

Superseded by a Subsequent Version of Document

OpenCable™ Specifications

OpenCable System Security Specification

OC-SP-SEC-I06-060413

ISSUED

Notice

This document is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in the document.

Neither CableLabs nor any member company is responsible to any party for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document, or any document referenced herein. This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, or fitness for a particular purpose of this document, or any document referenced herein.

© Copyright 2002-2006 Cable Television Laboratories, Inc.
All rights reserved.

Document Status Sheet

Document Control Number:	OC-SP-SEC-I06-060413			
Document Title:	OpenCable System Security Specification			
Revision History:	W01 – Released June 28, 2002 W02 – Released August 27, 2002 D01 – Released September 27, 2002 I01 – Issued November 26, 2002 I02 – Issued July 7, 2003 I03 – Issued November 21, 2003 I04 – Issued April 2, 2004 I05 – Issued August 31, 2004 I06 – Issued April 13, 2006			
Date:	April 13, 2006			
Status:	Work in Progress	Draft	Issued	Closed
Distribution Restrictions:	Author Only	CL/Member	CL/Member/ Vendor	Public

Key to Document Status Codes:

Work in Progress	An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
Draft	A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
Issued	A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
Closed	A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks:

DOCSIS®, eDOCSIS™, M-CMTS™, PacketCable™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, DCAS™, and CableLabs® are trademarks of Cable Television Laboratories, Inc.

Contents

1	SCOPE	1
1.1	Introduction and Overview	1
1.2	Purpose of Document	1
1.3	Organization of Document	1
1.4	Requirements	1
2	REFERENCES	3
2.1	Normative References	3
2.2	Reference Acquisition	4
3	TERMS AND DEFINITIONS	5
4	ABBREVIATIONS AND ACRONYMS	7
5	CERTIFICATE PROFILE AND MANAGEMENT	8
5.1	Generic Structure	8
5.1.1	Version	8
5.1.2	Public Key Type.....	8
5.1.3	Extensions	8
5.1.4	Signature Algorithm	9
5.1.5	SubjectName and IssuerName.....	9
5.1.6	Certificate Profile Notation.....	9
5.2	Device Certificate Management Architecture Overview	9
5.3	CableLabs Manufacturer Root CA Certificate	11
5.4	CableLabs Device CA Certificate.....	11
5.5	Device Certificate	12
5.6	Certificate Validation.....	14
5.7	Certificate Format.....	15
5.7.1	tbsCertificate.validity.notBefore and tbsCertificate.validity.notAfter	16
5.7.2	tbsCertificate.serialNumber	17
5.7.3	tbsCertificate.signature and signatureAlgorithm.....	17
5.7.4	tbsCertificate.issuer and tbsCertificate.subject.....	17
5.7.5	tbsCertificate.subjectPublicKeyInfo	18
5.7.6	tbsCertificate.issuerUniqueId and tbsCertificate.subjectUniqueId.....	18
5.7.7	signatureValue.....	18
5.8	Host and CableCARD Certificate Storage and Management.....	18
6	OCAP CERTIFICATE PROFILE AND MANAGEMENT	19
6.1	OCAP Manufacturer Code Verification Certificate	19
6.1.1	Common CVC Requirements	19
6.1.2	CableLabs Code Verification Root CA Certificate	20
6.1.3	CableLabs Code Verification CA Certificate	20

6.1.4	Manufacturer Code Verification Certificate	21
6.1.5	Cosigner Code Verification Certificate.....	21
6.1.6	CableLabs Application Code Verification CA Certificate	22
6.1.7	Application Manufacturer Code Verification Certificate	22
6.1.8	Cosigner Application Code Verification Certificate	23
6.1.9	Certificate Revocation Lists for CVCs	23
7	CRYPTOGRAPHIC ALGORITHMS	24
7.1	DES	24
7.2	RSA.....	24
7.3	TDEA.....	24
7.4	AES	24
7.5	Random Number Generation	24
8	PHYSICAL SECURITY	25
8.1	Protection for CableCARD/Host Key and Critical Security Parameter Storage	25
8.2	OpenCable Key Encapsulation	26
8.3	Robustness of CCI and Content Protection Within Card and Host Devices	26
9	SECURE SOFTWARE DOWNLOAD.....	27
9.1	Initialization of Download per DOCSIS Requirements.....	27
9.1.1	CVC Delivery	27
9.1.2	Time Varying Controls	29
9.2	Transport.....	29
9.3	Trigger Mechanism	30
9.3.1	Configuration File Initiated Download.....	30
9.3.2	SNMP Initiated Download.....	31
9.3.3	CVT Initiated Download.....	33
9.4	Image File Structure.....	33
9.4.1	Signed Data.....	33
9.4.2	Signed Content.....	34
9.4.3	Code Signing Keys	35
9.5	Image Validation.....	35
9.6	Error Codes.....	37
APPENDIX I	REVISION HISTORY	39

List of Figures

Figure 1 – CableLabs Device Certificate Hierarchy	10
Figure 2 – OpenCable CVC Certificate Hierarchy.....	19

List of Tables

Table 1 – CableLabs Manufacturer Root CA Certificate	11
Table 2 – CableLabs Device CA Certificate	12
Table 3 – Device Certificate.....	14
Table 4 – X.509 Basic Certificate Fields.....	16
Table 5 – CableLabs Code Verification Root CA Certificate	20
Table 6 – CableLabs Code Verification CA Certificate	20
Table 7 – Manufacturer Code Verification Certificate	21
Table 8 – Cosigner Code Verification Certificate	21
Table 9 – CableLabs Application Code Verification CA Certificate.....	22
Table 10 – Application Manufacturer Code Verification Certificate	22
Table 11 – Cosigner Application Code Verification Certificate.....	23
Table 12 – Code File Structure.....	33
Table 13 – PKCS #7 Signed Data	34

This page left blank intentionally.

1 SCOPE

1.1 Introduction and Overview

In digital Cable systems, high value movies and video programs (“content”) are protected by a conditional access scrambling system. A properly authorized CableCARD security device, previously referred to as a Point of Deployment (POD) module, removes the scrambling and, based on the Content Control Information from the Headend, may rescramble the content before delivering it to consumer receivers and set-top terminals (“Host devices”) across the CableCARD-Host interface defined in SCTE 28 and the CableCARD Copy Protection Specification [18].

This specification defines the system-wide characteristics and normative requirements for the security components that are not included in the CableCARD Copy Protection Specification [18], the CableCARD Interface Specification [16], and the OCAP 1.0 Specification [17].

This specification provides security requirements against unrestricted copying of content, prevention of harm by unauthorized devices, requirements for Public Key Infrastructure (PKI), requirements for Secure Software Download using a DOCSIS® modem (when included in host), and general physical security requirements in the handling of digital certificates and private keys.

1.2 Purpose of Document

This specification provides methods, profiles, and requirements for the Public Key Infrastructure (PKI) used by the cable industry devices for authenticating Host devices, for binding CableCARD Devices to Host devices, including X.509 digital certificate validation, for copy protection key generation, for rescrambling high value content to protect it against unauthorized copying (after the CableCARD Device employs the conditional access system to descramble it), and then descrambling by the Host, and for transmission and authentication of Copy Control Information.

This specification also provides methods, profiles, and requirements for the cable industry Public Key Infrastructure (PKI) used in OCAP software code signing for Secure Software Download and OCAP application code signing to trust these applications in the OCAP operating environment.

1.3 Organization of Document

The overall document is divided into five major sections: OpenCable device certificate profiles and requirements, OCAP certificate profiles and requirements, Cryptographic Algorithms, Physical Security and Secure Software Download.

1.4 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

“SHALL”/ “MUST”	This word or the adjective “REQUIRED” means that the item is an absolute requirement of this specification.
“SHALL NOT”/ “MUST NOT”	This phrase means that the item is an absolute prohibition of this specification.

