreamCipher`.

Transition G2:G3. The end of video pixel data is signaled by `!videoData`.

State G3: Unknown Blank / Line Key Calc. In this state, the HDCP Device must begin to rekey the HDCP Cipher using `hdcpRekeyCipher`. The `hdcpRekeyCipher` operation must occur when `videoData` goes low and must be completed within 58 pixel clocks after `videoData` goes low, allowing the 58th pixel to be encrypted. No data period may begin until at least 58 pixel clocks following the fall of `videoData`. (See timing diagram D-2 in Appendix D)

Transitions G3:G1 and G3':G1: Frame Key Calc. The occurrence of an Encryption Enable signal (`ENC_EN`) when `AVMUTE` is false indicates that all of the data until the next Encryption Status Signaling will be encrypted.

Transitions G1:G2', G3':G2', and G3:G2'. Entering the packet data period of a Data Island period signals the beginning of auxiliary data encryption.

State G2': Encrypt/Decrypt Data Island. HDCP Devices encrypt/decrypt auxiliary data during this state using `hdcpStreamCipher`.

Transition G2':G3'. The end of Data Island packet data is signaled by `!packetData`.

State G3': Unknown Blank / No Line Key Calc. No HDCP activity occurs during this state.

Transitions G3:G0 and G3':G0. If Encryption Disable Signaling (`ENC_DIS`) occurs or when `AVMUTE` is true, then encryption has been disabled for the next frame.

4 HDCP Cipher

4.1 Overview

The HDCP Cipher is a special-purpose cipher designed for both the appropriate robustness of the authentication protocol as well as for the high-speed streaming requirement of uncompressed video data encryption.

The overall structure of the HDCP Cipher can be thought of as three layers. The first layer consists of a set of four Linear Feedback Shift Registers (LFSRs) that are combined to one bit. This one bit feeds into the middle layer when enabled via the rekey enable signal. The middle layer consists of two halves that are very similar in design. One half, *Round Function B*, performs one round of a block cipher using three 28-bit registers, B_x , B_y , and B_z . The other half, *Round Function K*, is similar in structure to Round Function B, but provides the output of latch K_y as a stream of 28-bit round keys to Round Function B at the rate of one 28-bit round key per clock pulse. The final layer takes four 28-bit register outputs from the round functions, B_y , B_z , K_y , and K_z , through a compression function to produce a 24-bit block of pseudo-random bits for every clock pulse.

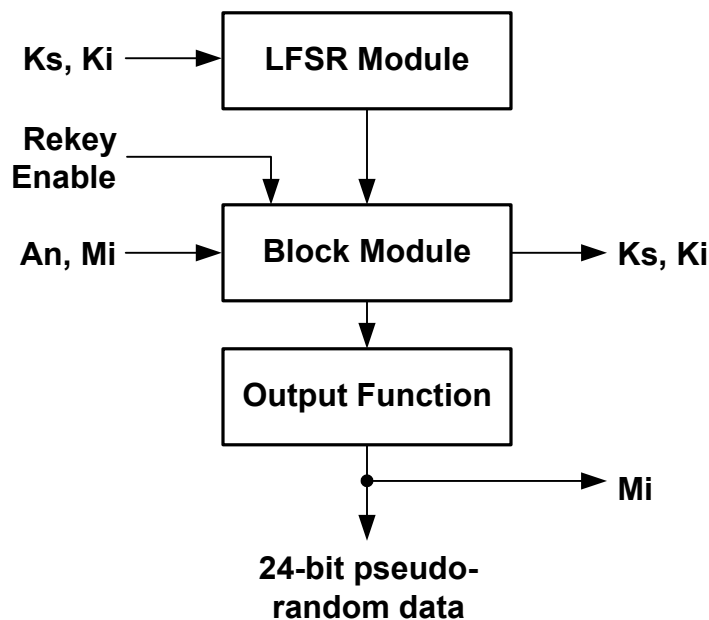


Figure 4-1. HDCP Cipher Structure

The block module operates as a block cipher during the authentication protocol. There is a single sequence, `hdcpBlockCipher`, which is used for both parts of the authentication protocol. Although decryption in block mode is possible for the HDCP cipher, it is not necessary for this application and thus is not described in this document.

The block module and the output function are used together to produce the 24-bit pseudo random sequence that is used to encrypt the HDCP Content. In this mode, `hdcpStreamCipher`, the module produces 24 bits of output for every input clock.

The LFSR module is used to re-key the block module between lines of video.

4.2 Linear Feedback Shift Register Module

The linear feedback shift register module consists of four LFSRs of different lengths and a combining function that produces a single bit stream from them. The combining function takes three taps from each LFSR. The generator polynomials and combining function taps for the LFSRs are specified in Table 4-1.

| LFSR | Polynomial | Combining Function Taps | | |
|------|--|-------------------------|----|----|
| | | 0 | 1 | 2 |
| 3 | $x^{17} + x^{15} + x^{11} + x^5 + 1$ | 5 | 11 | 16 |
| 2 | $x^{16} + x^{15} + x^{12} + x^8 + x^7 + x^5 + 1$ | 5 | 9 | 15 |
| 1 | $x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^4 + 1$ | 4 | 8 | 13 |
| 0 | $x^{13} + x^{11} + x^9 + x^5 + 1$ | 3 | 7 | 12 |

Table 4-1. LFSR Generation and Tapping

Figure 4-2 illustrates the tap locations of LFSR0 as well as the XOR term feedback into the least significant bit of LFSR0.

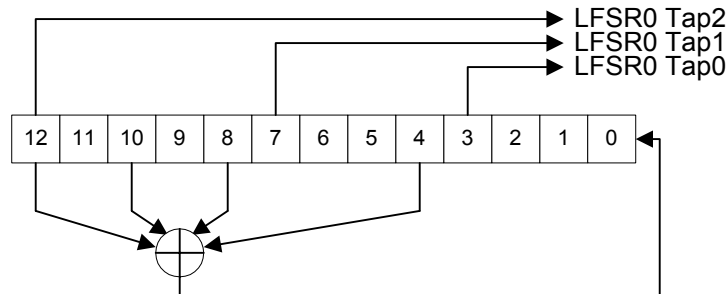


Figure 4-2. LFSR0

The combining function contains four cascaded shuffle networks, each of which includes two state bits. One tap from each of the four LFSRs is XORed together to form the data input to the first shuffle network. One tap from each of the four LFSRs is used as the select input to one of the four shuffle networks. The output of the fourth shuffle network is XORed together with one tap from each of the LFSRs. The Combiner Function illustrated in Figure 4-3.

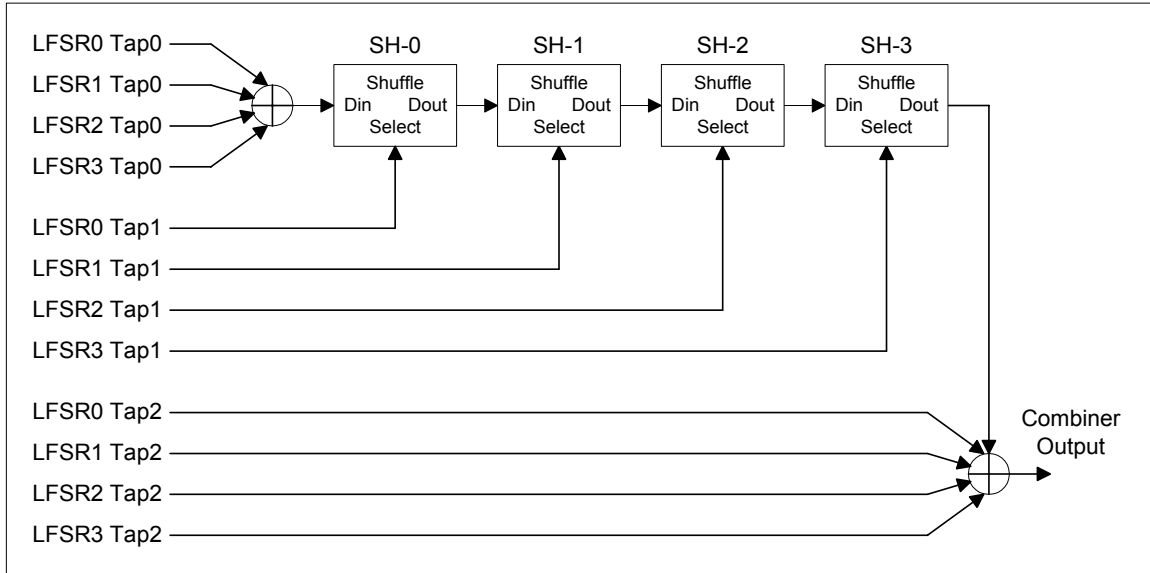


Figure 4-3. LFSR Module Combiner Function

The shuffle network is represented schematically in Figure 4-4. If the shuffle network contains the ordered pair of boolean values (A, B) and has boolean data input D and selection input S, the S value controls the next state. If S is zero, it outputs A and assumes state (B, D). If S is one, it outputs B and assumes state (D, A).

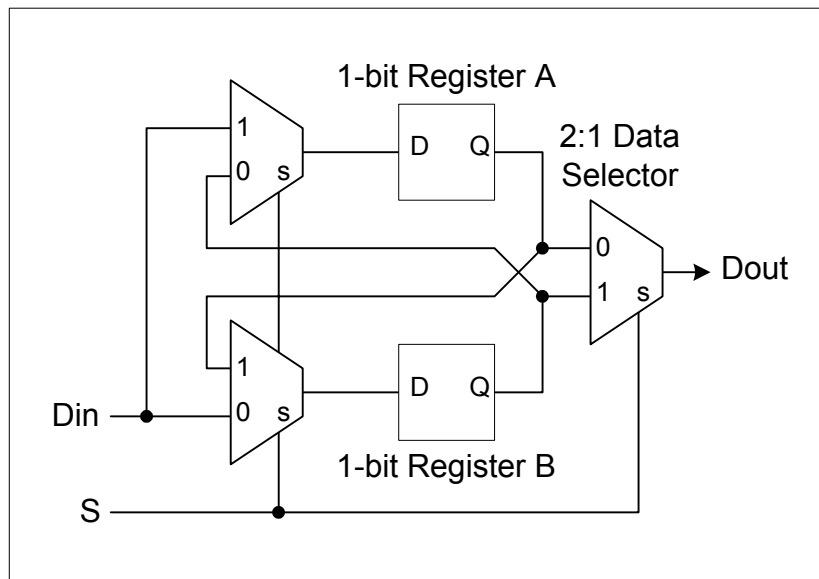


Figure 4-4. Shuffle Network

In all modes of operation the LFSRs and combining function are initialized by a 56-bit value. The 60 bits of LFSR state use these 56 bits directly plus the complements of four of the bits. The shuffle networks are each initialized with the same constant value. The initialization of the LFSR module is specified in Table 4-2 for a 56-bit initialization value.

| | Bit Field | Initial Value |
|-------------------------|------------|----------------------------|
| LFSR3 | [16] | Complement of input bit 47 |
| | [15:0] | Input bits [55:40] |
| LFSR2 | [15] | Complement of input bit 32 |
| | [14:0] | Input bits [39:25] |
| LFSR1 | [13] | Complement of input bit 18 |
| | [12:0] | Input bits [24:12] |
| LFSR0 | [12] | Complement of input bit 6 |
| | [11:0] | Input bits [11:0] |
| Shuffle Networks | Register A | 0 |
| | Register B | 1 |

Table 4-2. LFSR Module Initialization

This one-bit stream output of the combining function is the only output from the LFSR module. This bit stream provides key material to the block module when the rekey enable signal is active.

4.3 Block Module

The block module consists of two separate “round function” components. One of these components, *Round Function K*, provides a key stream for the other component, *Round Function B*. Each of these two components operates on a corresponding set of three 28-bit registers. The structure of the block module is diagrammed in Figure 4-5.

For Round Function K, bit 13 of the K_y register takes its input from the LFSR module output stream when the external rekey enable signal is asserted.

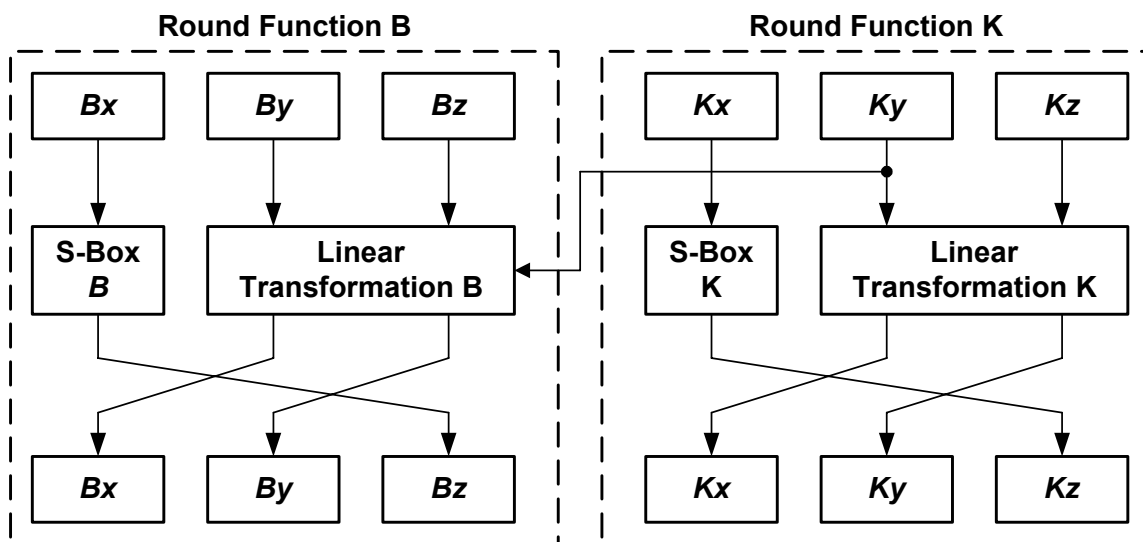


Figure 4-5. Block Module

The S-Boxes for both round functions consist of seven 4 input by 4 output S-boxes. Round function K S-Boxes are labeled SK0 through SK6 and round function B S-Boxes are labeled SB0 through SB6. The I^{th} input to box J is bit $I*7+J$ from the round x register (Bx or Kx), and output I of box J goes to bit $I*7+J$ of register z of the round function (Bz or Kz). Bit 0 is the least significant bit. The S-box permutations of round functions K and B are specified in Table 4-3.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| SK0 | 8 | 14 | 5 | 9 | 3 | 0 | 12 | 6 | 1 | 11 | 15 | 2 | 4 | 7 | 10 | 13 |
| SK1 | 1 | 6 | 4 | 15 | 8 | 3 | 11 | 5 | 10 | 0 | 9 | 12 | 7 | 13 | 14 | 2 |
| SK2 | 13 | 11 | 8 | 6 | 7 | 4 | 2 | 15 | 1 | 12 | 14 | 0 | 10 | 3 | 9 | 5 |
| SK3 | 0 | 14 | 11 | 7 | 12 | 3 | 2 | 13 | 15 | 4 | 8 | 1 | 9 | 10 | 5 | 6 |
| SK4 | 12 | 7 | 15 | 8 | 11 | 14 | 1 | 4 | 6 | 10 | 3 | 5 | 0 | 9 | 13 | 2 |
| SK5 | 1 | 12 | 7 | 2 | 8 | 3 | 4 | 14 | 11 | 5 | 0 | 15 | 13 | 6 | 10 | 9 |
| SK6 | 10 | 7 | 6 | 1 | 0 | 14 | 3 | 13 | 12 | 9 | 11 | 2 | 15 | 5 | 4 | 8 |
| SB0 | 12 | 9 | 3 | 0 | 11 | 5 | 13 | 6 | 2 | 4 | 14 | 7 | 8 | 15 | 1 | 10 |
| SB1 | 3 | 8 | 14 | 1 | 5 | 2 | 11 | 13 | 10 | 4 | 9 | 7 | 6 | 15 | 12 | 0 |
| SB2 | 7 | 4 | 1 | 10 | 11 | 13 | 14 | 3 | 12 | 15 | 6 | 0 | 2 | 8 | 9 | 5 |
| SB3 | 6 | 3 | 1 | 4 | 10 | 12 | 15 | 2 | 5 | 14 | 11 | 8 | 9 | 7 | 0 | 13 |
| SB4 | 3 | 6 | 15 | 12 | 4 | 1 | 9 | 2 | 5 | 8 | 10 | 7 | 11 | 13 | 0 | 14 |
| SB5 | 11 | 14 | 6 | 8 | 5 | 2 | 12 | 7 | 1 | 4 | 15 | 3 | 10 | 13 | 9 | 0 |
| SB6 | 1 | 11 | 7 | 4 | 2 | 5 | 12 | 9 | 13 | 6 | 8 | 15 | 14 | 0 | 3 | 10 |

Table4-3. Block Module S-Box Values

Both linear transformation K and linear transformation B produce 56 output values. These values are the combined outputs from eight diffusion networks that each produces seven outputs. The diffusion network function is specified in Table 4-4. Each diffusion network has seven data inputs labeled $I_0 - I_6$, seven outputs $O_0 - O_6$, plus an additional seven optional key inputs $K_0 - K_6$.