- “SHOULD” This word or the adjective “RECOMMENDED” means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
- “SHOULD NOT” This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- “MAY” This word or the adjective “OPTIONAL” means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

2 REFERENCES

2.1 Normative References

In order to claim compliance with this specification, it is necessary to conform to the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

- [1] ANSI/SCTE 41 2003 (Formerly DVS 301) POD Copy Protection Standard
- [2] Data-Over-Cable Service Interface Specifications, Baseline Privacy Plus Interface Specification, CM-SP-BPI+-I12-050812, August 12, 2005, Cable Television Laboratories, Inc.
- [3] Data-Over-Cable Service Interface Specifications, Radio Frequency Interface Specification, CM-SP-RFIV2.0-I10-051209, December 9, 2005, Cable Television Laboratories, Inc.
- [4] DFAST encryption technology (including U.S. Patent number 4,860,353 and related know-how) is licensed from CableLabs under the OpenCable POD-Host Interface License Agreement (PHILA) or the DFAST technology and described in CableLabs license materials. This technology is licensed to any interested party under reasonable and non-discriminatory terms. For licensing information, refer to the OpenCable website <www.opencable.com> or contact CableLabs at +1 (303) 661-9100.
- [5] EIA-679-B, National Renewable Security Standard (NRSS), Part B, March 30, 2000
- [6] FIPS PUB 140-2 "Security Requirements for Cryptographic Modules", May 25, 2001
- [7] FIPS PUB 186-2, "Digital Signature Standard" Federal Information Processing Standards Publication (FIPS PUB), January 27, 2000
- [8] FIPS-PUB 180-2, "Secure Hash Standard" Federal Information Processing Standards Publication (FIPS PUB), August 1, 2002
- [9] FIPS-PUB 46-3 "Data Encryption Standard", October 25, 1999
<http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- [10] FIPS-PUB 81 "DES Modes of Operation", December 2, 1980
<http://www.itl.nist.gov/fipspubs/fip81.htm>
- [11] IETF RFC 1750, Randomness Recommendations for Security, (Donald Eastlake, Stephen Crocker and Jeff Schiller), December 1994
- [12] IETF RFC 2104, HMAC: Keyed-Hashing for Message Authentication, (Krawczyk, Bellare, and Canetti), March 1996
- [13] IETF RFC 3369, Cryptographic Message Syntax (CMS), (R. Housley), August 2002
- [14] IETF RFC 3280, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", (R. Housley, W. Ford, W. Polk, D. Solo), January 2002
- [15] ITU-T Recommendation X.509 (03/2000), Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks
- [16] OC-SP-CCIF2.0-I05-060413, OpenCable CableCARD 2.0 Interface Specification, April 13, 2006, Cable Television Laboratories, Inc.
- [17] OC-SP-OCAP1.0-I16-050803, OCAP 1.0 Specification, August 3, 2005, Cable Television Laboratories, Inc.
- [18] OC-SP-CCCP2.0-I03-060413, OpenCable CableCARD Copy Protection Interface Specification, April 13, 2006, Cable Television Laboratories, Inc.
- [19] RSA1, "PKCS #1: RSA Encryption Standard", Version 1.5, RSA Laboratories, November 1993

- [20] OC-SP-HOST2.0-CFR-I08-060413, OpenCable Host Device 2.0 Core Functional Requirements, April 13, 2006, Cable Television Laboratories, Inc.
- [21] FIPS-PUB 197 “Advanced Encryption Standard”, November, 2001
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- [22] DVB-MHP 1.0.3 (ETSI TS 101 812 v1.3.1 May 2005), DVB Multimedia Home Platform 1.03
http://www.etsi.org/services_products/freestandard/home.htm

2.2 Reference Acquisition

CableLabs Specifications:

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027; Phone 303-661-9100; Fax 303-661-9199; Internet: <http://www.cablelabs.com>; www.opencable.com; www.cablemodem.com

EIA Standards:

- Global Engineering Documents, World Headquarters, 15 Inverness Way East, Englewood, CO USA 80112-5776; Phone 800-854-7179; Fax 303-397-2740; Internet <http://global.ihs.com>; email global@ihs.com

Federal Information Processing Standards:

- FIPS Publications, NIST, 100 Bureau Drive, Gaithersburg, MD 20899-3460; Internet: <http://www.itl.nist.gov/fipspubs/>

IETF RFCs:

- Internet Engineering Task Force (IETF) Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, Phone 703-620-8990, Fax 703-620-9071, Internet: www.ietf.org

ITU Standards:

- ITU Sales and Marketing Service, International Telecommunication Union, Place des Nations CH-1211, Geneva 20, Switzerland; Phone +41 22 730 6141; Fax +41 22 730 5194; Internet <http://www.itu.org>; email sales@itu.int

RSA:

- RSA Security Inc., 174 Middlesex Turnpike, Bedford, MA 01730, Tel: 877-RSA-4900 or 781 515 5000, Fax: 781 515 5010, <http://www.rsasecurity.com/rsalabs/pkcs/>

SCTE Standards:

- Society of Cable Telecommunications Engineers Inc., 140 Philips Road, Exton, PA 19341
Phone: 1-800-542-5040, Fax: 610-363-5898 Internet <http://www.scte.org>;
email: standards@scte.org

3 TERMS AND DEFINITIONS

This specification uses the following terms:

CA Certificate	An X.509 version 3 certificate of the CA that issues Device Certificates for device identity authentication. It is issued by the CableLabs Manufacturer Root CA.
CableCARD	A CableCARD device, also referred to as “Point of Deployment” (POD) module, is a detachable device distributed by cable providers that connects to the home receiver. The interface between the CableCARD device and the receiver is specified by the OpenCable platform. CableCARD functionality includes copy protection and signal demodulation.
CableLabs Device CA	An X.509 certificate authority authorized by the CableLabs Manufacturer Root CA to issue Card or Host Certificates.
CableLabs Manufacturer Root CA	The X.509 certificate authority controlled by CableLabs to issue CA Certificates. This authority is also called the PHICA.
CableLabs Security Policy	The highest-level document describing CableLabs security policies.
Card	The same as CableCARD brand removable security module.
Card CA Certificate	The CA Certificate installed in the Card at the factory or during a field code update.
Card Certificate	A Device Certificate issued to a Card. It is also called a POD Certificate or POD Device Certificate.
Card Certificate List	The Card CA Certificate and Card Certificate installed in the Card. This is also called a POD Certificate List.
Certificate	A message that, at least, states a name or identifies the CA, identifies the Subscriber, contains the Subscriber’s public key, identifies the Certificate’s Validity Period, contains a Certificate serial number, and is digitally signed by a CA
Certificate Chain	An ordered list of Certificates containing an end-user Subscriber Certificate and CA Certificates, which terminates in a root Certificate.
Certificate Policies (CP)	The principal statement of policy governing the PKI Hierarchy.
Certificate Revocation List (CRL)	A periodically (or exigently) issued list, digitally signed by a CA, of identified Certificates that have been revoked prior to their expiration dates. The list generally indicates the CRL issuers name, the date of issue, the date of the next scheduled CRL issue, the revoked Certificates serial numbers, and the specific times and reasons for revocation.
Certificate Signing Request	A message conveying a request to have a Certificate issued.
Certification Authority (CA)	An entity authorized to issue, manage, revoke, and renew Certificates in the PKI Hierarchy.
Certification Practice Statement (CPS)	A statement of the practices that CableLabs employs in approving or rejecting Certificate Applications and issuing, managing, and revoking Certificates. In the context of this CPS, “CPS” refers to this document.