The diffusion networks of round function K are specified in Table 4-5. Note that none of the round function K diffusion networks have the optional key inputs. The diffusion units of round function B are specified in Table 4-6. Half of these diffusion networks have key inputs that are driven from the K_y register of round function K. A dash in the table indicates that the key input is not present.

| | Diffusion Network Logic Function | | | | | | | |
|----------------------|----------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|-------|
| O₀ | $K_0 \oplus$ | $I_1 \oplus$ | $I_2 \oplus$ | $I_3 \oplus$ | $I_4 \oplus$ | $I_5 \oplus$ | I_6 | |
| O₁ | $K_1 \oplus$ | $I_0 \oplus$ | $I_2 \oplus$ | $I_3 \oplus$ | $I_4 \oplus$ | $I_5 \oplus$ | I_6 | |
| O₂ | $K_2 \oplus$ | $I_0 \oplus$ | $I_1 \oplus$ | $I_3 \oplus$ | $I_4 \oplus$ | $I_5 \oplus$ | I_6 | |
| O₃ | $K_3 \oplus$ | $I_0 \oplus$ | $I_1 \oplus$ | $I_2 \oplus$ | $I_4 \oplus$ | $I_5 \oplus$ | I_6 | |
| O₄ | $K_4 \oplus$ | $I_0 \oplus$ | $I_1 \oplus$ | $I_2 \oplus$ | $I_3 \oplus$ | $I_5 \oplus$ | I_6 | |
| O₅ | $K_5 \oplus$ | $I_0 \oplus$ | $I_1 \oplus$ | $I_2 \oplus$ | $I_3 \oplus$ | $I_4 \oplus$ | I_6 | |
| O₆ | $K_6 \oplus$ | $I_0 \oplus$ | $I_1 \oplus$ | $I_2 \oplus$ | $I_3 \oplus$ | $I_4 \oplus$ | $I_5 \oplus$ | I_6 |

Table 4-4. Diffusion Network Logic Function

| | K1 | K2 | K3 | K4 | K5 | K6 | K7 | K8 |
|----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| I₀ | Kz0 | Kz7 | Kz10 | Kz13 | Kz16 | Ky16 | Ky20 | Ky24 |
| I₁ | Kz1 | Kz8 | Kz11 | Kz14 | Kz17 | Ky17 | Ky21 | Ky25 |
| I₂ | Kz2 | Kz9 | Kz12 | Kz15 | Kz18 | Ky18 | Ky22 | Ky26 |
| I₃ | Kz3 | Ky0 | Ky3 | Ky6 | Ky9 | Ky19 | Ky23 | Ky27 |
| I₄ | Kz4 | Ky1 | Ky4 | Ky7 | Ky10 | Kz19 | Kz22 | Kz25 |
| I₅ | Kz5 | Ky2 | Ky5 | Ky8 | Ky11 | Kz20 | Kz23 | Kz26 |
| I₆ | Kz6 | Ky12 | Ky13 | Ky14 | Ky15 | Kz21 | Kz24 | Kz27 |
| O₀ | Kx0 | Ky0 | Ky1 | Ky2 | Ky3 | Kx1 | Kx2 | Kx3 |
| O₁ | Kx4 | Ky4 | Ky5 | Ky6 | Ky7 | Kx5 | Kx6 | Kx7 |
| O₂ | Kx8 | Ky8 | Ky9 | Ky10 | Ky11 | Kx9 | Kx10 | Kx11 |
| O₃ | Kx12 | Ky12 | Ky13 | Ky14 | Ky15 | Kx13 | Kx14 | Kx15 |
| O₄ | Kx16 | Ky16 | Ky17 | Ky18 | Ky19 | Kx17 | Kx18 | Kx19 |
| O₅ | Kx20 | Ky20 | Ky21 | Ky22 | Ky23 | Kx21 | Kx22 | Kx23 |
| O₆ | Kx24 | Ky24 | Ky25 | Ky26 | Ky27 | Kx25 | Kx26 | Kx27 |

Table 4-5. K Round Input and Output Mapping

| | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 |
|----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| I₀ | Bz0 | Bz7 | Bz10 | Bz13 | Bz16 | By16 | By20 | By24 |
| I₁ | Bz1 | Bz8 | Bz11 | Bz14 | Bz17 | By17 | By21 | By25 |
| I₂ | Bz2 | Bz9 | Bz12 | Bz15 | Bz18 | By18 | By22 | By26 |
| I₃ | Bz3 | By0 | By3 | By6 | By9 | By19 | By23 | By27 |
| I₄ | Bz4 | By1 | By4 | By7 | By10 | Bz19 | Bz22 | Bz25 |
| I₅ | Bz5 | By2 | By5 | By8 | By11 | Bz20 | Bz23 | Bz26 |
| I₆ | Bz6 | By12 | By13 | By14 | By15 | Bz21 | Bz24 | Bz27 |
| K₀ | Ky0 | – | – | – | – | Ky7 | Ky14 | Ky21 |
| K₁ | Ky1 | – | – | – | – | Ky8 | Ky15 | Ky22 |
| K₂ | Ky2 | – | – | – | – | Ky9 | Ky16 | Ky23 |
| K₃ | Ky3 | – | – | – | – | Ky10 | Ky17 | Ky24 |
| K₄ | Ky4 | – | – | – | – | Ky11 | Ky18 | Ky25 |
| K₅ | Ky5 | – | – | – | – | Ky12 | Ky19 | Ky26 |
| K₆ | Ky6 | – | – | – | – | Ky13 | Ky20 | Ky27 |
| O₀ | Bx0 | By0 | By1 | By2 | By3 | Bx1 | Bx2 | Bx3 |
| O₁ | Bx4 | By4 | By5 | By6 | By7 | Bx5 | Bx6 | Bx7 |
| O₂ | Bx8 | By8 | By9 | By10 | By11 | Bx9 | Bx10 | Bx11 |
| O₃ | Bx12 | By12 | By13 | By14 | By15 | Bx13 | Bx14 | Bx15 |
| O₄ | Bx16 | By16 | By17 | By18 | By19 | Bx17 | Bx18 | Bx19 |
| O₅ | Bx20 | By20 | By21 | By22 | By23 | Bx21 | Bx22 | Bx23 |
| O₆ | Bx24 | By24 | By25 | By26 | By27 | Bx25 | Bx26 | Bx27 |

Table 4-6. B Round Input and Output Mapping

4.4 Output Function

The Ky, Kz, By, and Bz registers drive the final output function. Each of the 24 outputs consists of the XOR of nine terms given by the following formula:

$$(B0 \bullet K0) \oplus (B1 \bullet K1) \oplus (B2 \bullet K2) \oplus (B3 \bullet K3) \oplus (B4 \bullet K4) \oplus (B5 \bullet K5) \oplus (B6 \bullet K6) \oplus B7 \oplus K7$$

Where “ \oplus ” represents a logical XOR function and “ \bullet ” represents a logical AND function. Table 4–7 specifies the input values B and K to the 24 logic functions.

For example, output bit 0 is computed as

$$(Bz17 \bullet Kz3) \oplus (Bz26 \bullet Kz6) \oplus (Bz22 \bullet Kz0) \oplus (Bz27 \bullet Kz9) \oplus (Bz21 \bullet Kz4) \oplus (Bz18 \bullet Kz22) \oplus (Bz2 \bullet Kz5) \oplus By5 \oplus Ky10.$$

| Input | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | K0 | K1 | K2 | K3 | K4 | K5 | K6 | K7 |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Origin | Bz | Bz | Bz | Bz | Bz | Bz | Bz | By | Kz | Kz | Kz | Kz | Kz | Kz | Kz | Ky |
| Output bit | | | | | | | | | | | | | | | | |
| 0 | 17 | 26 | 22 | 27 | 21 | 18 | 2 | 5 | 3 | 6 | 0 | 9 | 4 | 22 | 5 | 10 |
| 1 | 5 | 20 | 15 | 24 | 2 | 25 | 0 | 16 | 20 | 18 | 7 | 23 | 15 | 5 | 3 | 25 |
| 2 | 22 | 5 | 14 | 16 | 25 | 17 | 20 | 11 | 7 | 19 | 2 | 10 | 22 | 4 | 13 | 21 |
| 3 | 19 | 3 | 15 | 11 | 21 | 16 | 27 | 1 | 6 | 14 | 9 | 8 | 17 | 18 | 12 | 24 |
| 4 | 19 | 6 | 17 | 18 | 22 | 7 | 9 | 12 | 25 | 6 | 5 | 2 | 10 | 15 | 21 | 8 |
| 5 | 3 | 7 | 4 | 8 | 16 | 6 | 5 | 17 | 27 | 14 | 2 | 4 | 24 | 19 | 1 | 12 |
| 6 | 8 | 21 | 27 | 2 | 11 | 24 | 12 | 3 | 17 | 26 | 4 | 16 | 27 | 7 | 22 | 11 |
| 7 | 9 | 5 | 7 | 4 | 8 | 13 | 3 | 15 | 9 | 10 | 19 | 11 | 7 | 6 | 8 | 23 |
| 8 | 26 | 13 | 23 | 10 | 11 | 7 | 15 | 19 | 13 | 12 | 18 | 24 | 15 | 23 | 7 | 16 |
| 9 | 1 | 0 | 19 | 11 | 13 | 16 | 24 | 18 | 0 | 5 | 20 | 25 | 1 | 24 | 9 | 27 |
| 10 | 26 | 13 | 9 | 14 | 10 | 4 | 1 | 2 | 14 | 23 | 27 | 25 | 17 | 19 | 1 | 22 |
| 11 | 21 | 15 | 5 | 3 | 13 | 25 | 16 | 27 | 6 | 21 | 17 | 15 | 26 | 11 | 16 | 7 |
| 12 | 20 | 7 | 18 | 12 | 17 | 1 | 16 | 0 | 11 | 22 | 20 | 0 | 26 | 23 | 17 | 2 |
| 13 | 14 | 23 | 1 | 12 | 24 | 6 | 18 | 9 | 8 | 4 | 3 | 14 | 20 | 26 | 23 | 15 |
| 14 | 19 | 6 | 21 | 25 | 23 | 1 | 10 | 8 | 19 | 0 | 18 | 2 | 13 | 8 | 24 | 14 |
| 15 | 3 | 0 | 27 | 23 | 19 | 8 | 4 | 7 | 16 | 21 | 24 | 25 | 12 | 27 | 15 | 18 |
| 16 | 6 | 5 | 14 | 22 | 24 | 18 | 2 | 21 | 3 | 5 | 8 | 25 | 7 | 27 | 2 | 26 |
| 17 | 3 | 4 | 2 | 6 | 22 | 14 | 12 | 26 | 11 | 14 | 23 | 17 | 22 | 13 | 19 | 4 |
| 18 | 25 | 21 | 19 | 9 | 10 | 15 | 13 | 22 | 1 | 16 | 14 | 11 | 12 | 6 | 10 | 19 |
| 19 | 23 | 11 | 10 | 20 | 1 | 12 | 14 | 4 | 21 | 1 | 10 | 20 | 18 | 26 | 9 | 13 |
| 20 | 11 | 26 | 20 | 17 | 8 | 23 | 0 | 24 | 20 | 21 | 9 | 25 | 12 | 3 | 15 | 0 |
| 21 | 9 | 17 | 26 | 4 | 27 | 0 | 15 | 6 | 18 | 12 | 21 | 27 | 1 | 16 | 24 | 20 |
| 22 | 22 | 12 | 2 | 10 | 7 | 20 | 25 | 13 | 13 | 0 | 3 | 16 | 22 | 11 | 26 | 9 |
| 23 | 27 | 24 | 26 | 8 | 0 | 9 | 18 | 23 | 2 | 0 | 13 | 5 | 4 | 8 | 10 | 3 |

Table 4–7. Output Function Input and Output Mapping

4.5 Operation

The HDCP Cipher is used in four different ways during operation: `hdcpBlockCipher`, `hdcpStreamCipher`, `hdcpRekeyCipher`, and `hdcpRngCipher`. No change in HDCP Cipher state occurs that is not explicitly identified in the following descriptions.

hdcpBlockCipher

This sequence is used during the first part of authentication to establish the session key, K_s , and during the vertical blanking interval preceding encrypted frames to establish the frame key, K_i . Table 4–8 and Table 4–9 describes this sequence. The initial value for the B round register is specified with the concatenation operator “||”. 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.

| Step | Activity |
|------|--|
| 1 | Load B and K registers of the block module |
| 2 | Apply 48 clocks to the block module registers |
| 3 | Save the least significant 56 bits of the B register for future use as K_s/K_i |
| 4 | Transfer 84-bit B register values to the K registers |
| 5 | Reload B registers |
| 6 | Initialize the LFSR module |
| 7 | Assert rekey enable |
| 8 | Apply 56 clocks to the LFSR and block modules, saving the 64-bit M_i value during the last four clocks as specified in Table 4–11. |
| 9 | De-assert rekey enable |

Table 4–8. hdcpBlockCipher Sequence

| | Steps | clocks | LFSR init (56 bits) | K init | B init (65 bits) | B output (84 bits) | Output Function |
|--------------------------------------|-------|--------|------------------------|-----------------|-----------------------|-----------------------|--------------------|
| hdcpBlockCipher at Authentication | 1-3 | 48 | – | K_m (56 bits) | REPEATER A_n | K_s | – |
| | 6-9 | 56 | K_s | K_s (84 bits) | REPEATER A_n | – | R_0, M_0 |
| hdcpBlockCipher at Vertical Blank | 1-3 | 48 | – | K_s (56 bits) | REPEATER M_{i-1} | K_i | – |
| | 6-9 | 56 | K_i | K_i (84 bits) | REPEATER M_{i-1} | – | R_i, M_i |

Table 4–9. hdcpBlockCipher Initial Values and Outputs

For both the B and K round functions, the x, y, and z registers may be viewed as comprising a single register 84 bits in length, identified by B[83:0] and K[83:0]. The mapping of the x, y, and z registers into the full round register is specified by Table 4–10.

| Round Register | B[83:56] | B[55:28] | B[27:0] | K[83:56] | K[55:28] | K[27:0] |
|----------------|----------|----------|----------|----------|----------|----------|
| Sub Register | Bz[27:0] | By[27:0] | Bx[27:0] | Kz[27:0] | Ky[27:0] | Kx[27:0] |

Table 4–10. Round Register Bit Precedence

When fewer than 84 bits of output of a round register are required, the least significant bits are used. When fewer than 84 bits are available for initialization, the least significant bits are

filled and the most significant bits are set to zero. For example, the 65-bit concatenation of REPEATER with A_n will be loaded into the Bx and By registers, plus the least significant nine bits of the Bz register, and the most significant 19 bits of the Bz register are set to zero. Similarly, the 56 bits from the Bx and By registers are saved as K_s or K_i during hdcpBlockCipher.

The origin of the M_i and R_i bits from the output function is specified by Table 4–11.

| Warm-up Clock (Step 8) | Output Function Bits 23.....16 | Output Function Bits 15 0 |
|---------------------------|-----------------------------------|------------------------------------|
| 53 | – | M_i [63:48] |
| 54 | – | M_i [47:32] |
| 55 | r_i [15:8] | M_i [31:16] |
| 56 | r_i [7:0] | M_i [15:0] |

Table 4–11. hdcpBlockCipher Output Function Bit Map

hdcpStreamCipher

For every video data (as defined by the T.M.D.S data enable (DE) signal) and packet data (including info-frames) to be encrypted, hdcpStreamCipher produces 24-bits of output data. Both the LFSR and block modules are clocked. The rekey enable signal is de-asserted.

hdcpRekeyCipher

During horizontal blanking intervals that immediately follow active lines of pixel data, hdcpRekeyCipher moves new key material from the LFSR module into the Block module. No other initialization of the cipher state is made, and no outputs are taken from the cipher during re-keying. Both the LFSR and block modules are clocked 56 times. The rekey enable signal is asserted.

hdcpRngCipher

The HDCP Cipher must be used as defined in Figure 4-6 to produce the value An required for the authentication protocol. This state diagram references video transmitter states from Figure 2-6.

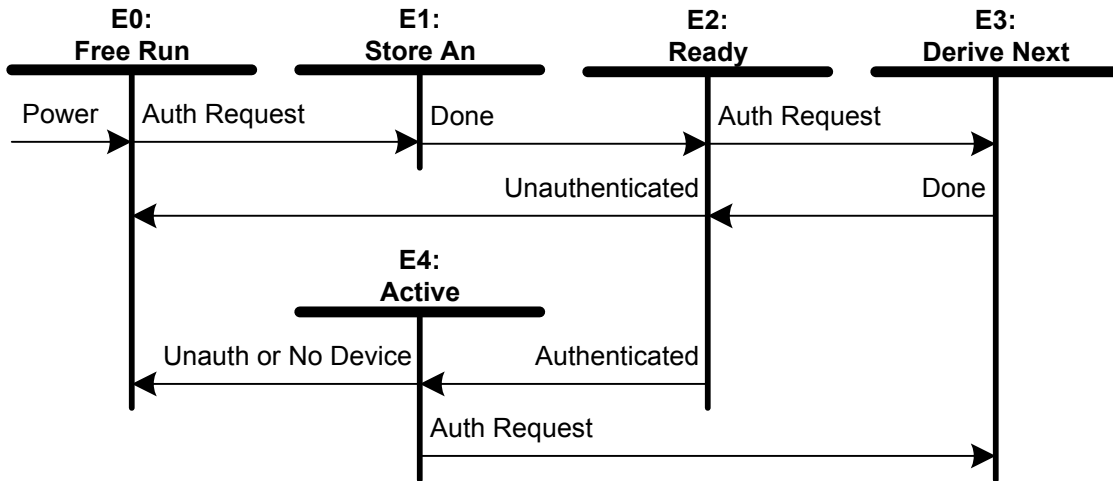


Figure 4-6. hdcpRngCipher State Diagram

Transition Any State:E0. On power up the HDCP Cipher is allowed to free run from its initial state, clocked by the pixel clock.

State E0: Free Run. The HDCP Cipher is clocked, from its current state, using the pixel clock.

Transition E0:E1. An authentication request to the HDCP Transmitter causes this transition. Authentication requests are identified by an HDCP Transmitter state transition to State:A1.

State E1: Store An. An is taken from the HDCP Cipher output function bits that are ordinarily used to produce Mi . This requires four pixel clocks.

Transition E1:E2. This transition is made immediately upon storage of An .

State E2: Ready. The An value is available for the authentication protocol.

Transition E2:E0. This transition is made if the current authentication fails, as indicated by an HDCP Transmitter state transition to State:A0.

Transition E2:E3. A new authentication request causes a new An value to be derived.

Transition E2:E4. The authentication protocol using the derived An is successful, as indicated by an HDCP Transmitter state transition to State:A4.

State E3: Derive Next. A new An is derived using the hdcpBlockCipher sequence, using the current values stored in the Mi and Ki registers.

Transition E3:E2. This transition is made immediately upon storage of An .

State E4: Active. The HDCP Transmitter is authenticated with a HDCP Receiver.

Transition E4:E0. This transition is made whenever the HDCP Transmitter becomes unauthenticated or if the HDCP Receiver is detached or goes inactive.

Transition E4:E3. An authentication request to the HDCP Transmitter causes this transition.

This pseudorandom number generator must implement a method to receive bits of outside influence. This method must mix the input influence bits into the values of the block register without replacement. That is, there must be no way to determine the value--only change it from whatever it is to another value. For example, one can exclusive-or the influence values into the state. However, any 1-to-1 operation that does not reduce the number of possible values or skew the otherwise uniform probability distribution of possible values is acceptable.

The bits of influence shall come from a source of reasonable variability or entropy. A reasonable level of variability or entropy is established if, given 1,000,000 different power up cycles on the HDCP transmitter logic such that the amount of time from power up to the initial authentication were controlled precisely enough to eliminate any variability from the free running of the cipher before initial authentication (i.e. the number of pixel clocks applied to the cipher in State E0 remains unchanged between different tests), and the A_n values from the first authentication attempt after the additional influence has been applied (using different content streams if this influence comes from the content stream), the probability of there being any duplicates in this list of 1,000,000 A_n values collected is less than 50%. This corresponds to about 40 (considering one million is about 2^{20}) random bits out of the 64 (or equivalent if the bits are biased).

An (incomplete) list of sources of entropy might include:

- a) a true Random Number Generator or analog noise source, even if a poor (biased) one
- b) a pseudo-random number generator (PRNG) where the state is stored in non-volatile memory after each use. (That is, every power on continues the sequence--it does produce not the same sequence each time). Flash memory or even disk is usable for this purpose as long as it is reasonably secure from tampering. The `hdcpRngCipher` combined with tamper-resistant non-volatile memory is one such solution.
- c) timers, network statistics, error correction information, radio/cable television signals, disk seek times, etc.
- d) Since the random number A_n is not used for secret material, a reliable (not manipulatable by the user) calendar and time-of-day clock can be used as seed. For example, some broadcast content sources may give reliable date and time information.

Different product environments have different resources available to them. There is generally no one source that is available in all environments.

The initial state of the `hdcpRngCipher` is not defined and is left to the implementer. Ideally, one would prefer that the initial value be different for each device power-on, though this is not possible in many environments. In addition, the Rekey enable signal may but need not be enabled during `hdcpRngCipher` operation.

The A_n values do not have to be secret, but must be fresh. That is, the method of producing new values must have integrity.

While each A_n value is already required to be fresh, dual-link transmitters or transmitters with multiple downstream links must ensure that each downstream link receives a distinct A_n value for each link. This ensures that each link between HDCP devices that have multiple inputs or outputs sharing the same device keys will produce distinct session keys (K_s), encryption keystreams, and authentication values (M_i , R_i , P_j).

5 Renewability

It is contemplated that an authorized participant in the authentication protocol may become compromised so as to expose the Device Private Keys it possesses for misuse by unauthorized parties. In consideration of this, each HDCP Receiver is issued a unique set of Device Private Keys, matched with a non-secret identifier (the KSV), referred collectively as the Device Key Set. Through a process defined in the HDCP Adopter's License, the Digital Content Protection LLC may determine that a set of Device Private Keys has been compromised. If so, it places the corresponding KSV on a revocation list that the HDCP Transmitter checks during authentication. Other authorized HDCP receivers are not affected by this revocation because they have different sets of Device Private keys.

The HDCP Transmitter is required to manage system renewability messages (SRMs) carrying the KSV 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.