Compromise	A violation (or suspected violation) of a security policy, in which an unauthorized disclosure of, or loss of control over, sensitive information may have occurred. With respect to private keys, a Compromise is a loss, theft, disclosure, modification, unauthorized use, or other compromise of the security of such private key.
Controlled Content	Content that has been transmitted from the CableCARD Device with the encryption mode indicator ("EMI") bits set to a value other than zero, zero (0,0).
Device Certificate	An X.509 version 3 certificate used for Card and Host identity authentication. It is issued by a CableLabs CA.
Device Certificate Naming Application	Exhibit C of CableLabs Digital Certificate Authorization Agreement submitted by the Manufacturer.
PKCS	RSA Security Inc. publications titled "Public Key Cryptography Standards."
PKCS #10	Public-Key Cryptography Standard #10, developed by RSA Security Inc., which defines a structure for a Certificate Signing Request.
PKCS #12	Public-Key Cryptography Standard #12, developed by RSA Security Inc., which defines a secure means for the transfer of private keys.
Public Key Infrastructure (PKI)	The architecture, organization, techniques, practices, and procedures that collectively support the implementation and operation of a Certificate-based public key cryptographic system.
Root CA Certificate	A self-signed X.509 version 3 certificate used for device identity authentication. It is maintained by CableLabs. This certificate is also referred to as the CableLabs Manufacturer Root CA Certificate. The Root CA Certificate is installed in both the Card and Host.
RSA	A public key cryptographic system invented by Rivest, Shamir, and Adelman.
Subscriber	In the case of an individual Certificate, a person who is the Subject of, and has been issued, a Certificate. In the case of an organizational Certificate, an organization that owns the equipment or device that is the Subject of, and that has been issued, a Certificate. A Subscriber is capable of using, and is authorized to use, the private key that corresponds to the public key listed in the Certificate.
Uni-Directional Receiving Device (UDRD)	A receiving device compliant with this standard that operates on cable plants that provide signals compliant with ANSI/SCTE 40 2003, and does not transmit data on any return data channel (i.e., unidirectional).
Validity Period	The period starting with the date and time a Certificate is issued (or on a later date and time certain if stated in the Certificate) and ending with the date and time on which the Certificate expires or is earlier revoked.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following abbreviations:

CA	Certification Authority
CHILA	CableCARD-Host Interface Licensing Agreement
CRL	Certificate Revocation List
CVC	Code Verification Certificate
NMS	Network Management System
OID	Object Identifier
PHILA	POD-Host Interface Licensing Agreement
POD	Point of Deployment Module
PKI	Public Key Infrastructure
RDN	Relative Distinguished Name

5 CERTIFICATE PROFILE AND MANAGEMENT

CableCARD Copy Protection SHALL employ X.509 Version 3 certificates for authenticating key exchanges between the CableCARD Device and Host as part of the CableCARD /Host binding process. The following sections define the certificate profile, the certificate contents of defined fields, and the hierarchy of trust for management and validation of CableCARD Copy Protection specification certificates. Except where otherwise noted, the CableCARD and Host certificates for CableCARD Copy Protection must be in compliance with IETF's PKIX standard RFC 3280 [14].

5.1 Generic Structure

5.1.1 Version

The Version of the certificates MUST be V3. All certificates MUST comply with [14] except where the non-compliance with the RFC is explicitly stated in this chapter of this document.

5.1.2 Public Key Type

RSA Public Keys are used throughout the hierarchy. The subjectPublicKeyInfo.algorithm.algorithm Object Identifier (OID) used MUST be 1.2.840.113549.1.1.1 (rsaEncryption).

The public exponent for all RSA keys MUST be $F_4 - 65537$.

5.1.3 Extensions

The following four extensions MUST be used as specified in the sections below. Any other certificate extensions MAY also be included but MUST be marked as non-critical.

5.1.3.1 subjectKeyIdentifier

The subjectKeyIdentifier extension MUST be included in the CableLabs Manufacturer Root CA certificates and the CableLabs Device CA certificates as required by [14]; in the CA Certificate, the extension's keyIdentifier is the 160-bit SHA-1 hash of the CA Certificate's subjectPublicKey BIT STRING value (excluding the tag, length and number of unused bits from the ASN.1 encoding) and the extension's critical flag is set to FALSE (see [14]). The subjectKeyIdentifier extension is not included in the OpenCable Device certificates.

The authorityKeyIdentifier extension MUST be included in all certificates, with the exception of root certificates, and MUST include a keyIdentifier value that is identical to the subjectKeyIdentifier of the CA that issued the certificate. The authorityKeyIdentifier extension's critical flag is set to FALSE.

5.1.3.2 KeyUsage

The KeyUsage extension MUST be used for all certificates. The KeyUsage extension MUST be marked as critical for all certificates. For the CableLabs Manufacturer Root CA certificates and the CableLabs Device CA certificates, the KeyUsage extension MUST have parameters set to keyCertSign = 1, cRLSign = 1 and critical = TRUE. For all device certificates the KeyUsage extension MUST have a value of "digitalSignature and keyEncipherment" when included as specified in [14].

5.1.3.3 BasicConstraints

The basicConstraints extension **MUST** be used for all CA certificates and **MUST** be marked as critical. The values for each certificate for basicConstraints **MUST** be marked as specified in each of the certificate description tables below.

5.1.4 Signature Algorithm

The signature mechanism used **MUST** be SHA-1 with RSA Encryption. The specific OID is 1.2.840.113549.1.1.5.

5.1.5 SubjectName and IssuerName

If a string cannot be encoded as a PrintableString, it **MUST** be encoded as a UTF8String (tag [UNIVERSAL 12]).

When encoding an X.500 Name:

1. Each RelativeDistinguishedName (RDN) **MUST** contain only a single element in the set of X.500 attributes.
2. The order of the RDNs in an X.500 name **MUST** be the same as the order in which they are presented in this specification.

It should be noted that [14] and X.509 defines constraints (i.e., upper bounds) on the length of the attribute values. For example, the maximum length for common name (CN), organization name (O) and organizational unit (OU) name values is 64 characters. Where this specification mandates the inclusion of a static string in one of these values (i.e., CN=CableLabs Device CA), companies **MUST** ensure that the addition of their identifying information does not cause the total length of the value to exceed the upper bound. In the case where a company's name causes the length of the value to exceed the upper bound, the vendor **MUST** truncate or abbreviate their information to ensure the total length does not exceed the upper bound.

5.1.6 Certificate Profile Notation

The tables below use the following notation:

- Extension details are specified by - [c:critical, n:non-critical; m:mandatory, o:optional].
- Optional subject naming attributes are surrounded by square brackets (e.g., [L = <city>]).
- Variable naming attribute values are surrounded by angle brackets. (e.g., CN = <Company> Device). Values not surrounded by angle brackets are static and cannot be modified.

5.2 Device Certificate Management Architecture Overview

The Device Certificate Management Architecture, shown in Figure 1, consists of a three-level hierarchy of trust supporting three types of X.509 Version 3 certificates:

- A single, self-signed CableLabs Manufacturer Root CA certificate
- CableLabs Device CA certificates
- CableCARD and Host certificates

The CableLabs Manufacturer Root Certification Authority serves as the root CA. The root CA issues certificates to the device CAs maintained by or on behalf of CableLabs, which are called the CableLabs Device CAs. The CableLabs Device CAs issue device certificates for CableCARD and Host devices. Note that CableLabs may maintain multiple CableLabs Device CAs to issue Device certificates to different manufacturers, and that multiple CableLabs Device CAs may issue device certificates to a single manufacturer. Distinct CableLabs Device CA issuer names and signing keys SHOULD be maintained for fulfilling certificate requests from each manufacturer.