The SRMs are delivered with content and must be checked when available. Additionally, devices compliant with HDCP 1.2 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 1.2 or higher updates the SRM stored in its non-volatile storage when presented with a newer SRM version is explained in Section 5.2.

5.1 SRM Size and Scalability

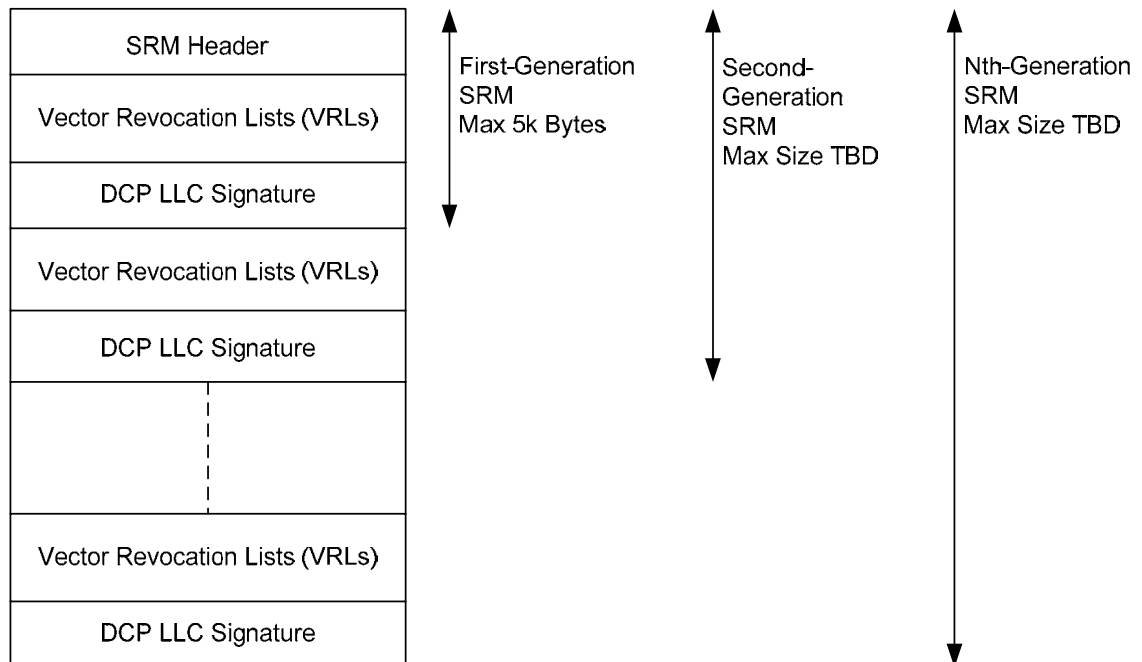


Figure 5-1. SRM Generational Format

As illustrated in the figure above, 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 KSVs. 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 SRM. All values are stored in big endian format.

| Name | Size (bits) | Function |
|-----------------------|--|--|
| SRM ID | 4 | A value of 0x8 signifies that the message is for HDCP. All other values are reserved |
| Reserved | 12 | Reserved for future definition. Must be 0x000 |
| SRM Version | 16 | Sequentially increasing unique SRM numbers. Higher numbered SRMs are more recent |
| SRM Generation Number | 8 | Indicates the generation of the SRM. The generation number starts at 1 and increases sequentially |
| VRL Length | 24 | Specifies the combined length of all vector revocation lists contained in this SRM. The length is in bytes and includes the three bytes of this field, the combined size of the vector revocation lists (only those contained in the first-generation SRM), and the 40 bytes of the Digital Content Protection LLC signature in the first-generation SRM |
| VRLs | Variable. Max 40544 (5068 bytes) | One or more VRLs, each in the format specified in the HDCP VRL format table below |
| DCP LLC Signature | 320 | A cryptographic signature of the SRM as defined by the Digital Signature Algorithm (DSA), as described in FIPS Publication 186-1 dated December 15, 1998. The first 160 bits is the big endian representation of the "r" value of the signature and the trailing 160 bits is the big endian representation of the "s" value produced by DSA |

Table 5–1. System Renewability Message Format

The SRM contains the vector revocation list, variable-length list of KSVs that belong to compromised devices. The format of the revocation list is specified in Table 5–2.

| Name | Size (bits) | Function |
|-------------------|-------------|---|
| Reserved | 1 | Set to 0. |
| Number of Devices | 7 | Specifies the number KSVs N in this list. |
| Device KSVs | 40 * N | Forty-bit KSVs follow the type/number byte. The first byte following the type byte is the most significant byte of the first KSV in the list. |

Table 5–2. Vector Revocation List Format

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

| Name | Size (bits) | Function |
|------------|-------------|--|
| VRL Length | 16 | Specifies the combined length in bytes of all VRLs in this generation extension, the 2 bytes of this field and the 40 bytes of the DCP LLC signature |
| VRLs | Variable | One or more VRLs, each in the format specified in the HDCP VRL format table below |
| DCP LLC | 320 | A cryptographic signature of the SRM as defined by the Digital |

| | | |
|-----------|--|--|
| Signature | | Signature Algorithm (DSA), as described in FIPS Publication 186-1 dated December 15, 1998. The first 160 bits is the big endian representation of the "r" value of the signature and the trailing 160 bits is the big endian representation of the "s" value produced by DSA. The signature field is calculated over all preceding fields of the SRM |
|-----------|--|--|

Table 5-3. Next-generation extension format

Table 5-4 gives the cryptographic parameters used to verify the digital signature of the SRM.

| Parameter | Value (hexadecimal) |
|---------------|--|
| Prime Modulus | d3c3f5b2fd1761b7018d75f79343786b17395b355a52c7b8a1a24fc36a7058ff8e7fa164f500e0dca0d284821d969e4b4f34dc0cae7c7667b844c747d4c6b983e52ba70e5447cf35f404a0bcd1974c3a10715509b3721530a73f3207b99820495c7b9c143275733b028a49fd968919542a39951c46edc2118c59802bf3287527 |
| Prime Divisor | ee8af2ce5e6db56acd6d14e297ef3f4df9c708e7 |
| Generator | 92f85d1b6a4d52131ae43e2445de1ab502afdeaca9bed7315d56d766cd2786118f5db14abdeca9d25162977da83effa88eedc6bfeb37e1a90e29cd0ca03d799e92dd2945f778585ff7c835642c21ba7fb1a0b6be81c8a5e3c8ab69b21da54242c98e9b8aab4a9dc251fa7dac29216fe8b93f185b2f67405b69462442c2ba0bd9 |
| Public Key | c70600526ba0b0863a80fbe0a3acff0d4f0d76658a1754a8e7654755f15ba78d56950e48654f0bbde16804de1b541874db22e14f031704db8d5cb2a417c4566c27ba973c43d84e0da2a70856fe9ea48d87259038b16553e662435ff7fd5206e27bb7ffbd886c241095c8dc8d66f662cbd88f9df7e9b3fb8362a9f7fa36e53799 |

Table 5-4. Cryptographic Parameters for Verifying SRM

See Table A-23 for a sample SRM that is signed with the production cryptographic parameters.

5.2 Updating SRMs

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

1. Verify that the version number of the new SRM is greater than the version number of the SRM currently stored in the device's non-volatile storage
2. If the version number of the new SRM is greater (implying that it is a more recent version), verify the signature on the new SRM
3. On successful signature verification, replace the current SRM in the device's non-volatile storage with the new SRM. If, for instance, the device supports only second-generation SRMs and the new SRM is a third-generation SRM, the device is not required to store the third-generation extension. Devices compliant with HDCP 1.2 or higher must be capable of storing at least 5kB bytes of the SRM (First-Generation SRM)

Appendix A. Test Vectors

Table A–1 gives facsimile key information for test purposes.

| | Transmitter A1 | Transmitter A2 | Receiver B1 | Receiver B2 |
|-----------------------------|-----------------------|-----------------------|--------------------|--------------------|
| Key Selection Vector | b70361f714 | 43f72d5066 | 511ef21acd | e72697f401 |
| Key 0 | 4da4588f131e69 | 9aaba1f9ef907c | bc13e0c75bf0fd | 93afe1ff4ca0ed |
| Key 1 | 1f823558e65009 | 34a0407731d1d0 | ae0d2c7f76443b | efb49d4a25a4e4 |
| Key 2 | 8a6a47abb9980d | 97c682992dc5d9 | 24bf2185a36c60 | e822d8a9335346 |
| Key 3 | f3181b52cbc5ca | da80caca68ed15 | f4bc6cbcd7a32f | 8812c3004e23d2 |
| Key 4 | fb147f6896d8b4 | 1866d9b51462a6 | a72e69c5eb6388 | dc63ba78d94263 |
| Key 5 | e08bc978488f81 | d9fc9599bb7498 | 7fa2d27a37d9f8 | 47ebdf52776fd5 |
| Key 6 | a0d064c8112c41 | 7a062ac883f528 | 32fd3529dea3d1 | 4bce49472e0464 |
| Key 7 | b39d5a28242044 | f5938c662af454 | 485fc240cc9bae | 0479bed7732682 |
| Key 8 | b928b2bdad566b | ec3075e82d3ef2 | 3b9857797d5103 | c5f800fad716d5 |
| Key 9 | 91a47b4a6ce4f6 | 536e376e7ffc49 | 0dd170be615250 | f53fd67ba9b9ec |
| Key 10 | 5600f8205e9d58 | 51c83a6cbeb116 | 1a748be4866bb1 | 6fb3901e5867f2 |
| Key 11 | 8c7fb706ee3fa0 | 79d44ae1bd5f50 | f9606a7c348cca | 24c46f520f1be5 |
| Key 12 | c02d8c9d7cbc28 | 674b2563e27393 | 4bbb037899eea1 | 2038176d369ed7 |
| Key 13 | 561261e54b9f05 | 7a1357efc538a2 | 190ecf9cc095a9 | 9ba9cd6a077a57 |
| Key 14 | 74f0de8ccac1cb | 6486e57ea46b02 | a821c46897447f | 5f2764b35c5591 |
| Key 15 | 3bb8f60efcdb6a | ddf27a1ce8a299 | 1a8a0bc4298a41 | ee32f1171f5356 |
| Key 16 | a02bbb16b22fd7 | dc8bd1fa5b46b9 | aefc0853e62082 | d20a9e2f4d57fa |
| Key 17 | 482f8e46785498 | 27ef71efef9b73 | f75d4a0c497ba4 | 439eb96d2daff0 |
| Key 18 | 66ae2562274738 | 187599f603c947 | ad6495fc8a06d8 | 1c68df6f868aaf |
| Key 19 | 3d4952a323ddf2 | 023ae9da303ecb | 67c2020c2b2e02 | dd50d7551dc6fb |
| Key 20 | e2d231767b3a54 | 3d1cf6533dea8e | 8f116b18f4ae8d | 50b85379165c5f |
| Key 21 | 4d581aede66125 | 34dd5525f1890c | e3053fa3e9fa69 | f45d64b097d6b5 |
| Key 22 | 326082bf7b22f7 | 367dd774a07f4c | 37d8002881c7d1 | a1a154e07adb4d |
| Key 23 | f61b463530ce6b | cdc34c8a6f56d1 | c3a5fd1c15669c | 0755ea83e47e71 |
| Key 24 | 360409f0d7976b | de3413927363a8 | 9e93d41e0811f7 | e1dca26293efe4 |
| Key 25 | a1e105618d49f9 | 21b11c739f45b3 | 2c4074509eec6c | e1092507ab8f45 |
| Key 26 | c98e9dd1053406 | 84440fadd281ac | 8b7fd819279b61 | 3d56680db98e15 |
| Key 27 | 20c36794426190 | 10f7900c65fef4 | d7caada0a06ce9 | 0a49af413de66b |
| Key 28 | 964451ceac4fc3 | 30070704c8aa06 | 9297dca1f8c1db | 90a814bbf971a0 |
| Key 29 | 3e904504e18c8a | f287cb4063cb9d | 5d1aaa99dea489 | 626b121ca0504f |
| Key 30 | 290010579c2dfc | 97033445a4d587 | 60cb56ddbba1d9 | 00f9bb7a94a1a7 |
| Key 31 | d7943b69e5b180 | 8051045091c10b | 85d4ad5e5ff2e0 | f485290cc5c1ba |
| Key 32 | 54c7ea5bdd7b43 | d18f282074da20 | 1280161221df6d | baa873c54fdedf |
| Key 33 | 74fb5887c790ba | f2679a98828400 | ca31a5f2406589 | 2d6a56233b8aba |
| Key 34 | 935cfa364e1de0 | a6f0b6042a3dd7 | 1d30e8cb198e6f | a60d0379512312 |
| Key 35 | 03075e159a11ae | 3e5ddad097f5e1 | d1c18bed07d3fa | 942582078dad8b |
| Key 36 | 05d3408a78fb01 | 3ad1f8a2e5958f | cec7ec09245b43 | 8395a4b022082f |
| Key 37 | 0059a5d7a04db3 | f025bb1c085d4f | b08129efedd583 | cb12fe97842b60 |
| Key 38 | 373b634a2c9e40 | 0864213d6d50c1 | 2134cf4ce286e5 | 282ffe78f2f95c |
| Key 39 | 2573bbb4562041 | 9018b0ff3ab170 | edeef9d099b78c | f6491f33c7ef53 |

Table A–1. Sample Device Keys

Transmitter Device #1 examines the KSV of Receiver Device #1 and combines its own secret device keys that correspond to the bit positions of all of the ones in the KSV. Receiver Device #1 examines the KSV of Transmitter Device #1 and combines its own secret device keys that correspond to the bit positions of all of the ones in the KSV. Table A-2 shows the 56-bit binary addition of keys performed by Transmitter Device #1 and Receiver Device #1, and the corresponding equivalent values derived for K_m and K_m' .

| Transmitter Device #1 Sum of Keys Calculation | | Receiver Device #1 Sum of Keys Calculation | |
|--|-----------------------|---|-----------------------|
| Key 0 | 4da4588f131e69 | Key 2 | 24bf2185a36c60 |
| Key 2 | 8a6a47abb9980d | Key 4 | a72e69c5eb6388 |
| Key 3 | f3181b52cbc5ca | Key 8 | 3b9857797d5103 |
| Key 6 | a0d064c8112c41 | Key 9 | 0dd170be615250 |
| Key 7 | b39d5a28242044 | Key 10 | 1a748be4866bb1 |
| Key 9 | 91a47b4a6ce4f6 | Key 12 | 4bbb037899eea1 |
| Key 11 | 8c7fb706ee3fa0 | Key 13 | 190ecf9cc095a9 |
| Key 12 | c02d8c9d7cbc28 | Key 14 | a821c46897447f |
| Key 17 | 482f8e46785498 | Key 15 | 1a8a0bc4298a41 |
| Key 20 | e2d231767b3a54 | Key 16 | aefc0853e62082 |
| Key 21 | 4d581aede66125 | Key 21 | e3053fa3e9fa69 |
| Key 22 | 326082bf7b22f7 | Key 22 | 37d8002881c7d1 |
| Key 23 | f61b463530ce6b | Key 24 | 9e93d41e0811f7 |
| Key 25 | a1e105618d49f9 | Key 25 | 2c4074509eec6c |
| Key 26 | c98e9dd1053406 | Key 32 | 1280161221df6d |
| Key 27 | 20c36794426190 | Key 33 | ca31a5f2406589 |
| Key 28 | 964451ceac4fc3 | Key 34 | 1d30e8cb198e6f |
| Key 32 | 54c7ea5bdd7b43 | Key 36 | cec7ec09245b43 |
| Key 36 | 05d3408a78fb01 | Key 37 | b08129efedd583 |
| Key 38 | 373b634a2c9e40 | Key 39 | edeef9d099b78c |
| RESULT (K_m): | 5309c7d22fcecc | RESULT (K_m') | 5309c7d22fcecc |

Table A-2. Sample K_m Calculation

Table A–3 gives test vectors for the four possible authentication pairs of test keys in Table A–1. The test vectors cover two lines of HDCP Content, each with eight pixels per line. The HDCP Receiver does not support downstream connections (REPEATER = 0).

| | A1 - B1 | A1 - B2 | A2 - B1 | A2 - B2 |
|--|-------------------|-------------------|-------------------|-------------------|
| <i>K_m</i> | 5309c7d22fcec | f6aee46089c923 | 4afe34dbec1205 | a423d78b8676a7 |
| <i>REPEATER</i> <i>A_n</i> | 034271c130c070403 | 0445e62a53ad10fe5 | 083bec2bb01c66e07 | 00351f7175406a74d |
| <i>K_s</i> | 54294b7c040e35 | 4e60d941d0e8b1 | 2c9bef71df792e | 1963deb799ee82 |
| <i>M₀</i> | a02bc815e73d001c | e7d28b9b2f46c49d | 8e1e91f6d8ae4c25 | d05d8c26378a126e |
| <i>R₀</i> | 8ae0 | fb65 | 3435 | 4fd5 |
| <i>K₁</i> | d692b7ee1d40e8 | e46f51311a959a | f3e27849d067c1 | 65f793e160ec27 |
| <i>M₁</i> | 1dbf44e50f523e56 | 445b5c6eebf657ff | 23d89127a5ee6c26 | 68be984885aafef7 |
| Line 1, Pixel 1 | R 59 G c0 B 3e | R 56 G bf B 8a | R 11 G 07 B d2 | R b8 G 2c B 9c |
| Line 1, Pixel 2 | R 9e G e5 B fe | R 2c G 26 B 03 | R b1 G 8f B 7f | R 9b G 34 B e3 |
| Line 1, Pixel 3 | R 9a G f9 B 19 | R 88 G 43 B dc | R 3c G fb B 8c | R 1c G fa B d7 |
| Line 1, Pixel 4 | R 5b G 5d B 6c | R 1d G db B bd | R a3 G 97 B 0c | R 00 G A0 B 08 |
| Line 1, Pixel 5 | R 55 G dc B de | R e6 G 32 B 13 | R 38 G 94 B 3e | R ce G c3 B f4 |
| Line 1, Pixel 6 | R e5 G 87 B 63 | R 36 G 34 B 24 | R ac G 84 B da | R f4 G 36 B 27 |
| Line 1, Pixel 7 | R be G fc B c7 | R 48 G 82 B 8f | R b8 G a4 B 73 | R b6 G 36 B f7 |
| Line 1, Pixel 8 | R a1 G b5 B 65 | R 99 G b9 B db | R 2f G c5 B c0 | R 24 G bd B 8b |
| Horizontal Blank Re-Key | | | | |
| Line 2, Pixel 1 | R 12 G 6b B 14 | R 9c G ac B 7b | R 6c G 64 B c7 | R 73 G 9f B 2e |
| Line 2, Pixel 2 | R 06 G 4a B 73 | R 40 G 11 B d0 | R ba G 05 B 8d | R f6 G 1e B 16 |
| Line 2, Pixel 3 | R f8 G bb B 15 | R aa G 3c B e6 | R 62 G 17 B ff | R e2 G 8c B 59 |
| Line 2, Pixel 4 | R cc G e6 B 21 | R e6 G e9 B ac | R f1 G e5 B df | R d9 G 8a B 86 |
| Line 2, Pixel 5 | R 87 G 95 B 78 | R 7a G d5 B 2e | R c2 G e6 B 92 | R c5 G eb B 96 |
| Line 2, Pixel 6 | R d2 G 03 B f7 | R 94 G 1f B 35 | R 47 G a4 B 94 | R c0 G b3 B ce |
| Line 2, Pixel 7 | R 62 G 81 B 44 | R a7 G 85 B 64 | R 59 G b7 B a1 | R eb G 26 B f3 |
| Line 2, Pixel 8 | R 80 G d8 B 75 | R f7 G 45 B 16 | R 9d G 96 B ea | R f4 G 9e B e1 |

Table A–3. Sample Authentication and Encryption Values (REPEATER = 0)

Table A–4 gives test vectors for the four possible authentication pairs of test keys in Table A–1. The test vectors cover two lines of HDCP Content, each with eight pixels per line. The HDCP Receiver supports downstream connections (REPEATER = 1).

| | A1 - B1 | A1 - B2 | A2 - B1 | A2 - B2 |
|--|-------------------|-------------------|-------------------|-------------------|
| <i>K_m</i> | 5309c7d22fcecc | f6aee46089c923 | 4afe34dbec1205 | a423d78b8676a7 |
| <i>REPEATER</i> <i>A_n</i> | 134271c130c070403 | 1445e62a53ad10fe5 | 183bec2bb01c66e07 | 10351f7175406a74d |
| <i>K_s</i> | bc607b21d48e97 | b7894f1754caaa | fe3717c12f3bb1 | aac4147081a2d0 |
| <i>M₀</i> | 372d3dce38bbe78f | 43d609c682c956e1 | 536dee1e44a58bf4 | 38b57ad3cdd1b266 |
| <i>R₀</i> | 6485 | 3f68 | dd9b | 7930 |
| <i>K₁</i> | 98b281e1876a9a | ffbfef4bc7fd2c | a1ec276b2ddaf0 | 0f0b83888e3209 |
| <i>M₁</i> | 016f9561e001f80d | 2a067368042fa1aa | b365f8813c45db0b | 06471e358f601ce4 |
| Line 1, Pixel 1 | R 33 G 4e B 55 | R bc G 9c B a4 | R 4a G c7 B d3 | R c2 G c8 B 84 |
| Line 1, Pixel 2 | R d2 G 37 B 4e | R 43 G 19 B df | R 30 G a7 B ec | R 2f G 7c B 68 |
| Line 1, Pixel 3 | R 0e G 22 B f5 | R b1 G e0 B 12 | R 2d G 6e B 36 | R 90 G 0b B e5 |
| Line 1, Pixel 4 | R c1 G 31 B 8f | R 27 G d0 B 5a | R e1 G 75 B b6 | R 9e G de B 54 |
| Line 1, Pixel 5 | R dc G a1 B a7 | R d8 G aa B 3d | R 94 G ff B fb | R 78 G cd B 8c |
| Line 1, Pixel 6 | R 27 G e7 B c3 | R 3f G 2a B 64 | R 11 G aa B c1 | R 38 G a5 B b8 |
| Line 1, Pixel 7 | R 56 G 3e B c9 | R 2e G 00 B 0a | R 5c G 71 B 66 | R 32 G ff B 1e |
| Line 1, Pixel 8 | R 10 G dc B 2f | R f2 G 47 B 63 | R be G 33 B 6f | R e4 G d9 B 0c |
| Horizontal Blank Re-Key | | | | |
| Line 2, Pixel 1 | R 73 G 03 B 22 | R e4 G 97 B f1 | R 0b G a7 B ec | R 62 G 0f B 61 |
| Line 2, Pixel 2 | R 69 G 01 B 36 | R df G 15 B 0e | R 4f G 10 B 1e | R 33 G 73 B 52 |
| Line 2, Pixel 3 | R 3d G 27 B 53 | R 2f G 44 B 7b | R fe G 16 b 16 | R cd G 96 B fd |
| Line 2, Pixel 4 | R fe G 41 B 50 | R 0c G 9b B ae | R 52 G e6 B 35 | R 53 G ea B d5 |
| Line 2, Pixel 5 | R a8 G 18 B 8d | R 93 G db B da | R db G 8d B b7 | R 33 G a9 B 31 |
| Line 2, Pixel 6 | R 1a G 02 B 91 | R a7 G f9 B 01 | R 18 G f0 B d9 | R cc G 34 B 86 |
| Line 2, Pixel 7 | R 8c G 29 B ce | R 1a G 39 B 9a | R f5 G 9a B 63 | R 6e G e0 B bb |
| Line 2, Pixel 8 | R 89 G cd B bf | R 4b G 54 B 00 | R d4 G ac B aa | R d2 G fc B 4b |

Table A–4. Sample Authentication and Encryption Values (REPEATER = 1)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x01e35 | 0x00040 | 0x025be | 0x15429 | 01 | 01 | 01 | 01 |
| 1 | 0x01c6b | 0x00081 | 0x04b7c | 0x0a853 | 10 | 10 | 10 | 10 |
| 2 | 0x018d6 | 0x00102 | 0x096f8 | 0x150a7 | 00 | 01 | 11 | 01 |
| 3 | 0x011ac | 0x00204 | 0x02df0 | 0x0a14e | 00 | 00 | 11 | 11 |
| 4 | 0x00358 | 0x00409 | 0x05be0 | 0x1429c | 00 | 00 | 10 | 11 |
| 5 | 0x006b0 | 0x00812 | 0x0b7c0 | 0x08539 | 00 | 00 | 01 | 10 |
| 6 | 0x00d60 | 0x01024 | 0x06f81 | 0x10a72 | 00 | 00 | 00 | 01 |
| 7 | 0x01ac0 | 0x02049 | 0x0df03 | 0x014e4 | 01 | 00 | 00 | 00 |
| 8 | 0x01581 | 0x00093 | 0x0be07 | 0x029c9 | 10 | 01 | 00 | 00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x01cbc | 0x03218 | 0x05712 | 0x0ab75 | 10 | 10 | 01 | 11 |
| 50 | 0x01979 | 0x02431 | 0x0ae24 | 0x156eb | 11 | 00 | 10 | 11 |
| 51 | 0x012f3 | 0x00863 | 0x05c48 | 0x0add7 | 10 | 01 | 01 | 10 |
| 52 | 0x005e6 | 0x010c6 | 0x0b891 | 0x15bae | 01 | 10 | 10 | 01 |
| 53 | 0x00bcc | 0x0218d | 0x07122 | 0x0b75c | 10 | 01 | 01 | 10 |
| 54 | 0x01799 | 0x0031a | 0x0e245 | 0x16eb8 | 01 | 00 | 11 | 00 |
| 55 | 0x00f32 | 0x00634 | 0x0c48b | 0x0dd70 | 10 | 10 | 01 | 10 |
| 56 | 0x01e65 | 0x00c69 | 0x08917 | 0x1bae1 | 00 | 01 | 11 | 01 |

Table A-5. LFSR Module States During A1 - B1 Authentication (REPEATER = 0)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0x22fcec | 0x5309c7d | 0x0000000 | 0xc070403 | 0x271c130 | 0x0000034 | |
| 1 | 0x000084c | 0xf458fff | 0x7f722dc | 0xa5d4b70 | 0x8ea8888 | 0x9f6066d | 0xbe70ee |
| 2 | 0x0ed9f8a | 0xb444236 | 0x3b62e76 | 0x8fa5383 | 0x5d17cd7 | 0x2e71e83 | 0x007023 |
| 3 | 0x70ef0ef | 0x9aa103f | 0x8aa659d | 0x49d0347 | 0xe71b545 | 0xd39af92 | 0xdd51b7 |
| 4 | 0xc8f3da5 | 0x8bbb85f | 0x58047e6 | 0x05add47 | 0xaf2ff95 | 0x4371447 | 0xae10f |
| 5 | 0x6b68710 | 0x1826042 | 0xc20a675 | 0x5693206 | 0xd034757 | 0x71f4c59 | 0xe0e624 |
| 6 | 0xd4c9cf4 | 0x0014506 | 0x6c11733 | 0xf679cf3 | 0xbe06351 | 0x412aafc | 0x6104f9 |
| 7 | 0x2ff2231 | 0x059031a | 0xd84c367 | 0x7c6878b | 0x735a2d2 | 0x2d4fba7 | 0x12c5e4 |
| 8 | 0x1c13406 | 0x516f805 | 0x3e231f5 | 0x61f3f4d | 0xccb03b9 | 0x3030a78 | 0x9f08dc |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0x7dc29a3 | 0x5895932 | 0x26047a5 | 0x12b9cbd | 0xe40581a | 0xc892f27 | 0x1cfd71 |
| 42 | 0xba7d2b0 | 0xf1cfeac | 0x36eb45d | 0xa8bab0f | 0x083213e | 0x38fd0ef | 0xb90f28 |
| 43 | 0xdd26650 | 0x29e8ca4 | 0xbf0109c | 0x04a0c9b | 0xf8cd136 | 0xb6b8827 | 0xf32344 |
| 44 | 0xf928c5b | 0xc70cecd | 0xcc71bb9 | 0x004c69f | 0xf8cfb57 | 0x20d8664 | 0xff2c26 |
| 45 | 0x491d801 | 0xf630446 | 0x43655f6 | 0x26727b8 | 0xb6866b1 | 0x48253f0 | 0xead81d |
| 46 | 0x9281463 | 0x891c25b | 0x2c40a10 | 0xe2e3627 | 0xce25f1d | 0x6fd76d2 | 0x7cb35d |
| 47 | 0x37ef335 | 0xbb8429b | 0xfad91c5 | 0x8bb8770 | 0x94322d6 | 0xbc24e18 | 0x4ac7aa |
| 48 | 0x7bd96ba | 0xee950f7 | 0x749f3d9 | 0xc040e35 | 0x54294b7 | 0x1c61d8e | 0x37d937 |
| Load | 0xc040e35 | 0x54294b7 | 0x1c61d8e | 0xc070403 | 0x271c130 | 0x0000034 | |
| 1 | 0x3772e0b | 0x6595cd5 | 0x93d46aa | 0xf5f1bea | 0x8ea8888 | 0x9f6066d | 0x5d74aa |
| 2 | 0xfcfc369 | 0x18f685a | 0x22626f1 | 0x48ec1f7 | 0x5d17cd7 | 0x083878b | 0x1e60bc |
| 3 | 0x67f044d | 0xd5eb45a | 0x8ca9144 | 0x034b338 | 0x3ac66a8 | 0xdc9e6f6 | 0x4c29b4 |
| 4 | 0x046af2c | 0x992df09 | 0xd7b21a9 | 0x845e47f | 0xce06983 | 0xc50059e | 0x1c3d69 |
| 5 | 0x1a7c13c | 0x6aed6fb | 0x57ba318 | 0xea50517 | 0xc09dcdf | 0xcdbf157 | 0x2d0855 |
| 6 | 0x82ff268 | 0xfd00a63 | 0xf4c6f06 | 0x00bc25d | 0xb24cd67 | 0xa94407a | 0xddb851 |
| 7 | 0xe602372 | 0xe4f1798 | 0x6487e18 | 0x47a81d0 | 0x3ca6b73 | 0x90eea67 | 0x5605dd |
| 8 | 0xa251408 | 0x26ca144 | 0x2c8a821 | 0x700ece4 | 0x1f2ccf5 | 0x575dec4 | 0x44236d |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0xade5581 | 0x026eead | 0x58676ad | 0x19978d8 | 0x207678c | 0x552b693 | 0x65e697 |
| 50 | 0xc1cdfad | 0x29eb9e5 | 0x85864c6 | 0x3a260ed | 0xd817a5a | 0xf2e4743 | 0xa341ef |
| 51 | 0x75114c3 | 0x6923621 | 0xc5367fa | 0x4c7b24b | 0x4c7ad96 | 0x4bf179e | 0x6c2f44 |
| 52 | 0x5e00de1 | 0x31ba2ec | 0x9352a05 | 0x21f7177 | 0x1ce1a8a | 0x5fe9127 | 0xdce5b0 |
| 53 | 0xa8a8b05 | 0x470ad68 | 0x35c28f6 | 0x3eaf43f | 0x194bf81 | 0xb8d5477 | 0x14a02b |
| 54 | 0x56a5801 | 0x5bd1d70 | 0xd724992 | 0xf41fb7d | 0x6aafc2c | 0x3fbf3ef | 0x54c815 |
| 55 | 0x6c30c38 | 0xf15bf0e | 0xfc5799d | 0xb673b37 | 0x921be44 | 0x956fe75 | 0x8ae73d |
| 56 | 0x8451307 | 0x58cff28 | 0x9ee2338 | 0x346ebe6 | 0x189def7 | 0xf04cb0e | 0xe0001c |

Table A-6. Block Module States During A1 - B1 Authentication (REPEATER = 0)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x018b1 | 0x03d0e | 0x06ca0 | 0x14e60 | 01 | 01 | 01 | 01 |
| 1 | 0x01162 | 0x03a1d | 0x0d941 | 0x09cc1 | 00 | 10 | 11 | 00 |
| 2 | 0x002c4 | 0x0343b | 0x0b282 | 0x13983 | 01 | 00 | 11 | 10 |
| 3 | 0x00588 | 0x02876 | 0x06504 | 0x07307 | 10 | 01 | 01 | 11 |
| 4 | 0x00b10 | 0x010ed | 0x0ca09 | 0x0e60f | 01 | 10 | 10 | 10 |
| 5 | 0x01620 | 0x021db | 0x09413 | 0x1cc1e | 10 | 00 | 11 | 00 |
| 6 | 0x00c40 | 0x003b7 | 0x02826 | 0x1983c | 01 | 10 | 10 | 10 |
| 7 | 0x01881 | 0x0076e | 0x0504d | 0x13078 | 11 | 01 | 00 | 11 |
| 8 | 0x01103 | 0x00edd | 0x0a09a | 0x060f0 | 11 | 10 | 01 | 10 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x005c3 | 0x016e4 | 0x0917e | 0x1efbd | 01 | 00 | 00 | 01 |
| 50 | 0x00b86 | 0x02dc8 | 0x022fd | 0x1df7a | 00 | 01 | 00 | 00 |
| 51 | 0x0170d | 0x01b90 | 0x045fb | 0x1bef4 | 00 | 00 | 10 | 00 |
| 52 | 0x00e1b | 0x03721 | 0x08bf6 | 0x17de9 | 00 | 00 | 00 | 10 |
| 53 | 0x01c36 | 0x02e42 | 0x017ed | 0x0fbd3 | 01 | 00 | 00 | 01 |
| 54 | 0x0186d | 0x01c84 | 0x02fda | 0x1f7a6 | 11 | 00 | 00 | 00 |
| 55 | 0x010db | 0x03909 | 0x05fb4 | 0x1ef4d | 10 | 01 | 00 | 00 |
| 56 | 0x001b6 | 0x03212 | 0x0bf68 | 0x1de9b | 01 | 00 | 10 | 00 |

Table A-7. LFSR Module States During A1 – B2 Authentication (REPEATER = 0)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|------------|-----------|-----------|----------|
| Load | 0x089c923 | 0xf6aee46 | 0x0000000 | 0xad10fe5 | 0x5e62a53 | 0x0000044 | |
| 1 | 0x000ace8 | 0x2bbe222 | 0xa84ba32 | 0xf8ee8f0 | 0x4c79444 | 0x649180e | 0xb24463 |
| 2 | 0xbe2db4d | 0xcd43e8 | 0x6cf4c5d | 0xb0bccb3 | 0xcd48ee4 | 0xfbde86b | 0x0ff14d |
| 3 | 0x59aaa16 | 0x420acae | 0x948ddf1 | 0x4f31d66 | 0x5e99939 | 0x8945bd4 | 0x5a7c22 |
| 4 | 0x6716e27 | 0xc71eabf | 0x728216a | 0x948e7ab | 0xb5980ca | 0x3969dfa | 0xe29870 |
| 5 | 0x2b8be74 | 0xc7b7cd8 | 0x1896efd | 0xdd99072 | 0xdd8b36e | 0x9005894 | 0x252d85 |
| 6 | 0x417f923 | 0xf719e90 | 0xd5c1459 | 0xdc0bba0 | 0x6178407 | 0x066cb0a | 0x5195fa |
| 7 | 0x6c1faa9 | 0xf7175fd | 0x50bb276 | 0xcafbcb7c | 0x32a2ec3 | 0xa479ab9 | 0xcd7d1 |
| 8 | 0x90a1447 | 0xad4dd26 | 0x59afdb6 | 0xfa48546 | 0x6ebb9cf | 0x890acc2 | 0xd92360 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0x456a8de | 0x218a73d | 0xefe8143 | 0xdb40d6f | 0x8adb81b | 0x7f17e90 | 0x4b21a1 |
| 42 | 0x5bb75c0 | 0x9e32509 | 0xcd4d66f | 0x94b2edc | 0x91aaaf6 | 0x3894216 | 0x537e81 |
| 43 | 0x692b31d | 0x40c7b06 | 0xeb692c8 | 0x5b4a26a | 0x7c0b63f | 0xb5e23ed | 0x71f997 |
| 44 | 0x4ac7e44 | 0x584dad4 | 0x2606dca | 0xb41c724 | 0xde66448 | 0x90f07c0 | 0x9b4c0f |
| 45 | 0x995c381 | 0xe782e99 | 0x500545a | 0x296761d | 0x33b5aa8 | 0xd7c96dd | 0xcce274 |
| 46 | 0x2a39ef6 | 0xb3509f9 | 0xbd26dfe | 0xf7d1275 | 0xd7972de | 0xa1c5513 | 0xa9e21a |
| 47 | 0xe937d30 | 0x7910780 | 0x03575d7 | 0x0e9e5a9 | 0x235c870 | 0x246431c | 0x8d7b49 |
| 48 | 0xb9af224 | 0x04c8a5f | 0x49c96b1 | 0x1d0e8b1 | 0x4e60d94 | 0x072bad0 | 0x1cfb41 |
| Load | 0x1d0e8b1 | 0x4e60d94 | 0x072bad0 | 0xad10fe5 | 0x5e62a53 | 0x0000044 | |
| 1 | 0x8adc6e8 | 0xb659c1e | 0x70ae5ce | 0x4c36286 | 0x4c79444 | 0x649180e | 0xfeaeeb |
| 2 | 0xe647934 | 0x7ec73a0 | 0xae21cfc | 0x57c3737 | 0xcd48ee4 | 0x131ec75 | 0xe6e976 |
| 3 | 0xfa28037 | 0x602e4c5 | 0xcc87a66 | 0x1fe7698 | 0xf433b91 | 0x990c71a | 0x47ee81 |
| 4 | 0x0d609b0 | 0x76b0413 | 0xbb909ab | 0xc160202 | 0x2e4b770 | 0xd5b0319 | 0x09463e |
| 5 | 0x8f2b473 | 0x00b1039 | 0x54e4007 | 0xf914da7 | 0xbd17a23 | 0x9746424 | 0x341d4a |
| 6 | 0x91fb8aa | 0x6445ea6 | 0x8649c97 | 0x623f7e9 | 0xf5e67b9 | 0xb986c8a | 0x61be45 |
| 7 | 0x88d8719 | 0x4f9ea67 | 0x5195717 | 0x2f6bf08 | 0x42af423 | 0x0f517b2 | 0x38c278 |
| 8 | 0x4e72913 | 0x5e4a60f | 0xef64d8e | 0xa7afa70 | 0x46d5f5f | 0x8599680 | 0x366d9f |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x4dda715 | 0x5cf4582 | 0x66dc877 | 0x4e69fc3 | 0x6790add | 0x692ce89 | 0x40f21c |
| 50 | 0x4db2b7f | 0xfb2f397 | 0x76dedec | 0x20ef253 | 0x81e7d6b | 0xf0b76f9 | 0x9c8062 |
| 51 | 0x6f8bf8a | 0x0579c7f | 0xa79d4cc | 0xf23684b | 0x79e04b8 | 0x71c4515 | 0xef455b |
| 52 | 0x57b4273 | 0x7cc013c | 0x4a37fd9 | 0xa63e183 | 0x13f3943 | 0xaf26eed | 0x9b00a8 |
| 53 | 0x6a718ef | 0x43667bb | 0x91c7a99 | 0x9383356 | 0x3f262d4 | 0xda416b4 | 0xbee7d2 |
| 54 | 0x5764f30 | 0xca377a9 | 0x61cb7fc | 0x75526c2 | 0x5439e56 | 0xc8e2a8a | 0x168b9b |
| 55 | 0x1aac873 | 0xf9340e8 | 0x0ce402a | 0x8504037 | 0x18ad8b4 | 0xb818ef9 | 0xfb2f46 |
| 56 | 0x365eb8d | 0x02468c0 | 0x31071ef | 0x01c71f2 | 0xc7ac9e7 | 0xc1ffc01 | 0x65c49d |