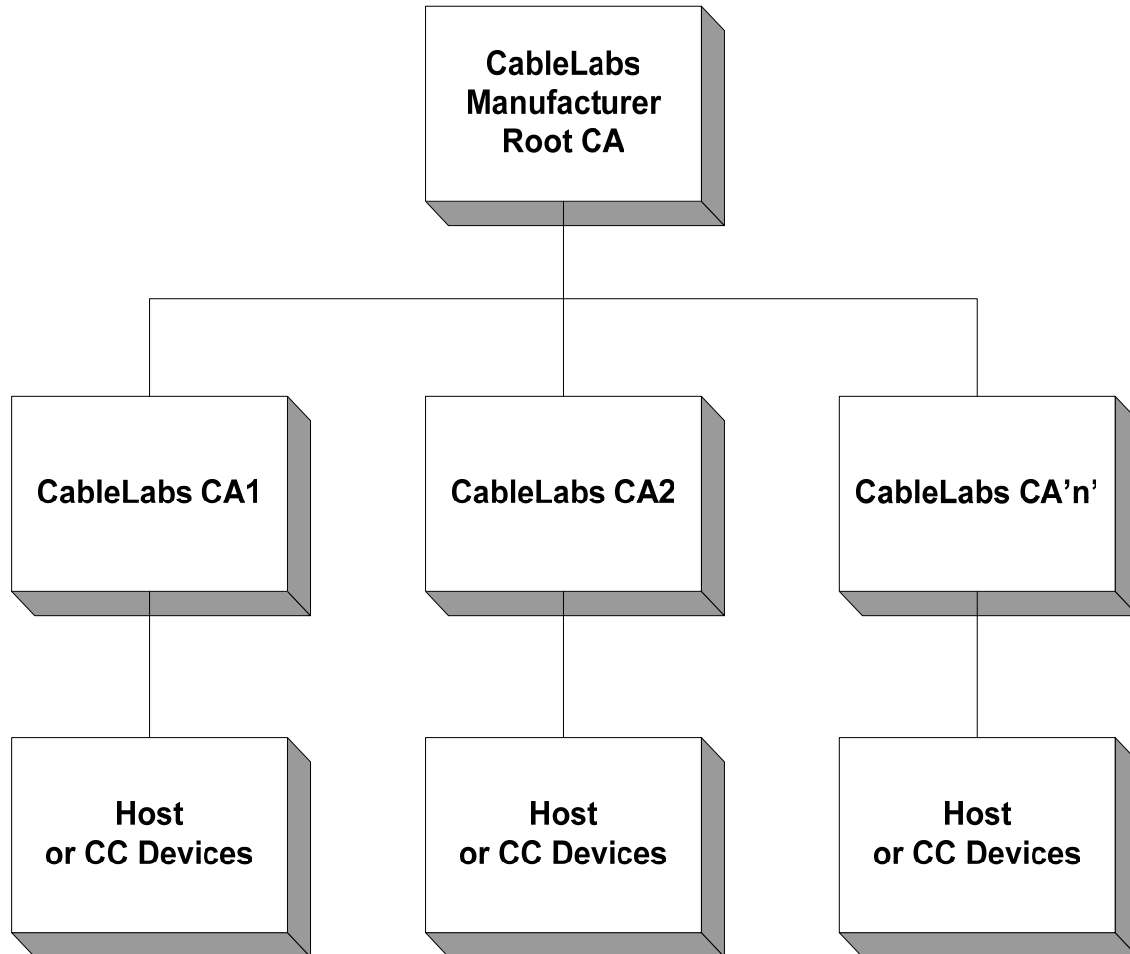


Figure 1 – CableLabs Device Certificate Hierarchy

The CableLabs Manufacturer Root CA shall be kept under tight physical controls. This CA shall be accessed infrequently to issue new CableLabs Device CA certificates. CableLabs shall be responsible for maintaining the CableLabs Manufacturer Root CA and each CableLabs Device CA. The CableLabs Manufacturer Root CA shall be responsible for generating and distributing to MSO's Certificate Revocation Lists (CRLs) identifying revoked CableLabs Device certificates. The manner in which CRLs are distributed to MSOs is outside the scope of the OpenCable System Security Specification.

Manufacturers will be responsible for either generating device RSA key pairs and device certificate requests or generating and providing Host or CableCARD IDs to the Device CA. File formats for requesting and receiving certificates from a CableLabs Device CA are either in PKCS #10 format when key pairs are generated or IDs submitted in files to the Device CA. Protocols for distributing these certificates to receiving CableCARD and Host devices shall be internal to the manufacturer and outside the scope of this specification. Note that care must be taken to ensure that each device is configured with a properly matched private key and certificate.

5.3 CableLabs Manufacturer Root CA Certificate

The CableLabs Manufacturer Root CA Certificate, CableLabs Device CA Certificate and the CableLabs Device Certificate are defined below.

Table 1 – CableLabs Manufacturer Root CA Certificate

Certificate	Certificate Field Description
Subject	C=US O=CableLabs CN=CableLabs Manufacturer Root CA
Validity	30+ years. It is intended that the validity period is long enough that this certificate is never re-issued.
subjectPublicKeyInfo	The certificate's RSA public key (modulus length is 2048 bits)
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=TRUE)

5.4 CableLabs Device CA Certificate

The CableLabs Device CA Certificate MUST be verified as part of a certificate chain containing the CableLabs Manufacturer Root CA Certificate and the CableLabs Device CA Certificate. In short, the CableLabs Device CA certificates MUST be issued and signed by the CableLabs Manufacturer Root CA.

The state/province and city are optional attributes. A manufacturer MAY have more than one manufacturer's CA certificate. If more than one CableLabs Device CA issues certificates for a single manufacturer, the device MUST have access to the appropriate CableLabs Device CA certificate as verified by matching the issuer name in the Device Certificate with the subject name in the CableLabs Device CA Certificate. The authorityKeyIdentifier of the Device Certificate MUST be matched to the subjectKeyIdentifier of the CableLabs Device CA certificate as described in [14].

The following checks SHALL be performed at the time of issuance of the CA certificate by the Root CA:

Validation of the keyUsage extension in the CA Certificate is accomplished by checking that:

- The extension is present
- The extension's keyCertSign parameter is set to 1
- The extension's cRLSign parameter is set to 1
- The extension's critical flag is set to TRUE

Validation of the authorityKeyIdentifier extension in the Device CA Certificate is accomplished by checking that:

- The extension is present
- The extension's keyIdentifier value is identical to the CA Certificate's subjectKeyIdentifier value
- The extension's critical flag is set to FALSE

Validation of the basicConstraints extension in the Device CA Certificate is accomplished by checking that:

- The extension is present
- The extension's cA parameter is set to TRUE
- The extension's pathLenConstraint parameter is set to 0
- The extension's critical flag is set to TRUE

Table 2 – CableLabs Device CA Certificate¹

Certificate Field	Certificate Field Description
Subject	C=US O=CableLabs, Inc. S=Colorado L=Louisville OU=<CA Designator> CN= <Common Name>
Validity	Up to 30 years
subjectPublicKeyInfo	The certificate's RSA public key (modulus length is 2048 bits)
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] authorityKeyIdentifier[n,m](keyIdentifier=<subjectKeyIdentifier value from CA certificate>) basicConstraints[c,m](cA=TRUE, pathLenConstraint=0) subjectAltName[n,o] (Directory Address)

5.5 Device Certificate

The Device Certificate MUST be verified as part of a certificate chain containing the CableLabs Manufacturer Root CA Certificate, CableLabs Device CA Certificate and the Device Certificate.

The manufacturer's facility is an optional attribute. The product name is an optional attribute. When present, the Product Name field SHALL reasonably allow CableLabs to identify Products by model. Manufacturers will cooperate with CableLabs in defining and using numbering systems that will permit such ready identification.

The device POD_ID and Host_ID MUST be expressed as hexadecimal digits in the ASCII representation of the Device Certificate. The Alpha HEX characters (A-F) MUST be expressed as uppercase letters.

The POD_ID defined in the CN field of the X.509 certificate MUST comply with the following requirements:

- The 24 MSBs of the POD_ID are reserved. CableCARD manufacturers shall set them to zero.
- CableLabs shall assign a 3 decimal digit CableCARD manufacturer number upon request by any PHILA or CHILA signatory who has successfully completed CableCARD qualification

¹ This certificate is issued to each CableLabs Device CA by the CableLabs Manufacturer Root CA and can be provided to each device either at manufacture time, or during a field code update. This certificate appears as a read-only parameter in the device. This certificate, along with the CableLabs Manufacturer Root CA Certificate and the Device Certificate, is used to authenticate the device identity. This certificate is signed by the CableLabs Manufacturer Root CA.

- CableCARD manufacturers shall set the 10 most significant bits of the remaining unreserved 40-bits of the POD_ID to the binary equivalent of their assigned CableCARD manufacturer number.
- CableCARD manufacturers shall limit the remaining 30-bits of the POD_ID to a value less than one billion, i.e., no value higher than binary 11,1011,1001,1010,1100,1001,1111,1111, 3B9AC9FF hexadecimal, to facilitate on-screen presentation to subscribers and manual report back.
- Any manufacturer who has credibly deployed products using more than 25% of an assigned ID range may request additional manufacturer numbers.
- The POD_ID (30-bits) assigned to each CableCARD device certificate MUST be unique to each CableCARD device.

The Host_ID defined in the CN field of the X.509 certificate MUST comply with the following requirements:

- CableLabs shall assign a 3 decimal digit Host manufacturer number upon request by any PHILA, CHILA, or DFAST signatory who has successfully completed Host certification.
- The 10 most significant bits of the Host ID SHALL be set equal to the binary equivalent of the Host manufacturer number.
- Host manufacturers shall set the 10 most significant bits of the 40-bit Host_ID to the binary equivalent of their assigned Host manufacturer number.
- Host manufacturers shall set the remaining 30-bits of the Host_ID to a value between zero and 999,999,999 decimal, 3B9AC9FF hexadecimal, to facilitate on-screen presentation to subscribers and manual report back.
- Any manufacturer who has credibly deployed products using more than 25% of an assigned ID range may request additional manufacturer numbers.
- The Host_ID (40-bits) assigned to each Host device certificate MUST be unique to each Host device.

A Device Certificate is permanently installed. Therefore, the Device Certificate MUST have a validity period greater than the operational lifetime of the specific device.

At the time of issuance of the Device Certificates, the Device CA MUST check that all certificates comply with the Certificate Field Descriptions described in Table 4. It is especially important to check the following:

- CN=<POD ID or HOST ID> is unique
- Signature Algorithm used is SHA-1 with RSA encryption
- RSA public key modulus length is 1024 bits
- Validity is not greater than 30 years
- C, and O fields are correctly populated from the Manufacturer supplied Device Certificate Naming Application
- keyUsage extension is present, and the critical flag set to TRUE
- keyUsage extension's digitalSignature, and keyEncipherment parameters (and nothing else) are set to 1
- authorityKeyIdentifier is present, and the critical flag is NOT set to TRUE
- authorityKeyIdentifier's keyIdentifier value is set to the subjectKeyIdentifier value of the CA Certificate

Table 3 – Device Certificate²

Certificate	Certificate Field Description
Subject	C=<country> O=<Company Name> [S=<state/province>] [L=<city>] OU=OpenCable [OU=<Product Name>] [OU=<Manufacturer's Facility>] CN=<POD ID or Host ID> ³ [OU=MFG ID] ⁴
Validity	Up to 30 years
subjectPublicKeyInfo	The certificate's RSA public key (modulus length is 1024 bits)
Extensions	keyUsage[c,m](digitalSignature, keyEncipherment), authorityKeyIdentifier [n, m](keyIdentifier=<subjectKeyIdentifier value from CA Certificate>)

5.6 Certificate Validation

Certificate validation involves validation of a linked chain of certificates from the end entity certificates to the valid Root. For example, the signature on the Device Certificate is verified with the CableLabs Device CA Certificate and then the signature on the CableLabs Device CA Certificate is verified with the CableLabs Manufacturer Root CA Certificate. The CableLabs Manufacturer Root CA Certificate is self-signed and this certificate is received from a trusted source in a secure way. The public key present in the CableLabs Manufacturer Root CA Certificate is used to validate the signature on this same certificate.

The exact rules for certificate validation MUST comply with the certificate validation requirements in this section and with RFC 3280 [14], where they are referred to as “Certificate Path Validation.” Note that because the certificate revocation check described in section 6.1.3 (a) (3) of RFC 3280 [14] cannot be performed, it is omitted during the certificate path validation process. The certificate revocation check will be performed as part of the Authentication Phase 2 described in the OpenCable CableCARD Copy Protection Interface Specification [18].

In general, X.509 certificates support an open set of rules for determining if the issuer name of a certificate matches the subject name of another. The rules are such that two name fields may be declared to match even though a binary comparison of the two name fields does not indicate a match. RFC 3280 [14] recommends that certificate authorities restrict the encoding of name fields so that an implementation can declare a match or mismatch using simple binary comparison. This security specification follows the encoding recommendation. Accordingly, the DER-encoded `tbsCertificate.issuer` field of a certificate MUST be an exact match to the DER-encoded `tbsCertificate.subject` field of its issuer certificate. An implementation MAY compare an issuer name to a subject name by performing a binary comparison of the DER-encoded `tbsCertificate.issuer` and `tbsCertificate.subject` fields.

² This certificate is issued by a CableLabs Device CA and installed in the device at the factory. The head-end cannot update this certificate. This certificate appears as a read-only parameter in the device.

³ There are 40 bits for the Host ID, which are represented as 10 ASCII Hex bytes in the CN field, and there are 64 bits for the POD ID, which are represented as 16 ASCII Hex bytes in the CN field.

⁴ “MFG ID” is the 3 decimal digit Device manufacturer number assigned by CableLabs upon request by any PHILA or DFAST signatory who has successfully completed Host certification.

The validation of validity periods for nesting is not checked and intentionally not enforced, which is compliant with current standards. At the time of issuance, the validity start date for any end-entity certificate MUST be the same as or later than the start date of the issuing CA certificate validity period and fall within the validity period of the issuing CA certificate. The validity end date for entities may be before, the same as or after the validity end date for the issuing CA as specified in the Certificate tables.

Validity period checking is accomplished by checking that:

- The certificate validity period includes the current time.

Validation of the Common Name (CN) in the Certificate received for binding is accomplished by checking that:

- The CN is present in the subject Distinguished Name of the received certificate.
- If the receiving device is a Host then the length of the CN in the received certificate SHALL be 16 ASCII characters.
- If the receiving device is a CableCARD then the length of the CN in the received certificate SHALL be 10 ASCII characters.

Validation of the keyUsage extension in the Device Certificate is accomplished by checking that:

- The extension is present
- The extension's digitalSignature parameter is set to 1
- The extension's keyEncipherment parameter is set to 1

Validation of the authorityKeyIdentifier extension in the Device Certificate is accomplished by checking that:

- The extension is present
- The extension's keyIdentifier value is identical to the Device CA Certificate's subjectKeyIdentifier

A certificate SHALL NOT be rejected as invalid solely on the basis of missing optional fields.

5.7 Certificate Format

This section describes the X.509 version 3 certificate format and certificate extensions used in CableCARD Copy Protection Specification. The following table summarizes the basic fields of an X.509 Version 3 certificate.

Table 4 – X.509 Basic Certificate Fields

X.509 v3 Field	Descriptions
tbsCertificate.version	Indicates the X.509 certificate version. Always set to version 3 (value of 2).
tbsCertificate.SerialNumber	Unique integer value that the CA assigns to the certificate.
tbsCertificate.Signature	OID and option parameters defining the algorithm used to sign the certificate. This value must contain the same algorithm identifier as the signatureAlgorithm field defined below.
tbsCertificate.issuer	Distinguished name of the CA issuing the certificate.
tbsCertificate.validity	Specifies when the certificate becomes valid and when it expires.
tbsCertificate.subject	Distinguished Name identifying the entity whose public key is certified in the subjectPublicKeyInfo field.
tbsCertificate.subjectPublicKeyInfo	This field contains public key material (public key and parameters) and the identifier of the algorithm with which the key is used.
tbsCertificate.issuerUniqueIDs	Optional field to allow the reuse of Issuer names over time.
tbsCertificate.subjectUniqueIDs	Optional field to allow the reuse of Subject names over time.
tbsCertificate.extensions	The certificate extension data.
tbsCertificate.signatureAlgorithm	OID and option parameters defining the algorithm used to sign the certificate. This value must contain the same algorithm identifier as the Signature field defined above.
tbsCertificate.signatureValue	Digital Signature computed upon the ASN.1 DER encoded tbsCertificate.

All certificates and CRLs described in this specification **MUST** be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1 [19]; SHA-1 is described in [8]. This is just one example of how CableCARD Copy Protection Specification restricts the values of the X.509 Certificates basic fields. All of these restrictions are described below.