Table A-8. Block Module States During A1 – B2 Authentication (REPEATER = 0)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x0192e | 0x01df7 | 0x077b8 | 0x02c9b | 01 | 01 | 01 | 01 |
| 1 | 0x0125c | 0x03bef | 0x0ef71 | 0x05936 | 11 | 00 | 10 | 10 |
| 2 | 0x004b8 | 0x037df | 0x0dee3 | 0x0b26c | 11 | 10 | 01 | 01 |
| 3 | 0x00970 | 0x02fbf | 0x0bdc7 | 0x164d8 | 01 | 11 | 00 | 11 |
| 4 | 0x012e0 | 0x01f7f | 0x07b8e | 0x0c9b0 | 11 | 01 | 01 | 10 |
| 5 | 0x005c1 | 0x03eff | 0x0f71d | 0x19360 | 01 | 10 | 10 | 11 |
| 6 | 0x00b82 | 0x03dfe | 0x0ee3b | 0x126c1 | 00 | 01 | 11 | 10 |
| 7 | 0x01705 | 0x03bfd | 0x0dc76 | 0x04d82 | 00 | 00 | 11 | 01 |
| 8 | 0x00e0b | 0x037fb | 0x0b8ed | 0x09b04 | 00 | 00 | 10 | 10 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x016ef | 0x004ea | 0x08ffb | 0x18374 | 01 | 11 | 11 | 10 |
| 50 | 0x00dde | 0x009d4 | 0x01ff7 | 0x106e8 | 10 | 11 | 11 | 01 |
| 51 | 0x01bbd | 0x013a9 | 0x03fee | 0x00dd0 | 01 | 01 | 11 | 11 |
| 52 | 0x0177b | 0x02753 | 0x07fdd | 0x01ba0 | 00 | 10 | 11 | 11 |
| 53 | 0x00ef6 | 0x00ea6 | 0x0ffbb | 0x03740 | 01 | 01 | 01 | 11 |
| 54 | 0x01dec | 0x01d4d | 0x0ff77 | 0x06e81 | 10 | 11 | 00 | 11 |
| 55 | 0x01bd9 | 0x03a9b | 0x0feef | 0x0dd02 | 01 | 01 | 10 | 01 |
| 56 | 0x017b3 | 0x03537 | 0x0fddf | 0x1ba04 | 10 | 11 | 01 | 00 |

Table A-9. LFSR Module States During A2 – B1 Authentication (REPEATER = 0)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0xbec1205 | 0x4afe34d | 0x0000000 | 0x1c66e07 | 0xbec2bb0 | 0x0000083 | |
| 1 | 0x888e2ea | 0x414b444 | 0x97a0589 | 0xf087578 | 0x3d5b332 | 0x610f071 | 0x001a91 |
| 2 | 0x4625e41 | 0xcd48c5f | 0x3a77722 | 0x17b01a9 | 0x0638644 | 0xb71a3c5 | 0x892758 |
| 3 | 0xc9402d8 | 0x5ce2e8b | 0x2d46dd1 | 0xcba2da3 | 0x45c8159 | 0x0c27e9f | 0xd3c6e1 |
| 4 | 0x9f4f7b0 | 0x4c9fc33 | 0x7975e63 | 0xb1a5c1f | 0x37140d4 | 0x78f6cfb | 0x916ff8 |
| 5 | 0xa52c6b9 | 0x0ab1bea | 0x3f59b80 | 0x66c7c4e | 0xef8a601 | 0xd5f6819 | 0x21475c |
| 6 | 0xe828e8c | 0x1f4fe28 | 0xf9ae9ca | 0xa6e1944 | 0x11989fd | 0x4338020 | 0x729008 |
| 7 | 0x3d9656f | 0x9313d6c | 0xd525839 | 0x3d3cf97 | 0x2d456aa | 0x5592482 | 0x2c2762 |
| 8 | 0x0b5904f | 0xe168c0e | 0x8549a6c | 0x8e384cb | 0xfd25ff0 | 0x40578b4 | 0xa66b25 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0xf907779 | 0x8add56d | 0xa2bf28b | 0xb6d2591 | 0x8cbe163 | 0x1db3ce9 | 0x55f6f1 |
| 42 | 0xbb149e8 | 0x34b44fe | 0xe899a28 | 0x7ec27a0 | 0xbdae914 | 0xbcc46bf | 0xb1c490 |
| 43 | 0x852bc22 | 0x30c541b | 0x4ba8ad0 | 0xbacaa81 | 0xf2df6bc | 0x7796efa | 0x134543 |
| 44 | 0xe0dcc66 | 0x3380692 | 0x2f59c16 | 0x5875f9a | 0x03ea16f | 0x80bc2ab | 0xf8b3c8 |
| 45 | 0xbd69a67 | 0x11e9f3b | 0xb0d15db | 0xcd318e7 | 0xbcace72 | 0x5aa586f | 0x49d410 |
| 46 | 0x992aba4 | 0x79ccd6c | 0x374d0da | 0x4a507c8 | 0xd761f3d | 0x3849c30 | 0x4d30b7 |
| 47 | 0x02d7a9c | 0x69e0827 | 0x75c491b | 0x1c3734c | 0x1ebaf33 | 0x8e6e1e4 | 0x9df48b |
| 48 | 0x28d5897 | 0x4f55c34 | 0x1bf2686 | 0xdf792e | 0x2c9bef7 | 0x07b1c9f | 0xebdeef |
| Load | 0xdf792e | 0x2c9bef7 | 0x07b1c9f | 0x1c66e07 | 0xbec2bb0 | 0x0000083 | |
| 1 | 0xfd88a6c | 0x1aec3ba | 0x548b6d5 | 0xfb705c6 | 0x3d5b332 | 0x610f071 | 0x636064 |
| 2 | 0x0876369 | 0x710f070 | 0x03a9952 | 0x68afa97 | 0x0638644 | 0x2a048b2 | 0x3a375c |
| 3 | 0xfdcf763 | 0x64400d6 | 0x6888c5c | 0x81f7bc9 | 0xab26acb | 0x5146df0 | 0x1b8dbf |
| 4 | 0x0cb1f80 | 0x6710244 | 0xd810320 | 0x8a558ef | 0xc4934bb | 0xfcbe390 | 0x2fba5d |
| 5 | 0x7a77bb1 | 0x545b44d | 0xacc6c17 | 0xefc1031 | 0x8a7bd55 | 0x6f02498 | 0x66bde4 |
| 6 | 0x629697d | 0xdc585bb | 0x5b8f82d | 0x9e3cd09 | 0xe34bee9 | 0xad76510 | 0x9b04a5 |
| 7 | 0x2d0fd29 | 0x6095002 | 0x10fd4d1 | 0x161afae | 0x9356147 | 0xf76daf9 | 0x9467c6 |
| 8 | 0x7745ff4 | 0xddcd316 | 0x042bd5c | 0x9cc0fc2 | 0x7262896 | 0x73c7ad4 | 0xa7a735 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x3e266d1 | 0xc895108 | 0x65cffa5 | 0xbbf95cd | 0x063edad | 0x9f1843e | 0xd2a1f8 |
| 50 | 0x1aff812 | 0xc8cc3bb | 0x2e34b69 | 0x548d48b | 0x0fc340a | 0x7ca499b | 0xdeebe6 |
| 51 | 0xeb214ef | 0x067b1f8 | 0x19c630a | 0xe7c0a44 | 0x66f4697 | 0x541cbf6 | 0x4420a7 |
| 52 | 0x2403450 | 0x5331c01 | 0x59f99e8 | 0xa39e281 | 0x8971df1 | 0x4c21780 | 0x9f6e12 |
| 53 | 0x96b81f7 | 0xc44f275 | 0x3e91d6c | 0x644040d | 0xd338e4e | 0x0afa6f2 | 0xd38e1e |
| 54 | 0xaf435aa | 0x8ba5ab2 | 0x90519f8 | 0x72a4777 | 0xc552143 | 0x2630971 | 0x6c91f6 |
| 55 | 0x011f064 | 0x0a7aa39 | 0x072d48d | 0x2802af7 | 0x15041a9 | 0xea862e3 | 0x34d8ae |
| 56 | 0x7532414 | 0x0a296c3 | 0xa5510c1 | 0x6891e10 | 0x5316410 | 0x45e1c10 | 0x354c25 |

Table A-10. Block Module States During A2 – B1 Authentication (REPEATER = 0)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x01e82 | 0x0399e | 0x0ef5b | 0x11963 | 01 | 01 | 01 | 01 |
| 1 | 0x01d04 | 0x0333c | 0x0deb7 | 0x032c7 | 00 | 10 | 10 | 10 |
| 2 | 0x01a09 | 0x02678 | 0x0bd6f | 0x0658e | 00 | 01 | 01 | 00 |
| 3 | 0x01413 | 0x00cf0 | 0x07adf | 0x0cb1c | 01 | 10 | 10 | 00 |
| 4 | 0x00827 | 0x019e1 | 0x0f5bf | 0x19638 | 11 | 00 | 11 | 00 |
| 5 | 0x0104e | 0x033c2 | 0x0eb7e | 0x12c71 | 10 | 10 | 10 | 01 |
| 6 | 0x0009d | 0x02785 | 0x0d6fd | 0x058e3 | 01 | 11 | 01 | 00 |
| 7 | 0x0013b | 0x00f0b | 0x0adfb | 0x0b1c7 | 10 | 11 | 10 | 10 |
| 8 | 0x00276 | 0x01e17 | 0x05bf7 | 0x1638e | 00 | 11 | 01 | 01 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x0055e | 0x02e73 | 0x08f69 | 0x07085 | 11 | 00 | 11 | 10 |
| 50 | 0x00abd | 0x01ce7 | 0x01ed3 | 0x0e10b | 11 | 01 | 01 | 01 |
| 51 | 0x0157b | 0x039cf | 0x03da6 | 0x1c217 | 11 | 11 | 00 | 11 |
| 52 | 0x00af6 | 0x0339f | 0x07b4c | 0x1842f | 10 | 11 | 01 | 10 |
| 53 | 0x015ed | 0x0273f | 0x0f699 | 0x1085e | 01 | 01 | 10 | 01 |
| 54 | 0x00bdb | 0x00e7f | 0x0ed32 | 0x010bc | 00 | 10 | 11 | 00 |
| 55 | 0x017b6 | 0x01cff | 0x0da64 | 0x02179 | 00 | 00 | 11 | 01 |
| 56 | 0x00f6c | 0x039fe | 0x0b4c8 | 0x042f3 | 10 | 00 | 01 | 11 |

Table A-11. LFSR Module States During A2 – B2 Authentication (REPEATER = 0)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0xb8676a7 | 0xa423d78 | 0x0000000 | 0x406a74d | 0x51f7175 | 0x0000003 | |
| 1 | 0x666e2c6 | 0x1fb7111 | 0x802f1c8 | 0xf7f1edb | 0x7052777 | 0xf40723 | 0xf05140 |
| 2 | 0x222564c | 0xeacf83b | 0x56392e2 | 0xf8c5faf | 0x9e2408b | 0x787caa9 | 0x91937d |
| 3 | 0x3a7d9e3 | 0x39004ba | 0x11f7a6a | 0xd50bb43 | 0x88db561 | 0x91040c2 | 0x026852 |
| 4 | 0x47614d8 | 0x6494d8a | 0x3b4f25b | 0x4395a00 | 0x53d0514 | 0xe2e383d | 0x3bc587 |
| 5 | 0xdb4e14e | 0x845a7cc | 0xbf7698d | 0xbeab442 | 0xbe1b11f | 0x6a72f32 | 0xb649af |
| 6 | 0x9f50e9a | 0x72b9f8a | 0xe83d832 | 0x2446aa1 | 0x2711b9c | 0xcdda1d2 | 0x76b8c5 |
| 7 | 0x3ea1bc9 | 0x2ef84ca | 0x8b460ed | 0xff20d53 | 0x0d6ac1d | 0x45a75c4 | 0x1cfba1 |
| 8 | 0x16166f2 | 0xaa7c2ef | 0x1d92ed2 | 0x962b376 | 0x2b810f5 | 0x085c932 | 0x34494d |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0x2b7a4ee | 0x76aaca6 | 0x990b686 | 0xe19348b | 0xfea6035 | 0xa9afaf0 | 0x37e446 |
| 42 | 0x2420fda | 0xc71cbcb | 0xd3a43cf | 0x3b01c23 | 0xa98bd4f | 0x4c62274 | 0x58a13f |
| 43 | 0x1b38c46 | 0x7b286a6 | 0x1d6e079 | 0x7fd5dd1 | 0xd04a459 | 0x7c16c08 | 0xd854bb |
| 44 | 0x9ecc174 | 0xa97266e | 0xa162b3f | 0xbab8ead | 0xff58f91 | 0x7740eea | 0x5b3ceb |
| 45 | 0x039d3b7 | 0x039e9b4 | 0xbc7dd68 | 0xfa0a1ce | 0xb752298 | 0xb13d8cf | 0xdf6e53 |
| 46 | 0x5096513 | 0xc3ac236 | 0x4adda17 | 0xdc0290a | 0xff95916 | 0x9f7e6f6 | 0x1dbde4 |
| 47 | 0xc0f65b9 | 0x566da3d | 0x55dab36 | 0x179735f | 0x586589a | 0xba7cd32 | 0xc580c5 |
| 48 | 0x83f87f0 | 0xd6f60e1 | 0xb0ffacc | 0x799ee82 | 0x1963deb | 0xd2ecfc7 | 0x531799 |
| Load | 0x799ee82 | 0x1963deb | 0xd2ecfc7 | 0x406a74d | 0x51f7175 | 0x0000003 | |
| 1 | 0xc4e8ff1 | 0x68b3b95 | 0x5a86976 | 0x3729648 | 0x7052777 | 0xf40723 | 0xda19ca |
| 2 | 0xf2c964d | 0x2f49256 | 0x8ec9541 | 0xb06dc21 | 0x9e2408b | 0x11e91dc | 0xa8a0b8 |
| 3 | 0x26464e7 | 0xab964b8 | 0xc6112c9 | 0x72cfc92 | 0x4417ad5 | 0xc11c247 | 0xe28985 |
| 4 | 0x3b7c3f4 | 0x20c212b | 0x5a8464d | 0x235fdd1 | 0xc5a1984 | 0x7152f6d | 0x8d3851 |
| 5 | 0x0c23381 | 0x1700053 | 0xf79219e | 0x593da63 | 0xc18c5f2 | 0xaec1bce | 0xb484bf |
| 6 | 0x6c9733a | 0xaa9fab7 | 0x3ff3223 | 0x3295feb | 0x8e7c3b9 | 0x394597d | 0x30ed7d |
| 7 | 0xf811f2c | 0x5e2ced9 | 0x7d2aca5 | 0xe469c78 | 0xacc10da | 0xba93ae2 | 0xa60a41 |
| 8 | 0x1ed5c78 | 0xc42186b | 0xc39983c | 0x0c80d4e | 0xccbafef | 0x235ff24 | 0x25ab7f |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x7d252c0 | 0x081db0e | 0x329083e | 0x3036a4c | 0x4c638fc | 0x9042db0 | 0x9c7024 |
| 50 | 0xba0eaa9 | 0x1c0b139 | 0x9f56b08 | 0x4771510 | 0x4f22c73 | 0x6321faf | 0x4732f1 |
| 51 | 0x531015d | 0xe8cd792 | 0xceb6a51 | 0x9327e2f | 0xd768e6e | 0x5ca36be | 0x45edc6 |
| 52 | 0xd1a375c | 0xd925c31 | 0xc37b8b1 | 0xb098639 | 0x8316b0f | 0x7e66ad9 | 0x62404c |
| 53 | 0xb0a7396 | 0xd77e370 | 0xc279e10 | 0x0b2b48e | 0x3e28ad6 | 0xbb19243 | 0xc8d05d |
| 54 | 0xd5c53b3 | 0x9fb7633 | 0xb69eb4a | 0x88af562 | 0x5c2925d | 0x8b95f94 | 0x5c8c26 |
| 55 | 0x33dc74d | 0x9b22ce5 | 0xfd6ece8 | 0x2de6f79 | 0xab859d1 | 0x9fbbcfb | 0x4f378a |
| 56 | 0x96549f5 | 0x5e909b2 | 0xcd1638f | 0x7ed9156 | 0x95fcf36 | 0xa455e43 | 0xd5126e |

A-12. Block Module States During A2 – B2 Authentication (REPEATER = 0)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x01e97 | 0x01d48 | 0x03d90 | 0x1bc60 | 01 | 01 | 01 | 01 |
| 1 | 0x01d2f | 0x03a91 | 0x07b21 | 0x178c0 | 10 | 10 | 11 | 00 |
| 2 | 0x01a5f | 0x03522 | 0x0f642 | 0x0f180 | 01 | 01 | 11 | 10 |
| 3 | 0x014be | 0x02a45 | 0x0ec85 | 0x1e301 | 11 | 00 | 11 | 01 |
| 4 | 0x0097d | 0x0148b | 0x0d90a | 0x1c602 | 11 | 01 | 10 | 11 |
| 5 | 0x012fa | 0x02916 | 0x0b215 | 0x18c05 | 11 | 11 | 00 | 11 |
| 6 | 0x005f4 | 0x0122d | 0x0642a | 0x1180a | 01 | 11 | 10 | 01 |
| 7 | 0x00be9 | 0x0245b | 0x0c855 | 0x03015 | 10 | 11 | 01 | 10 |
| 8 | 0x017d3 | 0x008b6 | 0x090ab | 0x0602b | 01 | 10 | 11 | 00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x01f26 | 0x01ba1 | 0x004d1 | 0x01eb1 | 01 | 10 | 01 | 00 |
| 50 | 0x01e4d | 0x03742 | 0x009a3 | 0x03d62 | 11 | 01 | 10 | 00 |
| 51 | 0x01c9a | 0x02e84 | 0x01346 | 0x07ac5 | 11 | 10 | 01 | 10 |
| 52 | 0x01935 | 0x01d09 | 0x0268d | 0x0f58b | 11 | 01 | 10 | 11 |
| 53 | 0x0126b | 0x03a12 | 0x04d1b | 0x1eb16 | 10 | 10 | 11 | 10 |
| 54 | 0x004d7 | 0x03424 | 0x09a37 | 0x1d62d | 00 | 01 | 11 | 11 |
| 55 | 0x009ae | 0x02849 | 0x0346f | 0x1ac5b | 00 | 10 | 01 | 11 |
| 56 | 0x0135d | 0x01093 | 0x068df | 0x158b7 | 00 | 00 | 11 | 01 |