5.7.1 tbsCertificate.validity.notBefore and tbsCertificate.validity.notAfter

Host and CableCARD device certificates will not be renewable, and, thus, must have a validity period greater than the operational lifetime of the Host and CableCARD devices. A CableLabs Device CA certificate **MUST** be valid from the issuance date for a period of 20 years and re-issued in a period of 60 days. The CableLabs Manufacturer Root CA certificate **MUST** be valid from the date when the CableLabs Manufacturer Root CA starts operating for a period of 30 years.

This specification assumes the operational lifetime of a Host and CableCARD device will not exceed 20 years. The validity periods of a Host device and CableCARD device certificate **MUST** begin with the device's date of manufacture; the validity period **SHOULD** extend out to at least 20 years after that manufacturing date.

Validity periods **MUST** be encoded as UTCTime. UTCTime values **MUST** be expressed Greenwich Mean Time (Zulu) and **MUST** include seconds (i.e., times are YYMMDDHHMMSSZ), even where the number of seconds is zero. The year field (YY) **MUST** be interpreted as 20YY.

5.7.2 tbsCertificate.serialNumber

The serial number **MUST** be a positive integer assigned by the CA to each certificate. It **MUST** be unique for each certificate issued by a given CA (i.e., the issuer name and serial number identify a unique certificate). CAs **MUST** force the serialNumber to be a non-negative integer. The Manufacturer **SHOULD NOT** impose or assume a relationship between the serial number of the certificate and the serial number of the Host or CableCARD device to which the certificate is issued.

Given the uniqueness requirements above, serial numbers can be expected to contain long integers. Certificate users **MUST** be able to handle serialNumber values up to 20 octets. Conformant CAs **MUST NOT** use serialNumber values longer than 20 octets.

5.7.3 tbsCertificate.signature and signatureAlgorithm

All certificates and CRLs described in this specification **MUST** be signed with the RSA signature algorithm, using SHA-1 as the one-way hash function. The RSA signature algorithm is described in PKCS #1 [19]; SHA-1 is described in [8].

The ASN.1 OID used to identify the “SHA-1 with RSA” signature algorithm is:

```
sha-1WithRSAEncryption OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-1(1) 5}
```

When the sha-1WithRSAEncryption OID appears within the ASN.1 type AlgorithmIdentifier, as is the case with both tbsCertificate.signature and signatureAlgorithm, the parameters component of that type is the ASN.1 type NULL.

5.7.4 tbsCertificate.issuer and tbsCertificate.subject

X.509 Names are SEQUENCES of RelativeDistinguishedNames, which are in turn SETs of AttributeTypeAndValue. AttributeTypeAndValue is a SEQUENCE of an AttributeType (an OBJECT IDENTIFIER) and an AttributeValue. The value of the countryName attribute **MUST** be a 2-character PrintableString, chosen from ISO 3166; all other AttributeValues **MUST** be encoded as either UTF8String or PrintableString character strings. The PrintableString encoding **MUST** be used if the character string contains only characters from the PrintableString set. Specifically:

```
abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789
'()+,-./:=? and space.
```

The UTF8String **MUST** be used if the character string contains other characters.

The following OIDs are needed for defining issuer and subject Names in HOST AND CableCARD certificates:

```
id-at OBJECT IDENTIFIER ::= {joint-iso-ccitt(2) ds(5) 4}
id-at-commonName          OBJECT IDENTIFIER ::= {id-at 3}
id-at-countryName         OBJECT IDENTIFIER ::= {id-at 6}
id-at-localityName        OBJECT IDENTIFIER ::= {id-at 7}
id-at-stateOrProvinceName OBJECT IDENTIFIER ::= {id-at 8}
id-at-organizationName    OBJECT IDENTIFIER ::= {id-at 10}
id-at-organizationalUnitName OBJECT IDENTIFIER ::= {id-at 11}
```

The following subsections describe the format of the subject name field for each type of HOST AND CableCARD certificate. The issuer name field of a certificate matches the subject name field of the issuing certificate. Any certificate transmitted by a Host or CableCARD Device message **MUST** have name fields

that conform to the indicated format. A Host and CableCARD Device MUST be capable of processing the name fields of a certificate if the name fields conform to the indicated format.

5.7.5 tbsCertificate.subjectPublicKeyInfo

The tbsCertificate.subjectPublicKeyInfo field contains the public key and the public key algorithm identifier.

The tbsCertificate.subjectPublicKeyInfo.algorithm field is an AlgorithmIdentifier structure. The AlgorithmIdentifiers algorithm MUST be RSA encryption, identified by the following OID:

```
pkcs-1 OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
  rsadsi(113549) pkcs(1) 1}
rsaEncryption OBJECT IDENTIFIER ::= { pkcs-1 1}
```

The AlgorithmIdentifiers parameters field MUST have ASN.1 type NULL.

The RSA public key shall be encoded using the ASN.1 type RSAPublicKey :

```
RSAPublicKey ::= SEQUENCE {
  modulus                INTEGER, -- n
  publicExponent         INTEGER, -- e -- }
```

where modulus is the modulus n, and publicExponent is the public exponent e. The DER encoded RSAPublicKey is the value of the BIT STRING tbsCertificate.subjectPublicKeyInfo.subjectPublicKey.

5.7.6 tbsCertificate.issuerUniqueID and tbsCertificate.subjectUniqueID

The issuerUniqueID and subjectUniqueID fields MUST be omitted for both types of host or CableCARD certificate types.

5.7.7 signatureValue

In both Host and CableCARD certificate types, the signatureValue contains the RSA (with SHA-1) signature computed over the ASN.1 DER encoded tbsCertificate. The ASN.1 DER encoded tbsCertificate is used as input to the RSA signature function. The resulting signature value is ASN.1 encoded as a BIT STRING and included in the Certificates signatureValue field.

5.8 Host and CableCARD Certificate Storage and Management

CableLabs Device CA-issued Host or CableCARD certificates MUST be stored in non-volatile memory. Host and CableCARD devices that have factory-installed RSA private/public key pairs MUST also have factory-installed device certificates. Host and CableCARD devices that rely on internal algorithms to generate an RSA key pair MUST support a mechanism for installing a CableLabs Device CA-issued device certificate following key generation. Note: The private key is only present in the Host or CableCARD device for the Host or CableCARD device certificate. Private keys for the CableLabs Device CA certificate and the Manufacturer Root CA certificate are not present in the end-entity device.

The CableLabs Device CA certificate that signed and issued the Host or CableCARD device certificate MUST be stored in the cable device's non-volatile memory. The device MUST be capable of updating or replacing the CableLabs Device CA certificate via the OpenCable code download file. The CableLabs Device CA certificate MAY be embedded into the Host or CableCARD software.

The CableLabs Manufacturer Root CA certificate MUST be loaded into both CableCARD and Host devices at manufacture time and MUST be protected according to FIPS 140-2 level 1 [6] requirements. The specific CableLabs Device CA certificate installed by the Host or CableCARD device will be that identifying the issuer of that device certificate.

6 OCAP CERTIFICATE PROFILE AND MANAGEMENT

OCAP [17] Security shall employ X.509 Version 3 certificates for authenticating software download of OCAP implementations and OCAP applications. The following sections define the certificate profile, the certificate contents of defined fields, and the hierarchy of trust for management and validation of OCAP certificates. Except where otherwise noted, the OCAP certificates for Secure Software Download must be in compliance with IETF's PKIX standards RFC 3280 [14].

6.1 OCAP Manufacturer Code Verification Certificate

Figure 2 illustrates the CableLabs Code Verification Certificate (CVC) PKI. This PKI is generic in nature and applicable to all CableLabs projects needing CVCs. This means the basic infrastructure can be re-used for every CableLabs project. There may be differences in the end-entity certificates required for each project, but in the cases where end-entity certificates overlap, one end-entity certificate could be used to support the overlap.

The CableLabs CVC hierarchy only applies to OCAP implementations and OCAP applications.

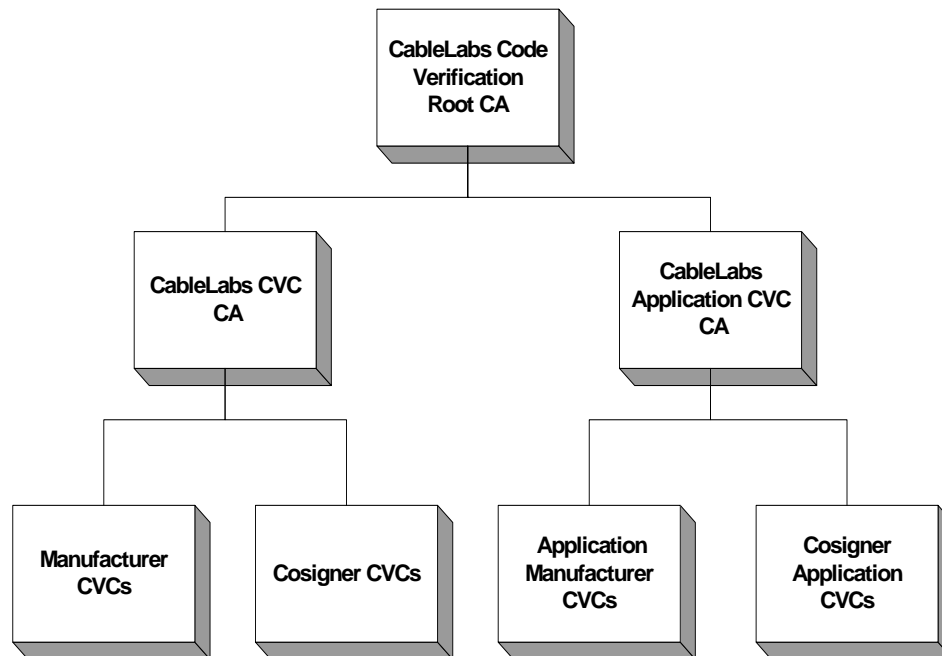


Figure 2 – OpenCable CVC Certificate Hierarchy

6.1.1 Common CVC Requirements

The following requirements apply to all Code Verification Certificates:

- Certificates **MUST** be DER encoded
- Certificates **MUST** be version 3
- Certificates **MUST** include the extensions that are specified in the following tables and **MUST NOT** include any additional extensions
- The public exponent **MUST** be F_4 (65537 decimal).

6.1.2 CableLabs Code Verification Root CA Certificate

This CVC certificate chain MUST be verified as containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA, and the Code Verification Certificates.

Table 5 – CableLabs Code Verification Root CA Certificate

CableLabs Code Verification Root CA Certificate	
Subject Name Form	C=US O=CableLabs CN=CableLabs CVC Root CA
Intended Usage	This certificate is used to sign Code Verification CA Certificates. This certificate MUST be included in the OPENCABLE HOST's non-volatile memory at manufacture time.
Signed By	Self-signed
Validity Period	30+ years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectkeyidentifier [n,m] basicConstraints [c,m](cA=true)

6.1.3 CableLabs Code Verification CA Certificate

The CableLabs Code Verification CA Certificate MUST be verified as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, CableLabs Code Verification CA Certificate and the Code Verification Certificate. There MAY be more than one CableLabs Code Verification CA. An OpenCable Host MUST support one CableLabs CVC CA at a time.

Table 6 – CableLabs Code Verification CA Certificate

CableLabs Code Verification CA Certificate	
Subject Name Form	C=US O=CableLabs CN=CableLabs CVC CA
Intended Usage	This CA certificate is issued to CableLabs by the CableLabs Code Verification Root CA. This CA issues Code Verification Certificates. This CA certificate MUST be included in the OpenCable Host's non-volatile memory at manufacture time.
Signed By	CableLabs Code Verification Root CA
Validity Period	20 years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectKeyIdentifier [n,m] authorityKeyIdentifier [n,m] basicConstraints [c,m](cA=true, pathLenConstraint=0) subjectAltName[n,o] (Directory Address)

6.1.4 Manufacturer Code Verification Certificate

This certificate **MUST** be verified as part of the certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA Certificate, and the Manufacturer Code Verification Certificate.

Table 7 – Manufacturer Code Verification Certificate

Manufacturer Code Verification Certificate	
Subject Name Form	C=<country> O=<Company Name> [S=<state/province>] [L=<city>] CN=<Company Name> Mfg CVC
Intended Usage	The CableLabs Code Verification CA issues this certificate to each authorized Manufacturer. It is used in the policy set by the cable operator for secure software download.
Signed By	CableLabs Code Verification CA
Validity Period	10 years
Modulus Length	1024, 1536, 2048
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m] keyUsage[c,m](digitalSignature, keyEncipherment)

The Company Name in the Organization **MAY** be different than the Company Name in the Common Name.

6.1.5 Cosigner Code Verification Certificate

The Cosigner Code Verification Certificate **MUST** be verified as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA Certificate, and the Cosigner Code Verification Certificate.

Table 8 – Cosigner Code Verification Certificate

Cosigner Code Verification Certificate	
Subject Name Form	C=US O=CableLabs/MSO/Eight character hexadecimal value CN=CableLabs/MSO CVC
Intended Usage	The CableLabs Code Verification CA issues this certificate. It is used to authenticate CableLabs or MSO certified code. It is used in the policy set by the cable operator for secure software download and is used in the cosigning of software implementations.
Signed By	CableLabs Code Verification CA
Validity Period	10 years
Modulus Length	1024
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m] keyUsage[c,m](digitalSignature, keyEncipherment)

6.1.6 CableLabs Application Code Verification CA Certificate

The CableLabs Application Code Verification CA Certificate MUST be verified as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, CableLabs Application Code Verification CA Certificate and the Application Code Verification Certificate. There MAY be more than one CableLabs Application Code Verification CA. An OpenCable Host MUST support one CableLabs CVC CA at a time.

Table 9 – CableLabs Application Code Verification CA Certificate

CableLabs Application Code Verification CA Certificate	
Subject Name Form	C=US O=CableLabs CN=CableLabs Application CVC CA
Intended Usage	This CA certificate is issued to CableLabs by the CableLabs Code Verification Root CA. This CA issues Application Code Verification Certificates. This CA certificate MUST be included in the OpenCable Host's non-volatile memory at manufacture time.
Signed By	CableLabs Code Verification Root CA
Validity Period	20 years
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectKeyIdentifier [n,m] authorityKeyIdentifier [n,m] basicConstraints [c,m](cA=true, pathLenConstraint=0)

6.1.7 Application Manufacturer Code Verification Certificate

This certificate MUST be verified as part of the certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Application Code Verification CA Certificate, and the Application Manufacturer Code Verification Certificate.

Table 10 – Application Manufacturer Code Verification Certificate

Application Manufacturer Code Verification Certificate	
Subject Name Form	C=<country> O=<Company Name>.<organization_ID> [ST=<state/province>] [L=<city>] CN=<Company Name> Application Mfg CVC
Intended Usage	The CableLabs Application Code Verification CA issues this certificate to each authorized application Manufacturer. It is used in the policy set by the cable operator for software applications.
Signed By	CableLabs Application Code Verification CA
Validity Period	10 years
Modulus Length	1024, 1536, 2048
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) subjectKeyIdentifier [n,m] authorityKeyIdentifier [n,m]

The Company Name in the Organization MAY be different than the Company Name in the Common Name. The “organization_ID” is issued by the Digital Video Broadcasting (DVB) consortium as specified in [22].

6.1.8 Cosigner Application Code Verification Certificate

The Cosigner Application Code Verification Certificate MUST be verified as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Application Code Verification CA Certificate, and the Cosigner Application Code Verification Certificate.

Table 11 – Cosigner Application Code Verification Certificate

Cosigner Application Code Verification Certificate	
Subject Name Form	C=US O=CableLabs/MSO.<organization_ID> CN=CableLabs/MSO Application Cosigner CVC
Intended Usage	The CableLabs Application Code Verification CA issues this certificate. It is used to authenticate OCAP application code. It is used in the policy set by the cable operator for software application cosigning.
Signed By	CableLabs Application Code Verification CA
Validity Period	10 years
Modulus Length	1024, 1536, 2048
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) subjectKeyIdentifier [n,m] authorityKeyIdentifier [n,m]

The “organization_ID” is issued by the Digital Video Broadcasting (DVB) consortium.