Table A-13. LFSR Module States During A1 – B1 Authentication (REPEATER = 1)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0x22fcec | 0x5309c7d | 0x0000000 | 0xc070403 | 0x271c130 | 0x0000134 | |
| 1 | 0x000084c | 0xf458fff | 0x7f722dc | 0xa5d4b70 | 0x9fb9989 | 0x9f6066d | 0xbe70ee |
| 2 | 0x0ed9f8a | 0xb444236 | 0x3b62e76 | 0x614bd63 | 0x1d52893 | 0x2e71e83 | 0x102031 |
| 3 | 0x70ef0ef | 0x9aa103f | 0x8aa659d | 0xe37a3ed | 0x6e17dcd | 0x861926f | 0xff57a7 |
| 4 | 0xc8f3da5 | 0x8bbb85f | 0x58047e6 | 0x0ed0c42 | 0xe3299e6 | 0xb4a6b97 | 0xb351be |
| 5 | 0x6b68710 | 0x1826042 | 0xc20a675 | 0x7e45c24 | 0xc398d39 | 0xa08a2f8 | 0x785499 |
| 6 | 0xd4c9cf4 | 0x0014506 | 0x6c11733 | 0x1395270 | 0xf15cafa | 0x1e1176c | 0xe2b59c |
| 7 | 0x2ff2231 | 0x059031a | 0xd84c367 | 0x2769c98 | 0x7d0946d | 0x0bf1b6a | 0xaa109 |
| 8 | 0x1c13406 | 0x516f805 | 0x3e231f5 | 0xe99e086 | 0xde5a665 | 0x22dff84 | 0x2ce1f3 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0x7dc29a3 | 0x5895932 | 0x26047a5 | 0x0755719 | 0x935cfbf | 0xb95d7e0 | 0x24e15b |
| 42 | 0xba7d2b0 | 0xf1cfeac | 0x36eb45d | 0x2a92c58 | 0x699d93d | 0x0eb7293 | 0x87309b |
| 43 | 0xdd26650 | 0x29e8ca4 | 0xbf0109c | 0xfa8cac0 | 0x1e322dc | 0x01e0bb2 | 0xb0f7f3 |
| 44 | 0xf928c5b | 0xc70cecd | 0xcc71bb9 | 0x9b0f0e5 | 0x89e6139 | 0x613ba0b | 0x800977 |
| 45 | 0x491d801 | 0xf630446 | 0x43655f6 | 0x4b35863 | 0x06237ac | 0xca3aa9e | 0x4fdd1d |
| 46 | 0x9281463 | 0x891c25b | 0x2c40a10 | 0xd0db4ac | 0x07ca5ad | 0x3745ef1 | 0x4fd875 |
| 47 | 0x37ef335 | 0xbb8429b | 0xfad91c5 | 0x1f0f4dc | 0xcb0f7af | 0x9858087 | 0x08d905 |
| 48 | 0x7bd96ba | 0xee950f7 | 0x749f3d9 | 0x1d48e97 | 0xbc607b2 | 0x98d9b45 | 0x2247f5 |
| Load | 0x1d48e97 | 0xbc607b2 | 0x98d9b45 | 0xc070403 | 0x271c130 | 0x0000134 | |
| 1 | 0x371f49a | 0x53afa6d | 0x1648023 | 0x7f3108b | 0x9fb9989 | 0x9f6066d | 0x7ccafe |
| 2 | 0x3271b4e | 0x7c7ab77 | 0x269baee | 0x879d9dd | 0x1d52893 | 0x40ef6b9 | 0xf3e3bb |
| 3 | 0x76928cd | 0x3c0c41e | 0x3ddb777 | 0x56aff98 | 0x80f974f | 0x6ed848c | 0x387685 |
| 4 | 0xcb38955 | 0x45f4b5a | 0x44b09f0 | 0x84f827e | 0xd8421d6 | 0x756a06d | 0xcac318 |
| 5 | 0x7e05951 | 0x7b4b7ce | 0x77213e7 | 0x8a65060 | 0x41308c0 | 0x172f316 | 0xbba079 |
| 6 | 0xf43b422 | 0x63ba5f7 | 0x15664df | 0xa546f91 | 0x6e221b2 | 0x5b52502 | 0x15723b |
| 7 | 0x02539f7 | 0x43b1c83 | 0xc6fba6e | 0x8c6d674 | 0x4234c5a | 0x64478ee | 0x6d962d |
| 8 | 0xf69c689 | 0xc41f360 | 0x04591c2 | 0xde7e4f0 | 0x803e2ed | 0x532a599 | 0xa8de7e |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x7bf9fa7 | 0x1a284c6 | 0x739fd87 | 0x461f4a1 | 0xf717fe1 | 0x32b1a29 | 0xf7f563 |
| 50 | 0xd779ca4 | 0xef3a891 | 0x60780be | 0xaa1ce2e | 0x9754a31 | 0x0b0bbfc | 0x664b98 |
| 51 | 0x900446f | 0x80e9401 | 0xc3bf1fb | 0xfebca94 | 0x4e6d371 | 0xe3b1944 | 0xd1dc3b |
| 52 | 0x83b3ab9 | 0x66e50bb | 0xe8c834c | 0xea84947 | 0x53787ed | 0xd15995d | 0xc6c650 |
| 53 | 0xd17e23d | 0xfd8c2ef | 0x618168a | 0x5091ea5 | 0x9e567a1 | 0x6b37e87 | 0x49372d |
| 54 | 0x6cc9afa | 0x560a656 | 0x3dd0e24 | 0xc214d9d | 0x71be498 | 0x3040f5e | 0x0e3dce |
| 55 | 0xcb2c184 | 0xdc614f7 | 0x5d3ee63 | 0x0bba955 | 0xaa48398 | 0xaf781e4 | 0x6438bb |
| 56 | 0x692a85f | 0xde2a833 | 0xff731e2 | 0xafal960 | 0xc8a6055 | 0xbcc4562 | 0x85e78f |

Table A-14. Block Module States During A1 – B1 Authentication (REPEATER = 1)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x01aaa | 0x0154c | 0x0278b | 0x0b789 | 01 | 01 | 01 | 01 |
| 1 | 0x01555 | 0x02a99 | 0x04f17 | 0x16f13 | 10 | 10 | 10 | 11 |
| 2 | 0x00aaa | 0x01533 | 0x09e2f | 0x0de26 | 01 | 01 | 11 | 01 |
| 3 | 0x01554 | 0x02a66 | 0x03c5e | 0x1bc4c | 00 | 10 | 11 | 10 |
| 4 | 0x00aa8 | 0x014cc | 0x078bd | 0x17898 | 00 | 00 | 11 | 11 |
| 5 | 0x01550 | 0x02999 | 0x0f17a | 0x0f131 | 00 | 00 | 10 | 11 |
| 6 | 0x00aa0 | 0x01332 | 0x0e2f4 | 0x1e262 | 01 | 00 | 00 | 11 |
| 7 | 0x01540 | 0x02664 | 0x0c5e9 | 0x1c4c4 | 10 | 10 | 00 | 10 |
| 8 | 0x00a81 | 0x00cc9 | 0x08bd2 | 0x18989 | 01 | 01 | 01 | 00 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x00c45 | 0x01b77 | 0x08130 | 0x052e4 | 00 | 01 | 00 | 01 |
| 50 | 0x0188b | 0x036ef | 0x00260 | 0x0a5c9 | 01 | 00 | 01 | 10 |
| 51 | 0x01117 | 0x02dde | 0x004c1 | 0x14b93 | 00 | 01 | 00 | 01 |
| 52 | 0x0022f | 0x01bbc | 0x00982 | 0x09727 | 01 | 00 | 01 | 00 |
| 53 | 0x0045e | 0x03779 | 0x01304 | 0x12e4f | 11 | 00 | 10 | 00 |
| 54 | 0x008bc | 0x02ef2 | 0x02608 | 0x05c9e | 10 | 10 | 01 | 00 |
| 55 | 0x01179 | 0x01de5 | 0x04c10 | 0x0b93d | 01 | 00 | 10 | 10 |
| 56 | 0x002f3 | 0x03bcb | 0x09821 | 0x1727b | 10 | 00 | 00 | 11 |

Table A-15. LFSR Module States During A1 – B2 Authentication (REPEATER = 1)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0x089c923 | 0xf6aee46 | 0x0000000 | 0xad10fe5 | 0x5e62a53 | 0x0000144 | |
| 1 | 0x000ace8 | 0x2bbe222 | 0xa84ba32 | 0xf8ee8f0 | 0x5d68545 | 0x649180e | 0xb24463 |
| 2 | 0xbe2db4d | 0xcd43e8 | 0x6cf4c5d | 0x5e52253 | 0x8d0daa0 | 0xfbde86b | 0x1fa15f |
| 3 | 0x59aaa16 | 0x420acae | 0x948ddf1 | 0xe59bdcc | 0xd7951b1 | 0x092c03c | 0x787a32 |
| 4 | 0x6716e27 | 0xc71eabf | 0x728216a | 0x84926be | 0xcaad80c | 0xec3a8a5 | 0xf27cef |
| 5 | 0x2b8be74 | 0xc7b7cd8 | 0x1896efd | 0x7d66727 | 0x5c571f8 | 0x8069a85 | 0x88a3ad |
| 6 | 0x417f923 | 0xf719e90 | 0xd5c1459 | 0x76bb30d | 0x5333af4 | 0xa18c913 | 0xd01f1b |
| 7 | 0x6c1faa9 | 0xf7175fd | 0x50bb276 | 0xd91bfa4 | 0x1a7d561 | 0x456e67c | 0xdc6f7c |
| 8 | 0x90a1447 | 0xad4dd26 | 0x59afdb6 | 0xa59b390 | 0x1794cd7 | 0x3453dff | 0x9276f6 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0x456a8de | 0x218a73d | 0xef8e143 | 0x4705e66 | 0xa0ab473 | 0x77d249d | 0x40cba0 |
| 42 | 0x5bb75c0 | 0x9e32509 | 0xcd4d66f | 0x4d4a0e2 | 0x02b580f | 0x2b49a78 | 0x1a3445 |
| 43 | 0x692b31d | 0x40c7b06 | 0xeb692c8 | 0x0d36661 | 0x3a20c13 | 0x8cf85c3 | 0x02f684 |
| 44 | 0x4ac7e44 | 0x584dad4 | 0x2606dca | 0xb39da54 | 0xc47d057 | 0xdca5d5d | 0xf7ef88 |
| 45 | 0x995c381 | 0xe782e99 | 0x500545a | 0x0710574 | 0x54607a7 | 0x42e8a1e | 0xf1a5cc |
| 46 | 0x2a39ef6 | 0xb3509f9 | 0xbd26dfe | 0x284e17f | 0x439d9e4 | 0x4dd18ce | 0x23402b |
| 47 | 0xe937d30 | 0x7910780 | 0x03575d7 | 0xdf9ad7d | 0x3c7791a | 0x6ddd61f | 0x95dc64 |
| 48 | 0xb9af224 | 0x04c8a5f | 0x49c96b1 | 0x754caa | 0xb7894f1 | 0xfcce020 | 0xcdaa1d |
| Load | 0x754caa | 0xb7894f1 | 0xfcce020 | 0xad10fe5 | 0x5e62a53 | 0x0000144 | |
| 1 | 0x1cfb5dd | 0xce2b088 | 0x2eec032 | 0x93dabe7 | 0x5d68545 | 0x649180e | 0x4bbc20 |
| 2 | 0xfa0338f | 0xdd9d11d | 0x26e8f45 | 0x91d34c5 | 0x8d0daa0 | 0xa42f29f | 0x0c1351 |
| 3 | 0x11ffc1e | 0xd8fc06f | 0x846a9c2 | 0x575d169 | 0x5f1d290 | 0xd8d250e | 0x14f5d7 |
| 4 | 0x004ea3a | 0xb8ae70e | 0x00f25c3 | 0x807911a | 0x442cc5a | 0x1f6d6e5 | 0xa0c9b8 |
| 5 | 0xffd1f46 | 0x63fce9f | 0x59e2583 | 0x0965cff | 0x912f65a | 0x9fad256 | 0x28067a |
| 6 | 0x86aa27f | 0x1bfc986 | 0x7559055 | 0xd307ffb | 0x11af6d1 | 0x4d14ec4 | 0xa73184 |
| 7 | 0xe438d81 | 0x2f72c2a | 0x065bebb | 0x2c48a34 | 0x00ed16b | 0xb2430a6 | 0x62d500 |
| 8 | 0xdc88b2a | 0x1b83e3e | 0xc719f35 | 0x3530afd | 0x2435827 | 0x62edd40 | 0xe4b982 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x6e1ecc7 | 0x2126ced | 0xa7ac884 | 0x0a7c511 | 0x278da73 | 0x3c52476 | 0x2afbb7 |
| 50 | 0x9b7983d | 0xd61a93c | 0x560de7f | 0x47467e0 | 0xf5c27f1 | 0x56257fb | 0xbf090b |
| 51 | 0x1848c4a | 0x6946104 | 0x97436c5 | 0x0ac81df | 0xac47979 | 0x84c004f | 0x6fffc7 |
| 52 | 0xb9ff03e | 0xfafd4f8 | 0x030217e | 0xb570368 | 0x4a63c44 | 0x8c9e6ff | 0x8f5af2 |
| 53 | 0x031fbfa | 0x20c4236 | 0x7181797 | 0xa99940c | 0x810cdc7 | 0x6eb5e1a | 0xda43d6 |
| 54 | 0xc67ef5d | 0xdee5ece | 0xb3296c2 | 0xd4f4edd | 0xe33bd04 | 0xcbee012 | 0xc409c6 |
| 55 | 0xa8244d2 | 0x3aef4b0 | 0x5c7f3ad | 0x7eb9d86 | 0xa72a66e | 0x5527b8c | 0x3f82c9 |
| 56 | 0xe3a9d07 | 0xce2e311 | 0xa20cd64 | 0xe15b166 | 0x74e9482 | 0x6a048e0 | 0x6856e1 |

Table A-16. Block Module States During A1 – B2 Authentication (REPEATER = 1)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x01bb1 | 0x012f3 | 0x00be0 | 0x1fe37 | 01 | 01 | 01 | 01 |
| 1 | 0x01763 | 0x025e7 | 0x017c1 | 0x1fc6e | 10 | 11 | 00 | 10 |
| 2 | 0x00ec7 | 0x00bce | 0x02f82 | 0x1f8dd | 01 | 11 | 10 | 01 |
| 3 | 0x01d8f | 0x0179d | 0x05f04 | 0x1f1bb | 00 | 11 | 11 | 00 |
| 4 | 0x01b1f | 0x02f3b | 0x0be08 | 0x1e377 | 10 | 01 | 11 | 01 |
| 5 | 0x0163f | 0x01e77 | 0x07c10 | 0x1c6ef | 01 | 10 | 11 | 11 |
| 6 | 0x00c7f | 0x03cee | 0x0f821 | 0x18ddf | 11 | 00 | 11 | 11 |
| 7 | 0x018fe | 0x039dd | 0x0f043 | 0x11bbf | 10 | 01 | 10 | 11 |
| 8 | 0x011fc | 0x033bb | 0x0e087 | 0x0377e | 11 | 00 | 01 | 11 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x00d13 | 0x03c38 | 0x09f02 | 0x16ea7 | 00 | 11 | 11 | 01 |
| 50 | 0x01a27 | 0x03870 | 0x03e04 | 0x0dd4f | 00 | 10 | 11 | 10 |
| 51 | 0x0144f | 0x030e1 | 0x07c09 | 0x1ba9e | 01 | 00 | 11 | 11 |
| 52 | 0x0089e | 0x021c3 | 0x0f812 | 0x1753c | 11 | 00 | 10 | 11 |
| 53 | 0x0113d | 0x00386 | 0x0f024 | 0x0ea78 | 01 | 10 | 00 | 11 |
| 54 | 0x0027b | 0x0070d | 0x0e048 | 0x1d4f0 | 11 | 01 | 00 | 01 |
| 55 | 0x004f7 | 0x00e1b | 0x0c091 | 0x1a9e0 | 10 | 10 | 01 | 10 |
| 56 | 0x009ee | 0x01c37 | 0x08122 | 0x153c1 | 01 | 00 | 11 | 01 |

Table A-17. LFSR Module States During A2 – B1 Authentication (REPEATER = 1)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0xbec1205 | 0x4afe34d | 0x0000000 | 0x1c66e07 | 0xbec2bb0 | 0x0000183 | |
| 1 | 0x888e2ea | 0x414b444 | 0x97a0589 | 0xf087578 | 0x2c4a233 | 0x610f071 | 0x001a91 |
| 2 | 0x4625e41 | 0xcd48c5f | 0x3a77722 | 0xf95ef49 | 0x467d200 | 0xb71a3c5 | 0x99774a |
| 3 | 0xc9402d8 | 0x5ce2e8b | 0x2d46dd1 | 0x6108d09 | 0xcc49d1 | 0x9c06127 | 0xf1c0f1 |
| 4 | 0x9f4f7b0 | 0x4c9fc33 | 0x7975e63 | 0xe5ed94e | 0xa6cfafe | 0x2632b27 | 0x3ce478 |
| 5 | 0xa52c6b9 | 0x0ab1bea | 0x3f59b80 | 0xc0165ea | 0xb0c5a07 | 0x52300a1 | 0x8091f8 |
| 6 | 0xe828e8c | 0x1f4fe28 | 0xf9ae9ca | 0x7849ad5 | 0x5c4c5dc | 0x8ba6a57 | 0xa1cf90 |
| 7 | 0x3d9656f | 0x9313d6c | 0xd525839 | 0xb882808 | 0xaf4cb4e | 0xe0eb86a | 0xd6d500 |
| 8 | 0x0b5904f | 0xe168c0e | 0x8549a6c | 0x720eb74 | 0xe3f004a | 0xbab4d22 | 0x1000c1 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0xf907779 | 0x8add56d | 0xa2bf28b | 0x170a7c3 | 0x35dc444 | 0x8e8c9fa | 0xa24983 |
| 42 | 0xbb149e8 | 0x34b44fe | 0xe899a28 | 0x298b048 | 0x32b7742 | 0xd005cfd | 0xea1835 |
| 43 | 0x852bc22 | 0x30c541b | 0x4ba8ad0 | 0x3eae65f | 0x158d372 | 0xcadc45a | 0xe1162f |
| 44 | 0xe0dcc66 | 0x3380692 | 0x2f59c16 | 0xe406ae7 | 0x605aa2c | 0x37ac1ab | 0x9e5a09 |
| 45 | 0xbd69a67 | 0x11e9f3b | 0xb0d15db | 0xedd1223 | 0x38397e2 | 0xa9aee0 | 0xb5955f |
| 46 | 0x992aba4 | 0x79ccd6c | 0x374d0da | 0x50ca3ca | 0x24fe7c5 | 0xab2ac15 | 0x8680ef |
| 47 | 0x02d7a9c | 0x69e0827 | 0x75c491b | 0xc2e075e | 0x27ef684 | 0x5569487 | 0x2f26b1 |
| 48 | 0x28d5897 | 0x4f55c34 | 0x1bf2686 | 0x12f3bb1 | 0xfe3717c | 0x4903692 | 0x490497 |
| Load | 0x12f3bb1 | 0xfe3717c | 0x4903692 | 0x1c66e07 | 0xbec2bb0 | 0x0000183 | |
| 1 | 0xa4b6650 | 0x0726307 | 0x51cb288 | 0x775f7b9 | 0x2c4a233 | 0x610f071 | 0xc6a91b |
| 2 | 0xb19afdf | 0x140ae14 | 0x6402f81 | 0xe318db4 | 0x467d200 | 0xbf592b0 | 0x5dcbb5 |
| 3 | 0x9159d90 | 0x4dec573 | 0xca5821f | 0xc90434c | 0x333bc3a | 0x8fd699e | 0x93cd20 |
| 4 | 0x958e6ac | 0x17a4c19 | 0x95d7367 | 0xf18d3a1 | 0xa0182d7 | 0x0608db9 | 0xa81d43 |
| 5 | 0x5637028 | 0x7fd4c2b | 0x235d32a | 0x012244a | 0x760a344 | 0x856619e | 0x73e788 |
| 6 | 0x30b4ded | 0x6cf793e | 0x75d7724 | 0x29dc723 | 0x363fbe6 | 0xc615e74 | 0x18faae |
| 7 | 0x0be6fa2 | 0x96a92c7 | 0x013fcf0 | 0x40c3e38 | 0x693a50c | 0x2c0f81f | 0x429d33 |
| 8 | 0x302975b | 0x762a198 | 0x0e1b7f2 | 0x0b403f5 | 0x1493775 | 0x0326946 | 0x743991 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0xaf2d2bb | 0xe13c1bf | 0xd5bf725 | 0xa861b70 | 0x30baed9 | 0x595a054 | 0xae82d |
| 50 | 0xd6b547a | 0xbcc8c65 | 0xaf1fe4b | 0x5e1ed44 | 0x3bdcf3f | 0x775ef00 | 0x574a8e |
| 51 | 0x8e47e11 | 0x1a9467f | 0xc074e74 | 0xf94ad69 | 0x78cca09 | 0x3f48c38 | 0x6d424b |
| 52 | 0x819e9c2 | 0xed51704 | 0x9cd77e9 | 0x03dd484 | 0x3b38f11 | 0x9e92103 | 0xbcdd40 |
| 53 | 0x274fca5 | 0x50dde0a | 0xe25ca16 | 0x462e7d7 | 0xa603ab6 | 0x48da00f | 0x97536d |
| 54 | 0x910b283 | 0x5dcf83d | 0x3a4f75f | 0xecacd6b | 0x7c0fb7b | 0x1b60ea8 | 0x0eee1e |
| 55 | 0xea791f3 | 0x92b86cf | 0x3be152b | 0xe0f4dc5 | 0xd3e247e | 0x6996c21 | 0xdd44a5 |
| 56 | 0xcb67cb7 | 0xab75038 | 0xf8a92f2 | 0x754b3d8 | 0x47f242a | 0x5d3f58c | 0x9b8bf4 |

Table A-18. Block Module States During A2 – B1 Authentication (REPEATER = 1)

| Sequence | LFSR 0 | LFSR 1 | LFSR 2 | LFSR 3 | SH 0 | SH 1 | SH 2 | SH 3 |
|-------------|---------|---------|---------|---------|------|------|------|------|
| Load | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| 47 | | | | | | | | |
| 48 | | | | | | | | |
| Load | 0x002d0 | 0x0281a | 0x08a38 | 0x0aac4 | 01 | 01 | 01 | 01 |
| 1 | 0x005a1 | 0x01034 | 0x01471 | 0x15588 | 00 | 11 | 00 | 10 |
| 2 | 0x00b42 | 0x02069 | 0x028e2 | 0x0ab11 | 00 | 10 | 01 | 00 |
| 3 | 0x01685 | 0x000d2 | 0x051c5 | 0x15623 | 01 | 00 | 11 | 00 |
| 4 | 0x00d0a | 0x001a5 | 0x0a38b | 0x0ac47 | 00 | 01 | 10 | 01 |
| 5 | 0x01a14 | 0x0034b | 0x04716 | 0x1588f | 01 | 00 | 11 | 00 |
| 6 | 0x01428 | 0x00697 | 0x08e2c | 0x0b11e | 10 | 00 | 01 | 10 |
| 7 | 0x00850 | 0x00d2e | 0x01c58 | 0x1623d | 01 | 01 | 00 | 01 |
| 8 | 0x010a1 | 0x01a5d | 0x038b1 | 0x0c47b | 11 | 00 | 01 | 10 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x017d1 | 0x002a0 | 0x0c549 | 0x10b2f | 10 | 00 | 00 | 11 |
| 50 | 0x00fa2 | 0x00540 | 0x08a93 | 0x0165f | 11 | 00 | 00 | 01 |
| 51 | 0x01f44 | 0x00a80 | 0x01526 | 0x02cbe | 01 | 10 | 00 | 10 |
| 52 | 0x01e89 | 0x01501 | 0x02a4c | 0x0597c | 10 | 00 | 01 | 01 |
| 53 | 0x01d12 | 0x02a03 | 0x05498 | 0x0b2f8 | 01 | 00 | 00 | 10 |
| 54 | 0x01a24 | 0x01406 | 0x0a931 | 0x165f1 | 11 | 00 | 00 | 00 |
| 55 | 0x01449 | 0x0280d | 0x05263 | 0x0cbe2 | 10 | 01 | 00 | 00 |
| 56 | 0x00892 | 0x0101a | 0x0a4c6 | 0x197c5 | 01 | 11 | 00 | 00 |

Table A-19. LFSR Module States During A2 – B2 Authentication (REPEATER = 1)

| Sequence | Kx | Ky | Kz | Bx | By | Bz | Output |
|-------------|-----------|-----------|-----------|-----------|-----------|-----------|----------|
| Load | 0xb8676a7 | 0xa423d78 | 0x0000000 | 0x406a74d | 0x51f7175 | 0x0000103 | |
| 1 | 0x666e2c6 | 0x1fb7111 | 0x802f1c8 | 0xf7f1edb | 0x6143676 | 0x0f40723 | 0xf05140 |
| 2 | 0x222564c | 0xeacf83b | 0x56392e2 | 0x162b14f | 0xde614cf | 0x787caa9 | 0x81c36f |
| 3 | 0x3a7d9e3 | 0x39004ba | 0x11f7a6a | 0x7fa1be9 | 0x01d7de9 | 0x01b5c18 | 0x206e42 |
| 4 | 0x47614d8 | 0x6494d8a | 0x3b4f25b | 0x25cec72 | 0x4a836ae | 0x2534ecb | 0xeaf263 |
| 5 | 0xdb4e14e | 0x845a7cc | 0xbf7698d | 0x4a208a3 | 0x30e92d8 | 0xa659bcf | 0x84539a |
| 6 | 0x9f50e9a | 0x72b9f8a | 0xe83d832 | 0xe5d510e | 0x442ab7d | 0x3cd4cd1 | 0xc822c1 |
| 7 | 0x3ealbc9 | 0x2ef84ca | 0x8b460ed | 0x1b4eb4a | 0xd2f25b6 | 0xeb1adbf | 0x37ed7a |
| 8 | 0x16166f2 | 0xaa7c2ef | 0x1d92ed2 | 0x1b5c7a1 | 0x25d261d | 0xf639672 | 0x0312ca |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 41 | 0x2b7a4ee | 0x76aaca6 | 0x990b686 | 0x7b9285b | 0xcea3e3a | 0xf0550a8 | 0xab9a38 |
| 42 | 0x2420fda | 0xc71cbcb | 0xd3a43cf | 0xaca9532 | 0xf5455b6 | 0xd465e50 | 0x6ccddb |
| 43 | 0x1b38c46 | 0x7b286a6 | 0x1d6e079 | 0xf25ba51 | 0xad5a148 | 0xbbb5468 | 0x0532d5 |
| 44 | 0x9ecc174 | 0xa97266e | 0xa162b3f | 0x3954aab | 0xc8cae06 | 0xe9ffa6a | 0x59de69 |
| 45 | 0x039d3b7 | 0x039e9b4 | 0xbc7dd68 | 0x76e0d88 | 0xf667013 | 0x5ca7484 | 0xa81811 |
| 46 | 0x5096513 | 0xc3ac236 | 0x4adda17 | 0x96a7579 | 0xccfde0b | 0x56352ce | 0x1d33c5 |
| 47 | 0xc0f65b9 | 0x566da3d | 0x55dab36 | 0x6ff16c4 | 0x198a2d8 | 0x97f7aef | 0x1ad8fa |
| 48 | 0x83f87f0 | 0xd6f60e1 | 0xb0ffacc | 0x081a2d0 | 0xaac4147 | 0x7734dfc | 0xd23a1e |
| Load | 0x081a2d0 | 0xaac4147 | 0x7734dfc | 0x406a74d | 0x51f7175 | 0x0000103 | |
| 1 | 0xlace2d1 | 0x14061ea | 0x0c44875 | 0xd086746 | 0x6143676 | 0x0f40723 | 0x4e6747 |
| 2 | 0xd88d8d4 | 0xdb895bd | 0x7e74e49 | 0x413ed54 | 0xde614cf | 0xdb03edb | 0x8d2332 |
| 3 | 0x95561d4 | 0xe90f704 | 0xfe35448 | 0x1cdbacf | 0xcd1bfeb | 0xbe705ef | 0xb7c367 |
| 4 | 0x6aabee2 | 0xeb64c24 | 0xb674c2a | 0xef4f673 | 0xd302546 | 0x75b8516 | 0x1c6484 |
| 5 | 0xfe3250b | 0xb039351 | 0x4a14ff3 | 0x5a879c9 | 0xd849947 | 0xa65f3bb | 0xb37177 |
| 6 | 0x7a6f7cc | 0xfbd0e84 | 0xce6bee1 | 0x0ad85e1 | 0x7a6282a | 0x7f78db0 | 0xe41787 |
| 7 | 0x581bf9a | 0xf637058 | 0x06205c2 | 0x0ff292e | 0x7d65bcc | 0x84473cb | 0x85be3b |
| 8 | 0x662ea9c | 0x99bf90a | 0x290e00f | 0xbad8a31 | 0x94d72cc | 0xb929192 | 0x5857cf |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 49 | 0x68a55fc | 0x5bc6412 | 0x5ca2595 | 0x14cc21e | 0x30c7bd6 | 0xb826f67 | 0x06a265 |
| 50 | 0xb7cd0f6 | 0x33813a4 | 0x7b3e868 | 0x78c9a94 | 0x94e586f | 0x1ea87f3 | 0x18c4db |
| 51 | 0x3cb03ff | 0xcb86820 | 0x7fa96de | 0x71c1620 | 0x7c602e4 | 0x60688eb | 0xc9abf0 |
| 52 | 0x1fee845 | 0x0a02783 | 0x371bc65 | 0x7d3cf2c | 0xcf8006d | 0x3206d1e | 0xb00bfa |
| 53 | 0x8b4c9c9 | 0x8c51ea6 | 0xd91c1db | 0xec51ba3 | 0x5652523 | 0x36ba88d | 0xb238b5 |
| 54 | 0xb5a6da8 | 0x7caf32e | 0x1724577 | 0x1a1a940 | 0xf96eb52 | 0x8929566 | 0x1c7ad3 |
| 55 | 0x8bde531 | 0xcbd6c1e | 0x0f35c36 | 0xc66fea6 | 0x0c3c692 | 0x6561bba | 0x79cdd1 |
| 56 | 0x6138d30 | 0x09b02ea | 0x3d45fab | 0x81c0f48 | 0xaa5211b | 0xbc2973b | 0x30b266 |

Table A-20. Block Module States During A2 – B2 Authentication (REPEATER = 1)

Table A-21 provides cryptographic parameters for verifying the facsimile SRMs provided in Table A-22. These parameters are not used in production devices or SRMs. Refer to Table 5-3 for the cryptographic parameters used in production SRMs.

| Parameter | Value (hexadecimal) |
|---------------|--|
| Prime Modulus | See Table 5-3. |
| Prime Divisor | See Table 5-3. |
| Generator | See Table 5-3. |
| Public Key | 8d13e19f340e11ceb0db95eb3eb0743195dfc402b7dc8caac7752e47ded8e8c00b115f8e5e08c7a664cbbba39786efd71c012e8394af79cd01f722a0926952e8de857cbd2e7295e6b1d88cc0ff5dcc0ab16d14fa11a48eb50fca83a37ed18de16d973565df8a784e854296ac700b2e030fd2a98183aa7b22a63b57bee5c2b946 |

Table A-21. Cryptographic Parameters for Verifying Facsimile SRMs

| KSVs Revoked | SRM Version | Value (sequence of hexadecimal bytes) |
|---------------------------|-------------|--|
| 511ef21acd, e72697f401 | 0005 | 80 00 00 05 01 00 00 36 02 51 1e f2 1a cd e7 26 97 f4 01 97 10 19 92 53 e9 f0 59 95 a3 7a 3b fe e0 9c 76 dd 83 aa c2 5b 24 b3 36 84 94 75 34 db 10 9e 3b 23 13 d8 7a c2 30 79 84 |

Table A-22. Facsimile SRMs

Table A-23 provides a sample SRM with an empty list of KSVs revoked. This sample SRM is signed the the production cryptographic parameters (Table 5-3):

| KSVs Revoked | SRM Version | Value (sequence of hexadecimal bytes) |
|--------------|-------------|---|
| (none) | 0001 | 80 00 00 01 00 00 00 2b d2 48 9e 49 d0 57 ae 31 5b 1a bc e0 0e 4f 6b 92 a6 ba 03 3b 98 cc ed 4a 97 8f 5d d2 27 29 25 19 a5 d5 f0 5d 5e 56 3d 0e |

Table A-23. Sample Empty SRM

Tables A-24 and A-25 provide two samples of V or V' values computation for repeaters. Sequences of one-byte values are ascending in time, memory addresses, or port addresses.

| | |
|-----------------------|---|
| Ksv0 | 0x35796a172e |
| Ksv1 | 0x478e71e20f |
| Ksv2 | 0x74e85397a6 |
| Bstatus | 0x0203 |
| M0 | 0x372d3dce38bbe78f |
| SHA-1 transform input | 2e 17 6a 79 35 0f e2 71 8e 47 a6 97 53 e8 74 03 02 8f e7 bb 38 ce 3d 2d 37 80 00 c8 |
| SHA-1 H0b | 0x0fcbd586 |
| SHA-1 H1 | 0xefc107ef |
| SHA-1 H2 | 0xccd70ald |
| SHA-1 H3 | 0xb1186dda |
| SHA-1 H4 | 0x1fb3ff5e |
| KSV FIFO (port 43) | 2e 17 6a 79 35 0f e2 71 8e 47 a6 97 53 e8 74 |
| Ports 41-42 | 03 02 |
| Ports 20-23 | 86 d5 cb 0f |
| Ports 24-27 | ef 07 c1 ef |
| Ports 28-2b | 1d 0a d7 cc |
| Ports 2c-2f | da 6d 18 b1 |
| Ports 30-33 | 5e ff b3 1f |

Table A-24 V computation for 3 repeaters with depth 2

| | |
|------------------------------|--|
| Ksv0 | 0x23a19cbe4d |
| Ksv1 | 0x0d7e993570 |
| Ksv2 | 0xd3458d7d09 |
| Ksv3 | 0xe2a2dce946 |
| Ksv4 | 0xf3148e499d |
| Ksv5 | 0x9345e95ca3 |
| Ksv6 | 0xda8cb307c5 |
| Ksv7 | 0x9901fa75ac |
| Ksv8 | 0x697f3a3c20 |
| Ksv9 | 0xc89758ed19 |
| Ksv10 | 0x2de3a8e869 |
| Ksv11 | 0xe0d9295af2 |
| Ksv12 | 0x6cde88a8b3 |
| Ksv13 | 0x6e219499f5 |
| Ksv14 | 0x31e3e1a572 |
| Bstatus | 0x30f |
| M0 | 0x372d3dce38bbe78f |
| First SHA-1 transform input | 4d be 9c a1 23 70 35 99 7e 0d 09 7d 8d 45 d3 46 e9 dc a2 e2 9d 49 8e 14 f3 a3 5c e9 45 93 c5 07 b3 8c da ac 75 fa 01 99 20 3c 3a 7f 69 19 ed 58 97 c8 69 e8 a8 e3 2d f2 5a 29 d9 e0 b3 a8 88 de |
| Second SHA-1 transform input | 6c f5 99 94 21 6e 72 a5 e1 e3 31 0f 03 8f e7 bb 38 ce 3d 2d 37 80 02 a8 |
| SHA-1 H0 | 0x6dad1995 |
| SHA-1 H1 | 0x7c0a62fc |
| SHA-1 H2 | 0x1b98fff2 |
| SHA-1 H3 | 0x0159cbb7 |
| SHA-1 H4 | 0xaeae604fe |
| KSV FIFO (port 43) | 4d be 9c a1 23 70 35 99 7e 0d 09 7d 8d 45 d3 46 e9 dc a2 e2 9d 49 8e 14 f3 a3 5c e9 45 93 c5 07 b3 8c da ac 75 fa 01 99 20 3c 3a 7f 69 19 ed 58 97 c8 69 e8 a8 e3 2d f2 5a 29 d9 e0 b3 a8 88 de 6c f5 99 94 21 6e 72 a5 e1 e3 31 |
| Ports 41-42 | 0f 03 |
| Ports 20-23 | 95 19 ad 6d |
| Ports 24-27 | fc 62 0a 7c |
| Ports 28-2b | f2 ff 98 1b |

| | |
|-------------|-------------|
| Ports 2c-2f | b7 cb 59 01 |
| Ports 30-33 | Fe 04 e6 ea |

Table A-25 V computation for 15 repeaters with depth 3

Here are the intermediate results for a DSA signature verification of the empty SRM in the specification table 5-1. This uses the real public key in the specification in table 5-3.

Table A-26, Empty SRM verification test vectors

```

message = 0x80, 0, 0, 1, 0, 0, 0, 0x2b
SHA-1 digest = e241fca6a4c634f1337e18b042d2f5fee7643c67
w = 4b5b63222239a1c0f40ca135ee5625048ff9d6ae
u1 = cee7d084a5356092cbde569ddb7587a75e9e47d4
u2 = a6ed36cacf4f02a627de1d27ed8a92083e023823
G^u1 =
147492f48e5351ddb71bd03f906759f06878d1f44580b8e1b0bed9c85bc0e2ab1a80f01
961b868fe3de271c6ca3ed536d368f6f55d100f44d4497319e7e57a4dd413dc32972331
0d08b40891bc29a130d0aab75c1c428b059f54aea62d4f220c6c14ca521a6141af7acb8
ee7aac68a6167a2270ebed875344979b88bfe63827b
Y^u2 =
c5d2c41ec216cc4ac1a07f6f6ad9caf504dc8cb71bf4f6e764e49bd6aec02c99a98f54a
55712fad3d86e56944ddad9ca6401c632948c5bbc9547078bf590b643f324b6d13c6a29
526f0d1d9ebfeb323f0d1c8f7109b75356963d227c6cb7fe4ce5f67fdadd6820696a27c
a9b2387b989d3384ab339f5521dc035f2ad09629cdc
product =
19eebd37c962302901b20dc529759b4d05a62b9602c97bae5a002e8c77178769e27f3ce
5ae5bdbd8427ebe0f7ace4288bae377f5cabca52c855b2f49d6e364186e636b6aa86f4a
5b4c7a8df0e1e174a7473469f00a438771c39fc735dd1f8b1e897bb798dea8df80287c0
9d5445e438bcc4c68313450f616c11858dd77869e1
v = d2489e49d057ae315b1abce00e4f6b92a6ba033b

```

| | REPEATER = 0 | | | | REPEATER = 1 | | | |
|----------|--------------|---------|---------|---------|--------------|---------|---------|---------|
| | A1 - B1 | A1 - B2 | A2 - B1 | A2 - B2 | A1 - B1 | A1 - B2 | A2 - B1 | A2 - B2 |
| R_0 | 8ae0 | fb65 | 3435 | 4fd5 | 6485 | 3f68 | dd9b | 7930 |
| R_1 | 6153 | d543 | 8991 | 0488 | ed21 | 6006 | ccc3 | e26b |
| R_2 | d189 | 295a | 0e1c | 0659 | 97e0 | dc65 | 960e | d9ff |
| R_3 | 2b74 | 1c17 | 20b0 | e1a7 | ec2b | 32dd | 7232 | 9ae9 |
| R_4 | 2147 | 4315 | 0ac4 | a809 | f82a | 5f01 | 1d5e | 39da |
| R_5 | 2570 | cc79 | 5412 | b077 | 4167 | 3618 | 8cba | c0b7 |
| R_{06} | 9204 | 8d46 | 7f3d | 52f4 | 03c3 | a885 | 5271 | 4d40 |
| R_{07} | 9648 | dac2 | 175c | b9ea | b900 | fcf1 | af20 | 426d |
| R_{08} | 6b01 | 8255 | 3d2a | fe59 | d672 | 669e | ad26 | 2464 |
| R_{09} | 6796 | 2642 | 2ba5 | 04b7 | 7ee6 | 3a5e | 5510 | 627c |
| R_{10} | a193 | 8768 | daa7 | efaa | 3d91 | 69b7 | 6654 | c7c1 |
| R_{11} | bffd | 498b | 57fd | 8b58 | b21b | ac84 | 10c3 | 8e5d |
| R_{12} | 4b20 | f41d | 6d17 | 7dbf | 77bf | e03f | 5e54 | 5a04 |
| R_{13} | 85b8 | 6ab9 | 7a43 | 6b05 | a95b | 4778 | 3b2c | 446b |
| R_{14} | f838 | 738e | f6a8 | 7fdf | 36de | 9d22 | bf2c | 0749 |
| R_{15} | 3e80 | d655 | f25c | 8cb1 | d19b | 56b2 | f389 | 0333 |

| | | | | | | | | |
|----------------------------|------|------|------|------|------|------|------|------|
| R_{16} | 390f | 9aff | ec0a | d3b9 | 3137 | c601 | 31a6 | 94d5 |
| R_{17} | 2ecb | 04f2 | f4b8 | 857f | 40d4 | 1eda | da43 | 8d22 |
| R_{18} | c279 | 93a5 | 1ca8 | 3153 | 6586 | bca0 | 0cd2 | 3c9d |
| R_{19} | 596f | 8f5d | 169c | 8fba | cfcb | 2a18 | 1be6 | 5406 |

Table A-27. R_i values

| Frame | A1 - B1 | | A1 - B2 | | A2 - B1 | | A2 - B2 | |
|-------------------------|---------|-------|---------|-------|---------|-------|---------|-------|
| | R_i | P_j | R_i | P_j | R_i | P_j | R_i | P_j |
| R_0 | 8ae0 | | fb65 | | 3435 | | 4fd5 | |
| 16 | | 0xD1 | | 0x8B | | 0x17 | | 0x61 |
| 32 | | 0x88 | | 0xD9 | | 0xB0 | | 0x06 |
| 48 | | 0xB4 | | 0xA4 | | 0x9C | | 0x41 |
| 64 | | 0x38 | | 0x68 | | 0xF1 | | 0xC2 |
| 80 | | 0xA7 | | 0x81 | | 0x54 | | 0xD0 |
| 96 | | 0x4C | | 0x6A | | 0x3C | | 0x96 |
| 112 | | 0xE4 | | 0x61 | | 0x34 | | 0xB8 |
| 128 | 0x6153 | 0x53 | 0xD543 | 0x43 | 0x8991 | 0x91 | 0x0488 | 0x88 |
| 144 | | 0x12 | | 0x70 | | 0xD8 | | 0xFA |
| 160 | | 0x28 | | 0x6F | | 0xE5 | | 0x38 |
| 176 | | 0x2D | | 0x05 | | 0xA1 | | 0x82 |
| 192 | | 0xE1 | | 0x6D | | 0xAA | | 0xF5 |
| 208 | | 0xD5 | | 0xB3 | | 0x4C | | 0x11 |
| 224 | | 0xF0 | | 0xA2 | | 0x7F | | 0xC0 |
| 240 | | 0x92 | | 0x97 | | 0x1D | | 0x10 |
| 256 | 0xD189 | 0x89 | 0x295A | 0x5A | 0x0E1C | 0x1C | 0x0659 | 0x59 |
| 272 | | 0xCD | | 0xA7 | | 0x52 | | 0xAD |
| 288 | | 0x9F | | 0x11 | | 0x24 | | 0xE1 |
| 304 | | 0xB3 | | 0x97 | | 0x8D | | 0xF8 |
| 320 | | 0x2F | | 0x7F | | 0xBF | | 0x58 |
| 336 | | 0x49 | | 0x10 | | 0x44 | | 0xE8 |
| 352 | | 0x11 | | 0x69 | | 0xB6 | | 0x0E |
| 368 | | 0x95 | | 0xEF | | 0xC5 | | 0x12 |
| 384 | 0x2B74 | 0x74 | 0x1C17 | 0x17 | 0x20B0 | 0xB0 | 0xE1A7 | 0xA7 |
| 400 | | 0x45 | | 0xE7 | | 0xD6 | | 0xF2 |
| 416 | | 0x2C | | 0x57 | | 0xF6 | | 0x5F |
| 432 | | 0x07 | | 0xF8 | | 0xB6 | | 0xB5 |
| 448 | | 0xC8 | | 0x5C | | 0xAC | | 0x09 |
| 464 | | 0x4B | | 0x27 | | 0x2C | | 0xBD |
| 480 | | 0x93 | | 0xAF | | 0x14 | | 0x5A |
| 496 | | 0xEF | | 0x6C | | 0xFE | | 0xE0 |
| 512 | 0x2147 | 0x47 | 0x4315 | 0x15 | 0x0AC4 | 0xC4 | 0xA809 | 0x09 |
| 528 | | 0xE7 | | 0x91 | | 0x5D | | 0xE4 |
| 544 | | 0x4C | | 0x29 | | 0xC5 | | 0xBB |
| 560 | | 0xDE | | 0xDA | | 0xE7 | | 0xC3 |
| 576 | | 0xD8 | | 0x6C | | 0xB6 | | 0xEB |
| 592 | | 0xC8 | | 0x89 | | 0xB4 | | 0x3B |
| 608 | | 0xC3 | | 0xB1 | | 0x97 | | 0x7E |
| 624 | | 0x05 | | 0x38 | | 0x53 | | 0x62 |
| 640 | 0x2570 | 0x70 | 0xCC79 | 0x79 | 0x5412 | 0x12 | 0xB077 | 0x77 |

Table A-28. R_i and P_j values (REPEATER = 0)

| Frame | A1 - B1 | | A1 - B2 | | A2 - B1 | | A2 - B2 | |
|----------------------|---------|------|---------|------|---------|------|---------|------|
| | Ri | Pj | Ri | Pj | Ri | Pj | Ri | Pj |
| R₀ | 6485 | | 3f68 | | dd9b | | 7930 | |
| 16 | | 0x73 | | 0x88 | | 0xE9 | | 0xF9 |
| 32 | | 0x32 | | 0x03 | | 0x1D | | 0x81 |
| 48 | | 0xB3 | | 0x24 | | 0x97 | | 0x73 |
| 64 | | 0xCD | | 0x05 | | 0xC0 | | 0x3C |
| 80 | | 0x98 | | 0x00 | | 0x11 | | 0xBC |
| 96 | | 0x18 | | 0x00 | | 0x25 | | 0x21 |
| 112 | | 0x97 | | 0xA1 | | 0x43 | | 0x00 |
| 128 | 0xED21 | 0x21 | 0x6006 | 0x06 | 0xCCC3 | 0xC3 | 0xE26B | 0x6B |
| 144 | | 0x8F | | 0x91 | | 0x37 | | 0x48 |
| 160 | | 0x09 | | 0xC6 | | 0x5D | | 0x77 |
| 176 | | 0x32 | | 0xFF | | 0x11 | | 0x18 |
| 192 | | 0xEA | | 0xB3 | | 0x14 | | 0x1B |
| 208 | | 0xA5 | | 0xB9 | | 0x5E | | 0x3B |
| 224 | | 0xB7 | | 0x10 | | 0xFF | | 0xAF |
| 240 | | 0xED | | 0x03 | | 0xC9 | | 0xB6 |
| 256 | 0x97E0 | 0xE0 | 0xDC65 | 0x65 | 0x960E | 0x0E | 0xD9FF | 0xFF |
| 272 | | 0x38 | | 0x31 | | 0x2F | | 0xB3 |
| 288 | | 0x92 | | 0xBF | | 0xC1 | | 0x5C |
| 304 | | 0xC7 | | 0x12 | | 0x41 | | 0x9D |
| 320 | | 0x5D | | 0x43 | | 0x1B | | 0x7E |
| 336 | | 0x86 | | 0xE9 | | 0x04 | | 0xA3 |
| 352 | | 0x66 | | 0xB6 | | 0x20 | | 0xC5 |
| 368 | | 0x98 | | 0xF9 | | 0xE7 | | 0x72 |
| 384 | 0xEC2B | 0x2B | 0x32DD | 0xDD | 0x7232 | 0x32 | 0x9AE9 | 0xE9 |
| 400 | | 0xD8 | | 0xB5 | | 0xCF | | 0xED |
| 416 | | 0x80 | | 0xAB | | 0xC7 | | 0x45 |
| 432 | | 0xA5 | | 0xB2 | | 0x27 | | 0x09 |
| 448 | | 0xE5 | | 0x48 | | 0xCA | | 0xAC |
| 464 | | 0xF8 | | 0x6F | | 0xDA | | 0xFC |
| 480 | | 0x8C | | 0xEC | | 0xAE | | 0x32 |
| 496 | | 0xB1 | | 0x68 | | 0x0D | | 0x2D |
| 512 | 0xF82A | 0x2A | 0x5F01 | 0x01 | 0x1D5E | 0x5E | 0x39DA | 0xDA |
| 528 | | 0x15 | | 0x85 | | 0xA4 | | 0x45 |
| 544 | | 0x66 | | 0x07 | | 0x95 | | 0x4E |
| 560 | | 0xF0 | | 0x63 | | 0xC4 | | 0x04 |
| 576 | | 0xF6 | | 0xE5 | | 0xFD | | 0xC1 |
| 592 | | 0x14 | | 0x4A | | 0xF1 | | 0xDD |
| 608 | | 0x8A | | 0x76 | | 0xE0 | | 0xA6 |
| 624 | | 0xC1 | | 0xFA | | 0x65 | | 0xAB |
| 640 | 0x4167 | 0x67 | 0x3618 | 0x18 | 0x8CBA | 0xBA | 0xC0B7 | 0xB7 |

Table A-30. R_i and P_j values (REPEATER = 1)

Appendix B. Confidentiality and Integrity of Values

Table B-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 Table B-1 as requiring integrity.

| Value | Size (Bytes) | Confidentiality Required [†] ? | Integrity Required [†] ? | Function |
|--------------------------------------|-------------------|---|-----------------------------------|---|
| <i>Ainfo</i> | 1 | No | No | Device A information |
| <i>Aksv</i> | 5 | No | No | HDCP Transmitter's Key Selection Vector |
| <i>An</i> | 8 | No | Yes* | Pseudo-random value sent to HDCP Receiver/repeater by transmitter |
| <i>Bksv</i> | 5 | No | Yes* | HDCP Receiver/repeater's Key Selection Vector |
| <i>Bx,By,Bz</i> | 84 bits | Yes | Yes | Cipher state |
| <i>Km,Km'</i> | 7 | Yes | Yes | Secret value generated by HDCP Transmitter and receiver/repeater during authentication |
| <i>Ks,Ks'</i> | 84 bits | Yes | Yes | Secret session key |
| <i>K_b, K_i'</i> | 84 bits | Yes | Yes | Secret frame key |
| <i>Akeys**</i> | 280 | Yes | Yes | HDCP Transmitter's device keys |
| <i>Bkeys**</i> | 280 | Yes | Yes | HDCP Receiver/repeater's device keys |
| <i>LFSR0,1,2,3</i> | 13,14,16, 17 bits | Yes | Yes | Cipher state |
| <i>M_b, M_i'</i> | 8 | Yes | Yes | Integrity verification key and HDCP cipher initialization value |
| <i>P_j</i> | 1 | No | No | Link synchronization verification value computed every 16 th frame from <i>r_i</i> and first pixel. |
| <i>r_b, r_i'</i> | 2 | No | No | HDCP Cipher outputs during frame key calculations. Every 128 th output becomes the video transmitter and receiver link synchronization verification value. Every 16 th output is used in computing <i>P_j</i> . |
| <i>R_b, R_i'</i> | 2 | No | No | Video transmitter and receiver link synchronization verification values |
| SH-0,1,2,3 | 2,2,2,2 bits | Yes | Yes | Cipher state |
| REPEATER | 1 bit | No | Yes | Video repeater capability status bit |

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

* Only within the video transmitter

** KSV position excluded (see *Aksv*, *Bksv*)

| | | | | |
|----------------------|---------|-----|-----|--|
| MAX_CASCADE_EXCEEDED | 1 bit | No | Yes | Video repeater topology error status bit |
| MAX_DEVS_EXCEEDED | 1 bit | No | Yes | Video repeater topology error status bit |
| DEVICE_COUNT | 7 bits | No | Yes | Video repeater topology status bit |
| DEPTH | 3 bits | No | Yes | Video repeater topology status bit |
| V' | 20 | No | No | KSV list integrity value generated by video repeater |
| V | 20 | Yes | Yes | KSV list integrity verification value generated by video transmitter |
| <i>KSV List</i> | Varies | No | Yes | List of downstream KSV gathered by video repeater devices |
| Kx, Ky, Kz | 84 bits | Yes | Yes | Internal HDCP cipher values |
| L^1 | 128 | No | Yes | Digital Content Protection LLC DSS Public Key |
| | | | | |

Table B-1 Confidentiality and Integrity of Values

Appendix C. Sample Algorithm for Ri Verification

The following algorithm or equivalent is suggested as a means to detect a lack of authentication or synchronization.

```
int check()
{
    int T1, T2, R;

    // Read the transmitter, then the receiver, then the
    // transmitter again. If the two transmitter reads agree
    // then we know that there has not been an Ri update, and
    // so the receiver must also agree
    T2 = readTransmitterRi();
    do
    {
        T1 = T2;
        R = readReceiverRi();
        T2 = readTransmitterRi();
    } while (T1 != T2);

    // we have stable values... do they agree?
    return ( T1 == R ) ? LINK_PASS : LINK_FAIL;
}
```

```
int check_synchronization()
{
    int T1, T2, R1, R2, count;

    count = 0;
    T1 = readTransmitterPj();
    R1 = readReceiverPj();
    while (T1 != R1)
    {
        T2 = readTransmitterPj();
        R2 = readReceiverPj();
        // Check for stable Pj values
        if (T1 == T2 && R1 == R2)
        {
            count++; // Count mismatches
            if (count >= 3)
                break;
            T1 = T2;
            R1 = R2;
            // Wait for any of the values to change
            do
            {
                T2 = readTransmitterPj();
                R2 = readReceiverPj();
            } while ((T2 == T1) && (R2 == R1));
        }
        T1 = T2;
        R1 = R2;
    }
    return (count < 3) ? LINK_PASS : LINK_FAIL;
}
```

}

Appendix D. Timing Diagram

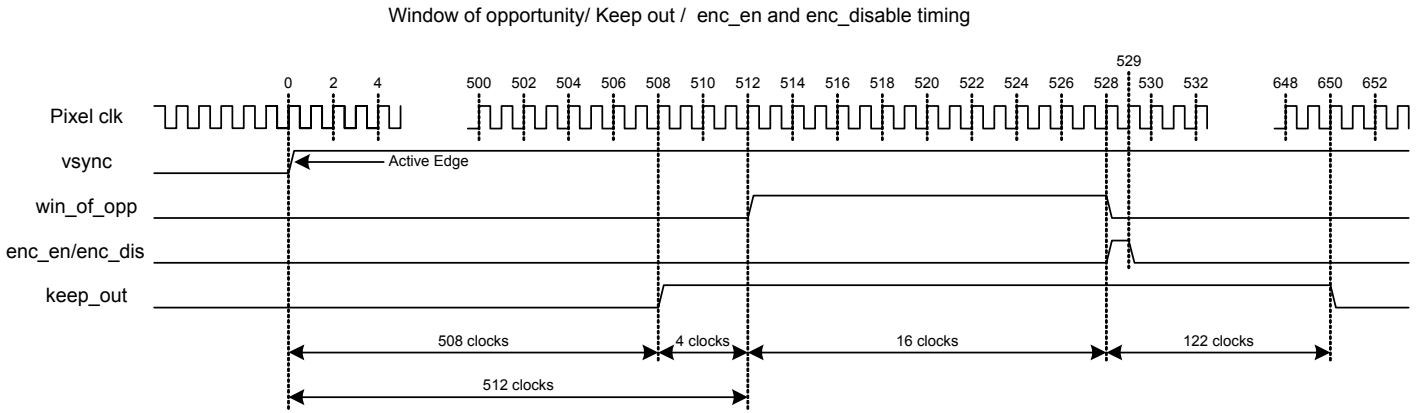


Fig. D-1 Timing

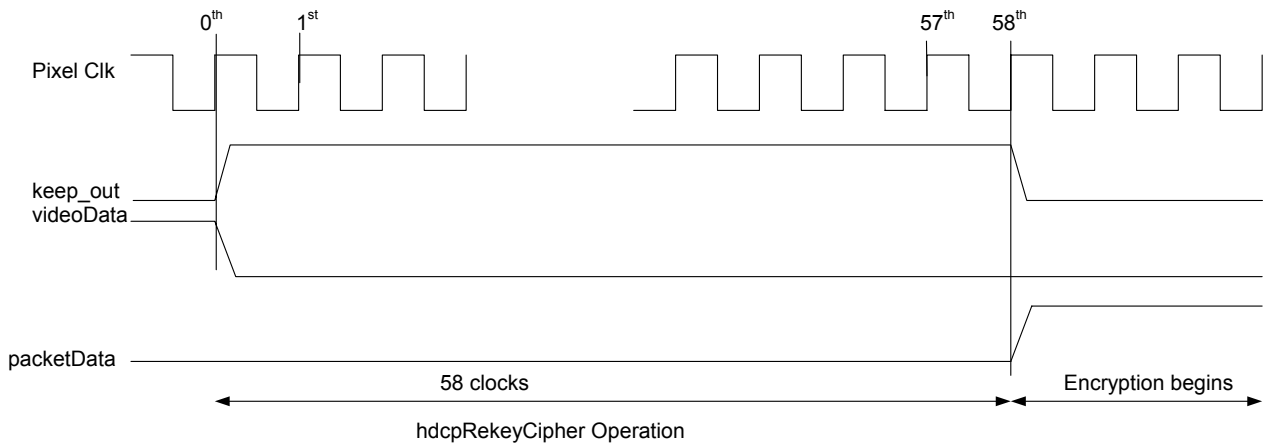


Fig. D-2 hdcpRekeyCipher Timing

Appendix E. Recommended Method for Synchronous Ri Checking

Synchronous reading of R_i every 128th frame does not guarantee reliable detection of link integrity failures. To understand this, consider the following two cases

1. Receiver does not detect CTLx signal (Missing CTLx Signal error)
2. Receiver detects an extra CTLx signal (Extra CTLx Signal error)

| Missing CTLx Signal | | Extra CTLx Signal | |
|----------------------------|-----------------|--------------------------|------------------------|
| <i>Transmitter</i> | <i>Receiver</i> | <i>Transmitter</i> | <i>Receiver</i> |
| $R_i[1]$ | $R_i'[1]$ | $R_i[1]$ | $R_i'[1]$ |
| $R_i[2]$ | $R_i'[2]$ | $R_i[2]$ | $R_i'[2]$ |
| $R_i[3]$ | X (Missed CTLx) | $R_i[3]$ | $R_i'[3]$ |
| $R_i[4]$ | $R_i'[3]$ | X | $R_i'[4]$ (Extra CTLx) |
| - | | $R_i[4]$ | $R_i'[5]$ |
| - | | - | |
| - | | - | |
| $R_i[127]$ | $R_i'[126]$ | $R_i[127]$ | $R_i'[128]$ |
| $R_i[128]$ | $R_i'[127]$ | $R_i[128]$ | $R_i'[129]$ |

In the case of Missing CTLx Signal, the link integrity verification will fail since the R_i and R_i' values differ, however there is no way to verify whether this is due to a loss of synchronization or a DDC read error. In the case of Extra CTLx Signal, loss of synchronization will not be detected by the transmitter since the R_i and R_i' values match.

A reliable method to detect loss of synchronization, therefore, is to read the R_i value prior to the R_i update (127th frame) and after 1 millisecond of the R_i update (128th frame). If the R_i' value is equal to the previously read R_i value, it indicates a Missing CTLx Signal and if the R_i value is equal to a previously read R_i' value it indicates an Extra CTLx Signal.

END OF DOCUMENT