6.1.9 Certificate Revocation Lists for CVCs

The OpenCable Host is not required to support Certificate Revocation Lists (CRLs) for CVCs.

7 CRYPTOGRAPHIC ALGORITHMS

This section describes the cryptographic algorithms used in this OpenCable System Security specification. When a particular algorithm is used, the algorithm **MUST** follow the corresponding specification.

7.1 DES

The Data Encryption Standard (DES) is specified in [9]. For Media Stream encryption, OpenCable does not require error checking on the DES key.

7.2 RSA

All public key signatures for OpenCable **MUST** be generated and verified using the RSA signature algorithm described in [8]. The format for all OpenCable signatures **MUST** be compliant with the Cryptographic Message Syntax [13].

7.3 TDEA

The Triple Data Encryption Algorithm (TDEA) is specified in [9]. For Media Stream encryption, OpenCable does not require error checking on the TDEA key.

7.4 AES

The Advanced Encryption Standard (AES) is specified in [21].

7.5 Random Number Generation

Good random number generation is vital to most cryptographic mechanisms. Implementations **SHOULD** do their best to produce true-random seeds; they should also use cryptographically strong pseudo-random number generation algorithms. RFC 1750 (See [11]) gives some suggestions; other possibilities include use of a per device secret installed at manufacture time and used in the random number generation process.

8 PHYSICAL SECURITY

8.1 Protection for CableCARD/Host Key and Critical Security Parameter Storage

The OpenCable System Security Specification requires that a CableCARD and Host device maintain persistent encryption and authentication keys and derived shared secret keys. Secret keys, private keys, and CSPs SHALL be protected within the cryptographic module from unauthorized disclosure, modification, and substitution. Public keys (certificates) SHALL be protected within the cryptographic module against unauthorized modification and substitution. A device SHOULD deter unauthorized physical access to this key material and Critical Security Parameters (CSPs).

The level of physical protection of key material required by the OpenCable System Security Specification for a CableCARD and Host device is specified in terms of the security level one as defined in the FIPS PUBS 140-2 [6], Security Requirements for Cryptographic Modules, standard. A POD or Host MUST, at a minimum, meet FIPS PUBS 140-2 Security Level 1 requirements for sections 4.5 (Physical Security), 4.6 (Operating Environment), and 4.7 (Cryptographic Key Management). (Note: OpenCable devices will be required to meet these FIPS 140-2 Level 1 requirements but will not require FIPS 140 formal certification.)

The level of physical protection of key material required by the OpenCable System Security Specification for an OpenCable Host Device 2.0 is specified in terms of the security level two as defined in the FIPS PUBS 140-2 [6], Security Requirements for Cryptographic Modules, standard. A Host 2.0 MUST, at a minimum, meet FIPS PUBS 140-2 Security Level 2 requirements for sections 4.5 (Physical Security), 4.6 (Operating Environment), and 4.7 (Cryptographic Key Management). (Note: OpenCable devices will be required to meet these FIPS 140-2 Level 2 requirements but will not require FIPS 140 formal certification.)

A device MUST also maintain in protected non-volatile memory another set of parameters called Critical Security Parameters (CSPs). The critical security parameters for the OpenCable project are defined as:

- DFAST constants
- Diffie Hellman Prime value
- Diffie Hellman Base value

The level of physical protection of Critical Security Parameters (CSPs) required by the OpenCable System Security Specification for a CableCARD and Host device is specified in terms of the security levels defined in the FIPS PUBS 140-2 [6], Security Requirements for Cryptographic Modules, standard. A CableCARD or Host MUST, at a minimum, meet FIPS PUBS 140-2 Security Level 1 requirements.

The level of physical protection of Critical Security Parameters (CSPs) required by the OpenCable System Security Specification for a OpenCable Host Device 2.0 is specified in terms of the security levels defined in the FIPS PUBS 140-2 [6], Security Requirements for Cryptographic Modules, standard. A Host 2.0 MUST, at a minimum, meet FIPS PUBS 140-2 Security Level 2 requirements.

The OpenCable System Security Specification requirements protect against unauthorized access to these network services by enforcing an end-to-end message integrity and encryption of signaling flows across the network and by employing an authenticated key management protocol. If an attacker is able to legitimately subscribe to a set of services and also gain physical access to a device containing key material and/or Critical Security Parameters (CSPs), then in the absence of strong physical protection of this information, the attacker can extract key material from the POD or Host and redistribute the keys to other users running modified illegitimate CableCARD and Host devices, effectively allowing the theft of copy protected content and possible cloning of devices.

8.2 OpenCable Key Encapsulation

As stated in the previous section, FIPS PUB 140-2 Security Levels specify that a CableCARD, Host, and Host 2.0 device **MUST** incorporate physical security to deter unauthorized “physical” access to its key material. This restricted access also includes any ability to directly read the key material using any of the CableCARD and Host interfaces.

It is strongly recommended that any persistent key material **SHOULD** be encapsulated such that there is no way to extract the key material from the CableCARD, Host, or Host 2.0 device using any of that device's interfaces without modifications to the Host and CableCARD device.

8.3 Robustness of CCI and Content Protection Within Card and Host Devices

As required in SCTE 41, Copy Control Information, CCI, is delivered to Cards via a secure CA system and delivered to Host Devices across the CableCARD Interface via an authenticated tunnel protocol. Authorized output interfaces, e.g., 1394/5C, specify methods of content protection within their domains. To give effect to the intended content security Card and Host devices must robustly store, distribute and apply the received CCI values within their implementation. Because CCI is temporal and may change over time on a given channel, it must remain temporally associated with the intended content.

Card and Host devices **SHALL** robustly protect CCI values against unauthorized modification, substitution, and loss of temporal association such that the CCI delivered for specific content will control the specified parameters for output of that content.

Specifically, Card and Host devices **SHALL NOT** include (i) switches, buttons, jumpers or software equivalents of any of the foregoing, (ii) specific traces that can be cut, or (iii) service menus or functions (including remote-control functions), in each case by which intended content protection can be defeated or by which Controlled Content can be exposed to unauthorized copying.

9 SECURE SOFTWARE DOWNLOAD

An OpenCable Host 2.0 that implements OCAP 1.0 or 2.0 requirements includes an embedded DOCSIS cable modem that **MUST** implement BPI+ [2], except for the changes noted in this section. OpenCable Host devices implementing OCAP 1.0 [17] or 2.0 **MUST** also be able to have their software upgraded according to the requirements of the CableCARD Interface Specification [16].

When upgrading software using eDOCSIS cable modem, the Host device will verify the code file using download parameters that include the CableLabs CVC Root CA Certificate, the CableLabs CVC CA Certificate, and the Code Verification Certificate (CVC). Requirements for DOCSIS software upgrade are specified in references [3] and [2]. Note that the code verification is performed against the CableLabs CVC Root CA instead of the DOCSIS Root CA.

9.1 Initialization of Download per DOCSIS Requirements

In support of code verification, the Host Configuration File [3] is used as an authenticated means in which to initialize the code verification process. In the configuration file, the OpenCable Host receives configuration settings relevant to code upgrade verification.

9.1.1 CVC Delivery

The configuration file [3] **SHOULD** always include the most up-to-date CVC applicable for the destination OpenCable Host. When the configuration file is used to initiate a code upgrade, it **MUST** include a CVC to initialize the OpenCable Host for accepting code files. Regardless of whether a code upgrade is required, a CVC in the configuration file **MUST** be processed by the OpenCable Host. A configuration file **MAY** contain:

- No CVC – The OpenCable Host **MUST NOT** accept a code file.
- A Manufacturer's CVC only – The OpenCable Host **MUST** verify that the manufacturer's CVC chains up to the CableLabs CVC Root before accepting a code file. When the OpenCable Host configuration file only contains a valid Manufacturer's CVC, then the OpenCable Host will only require a manufacturer signature on the code files. In this case, the OpenCable Host **MUST NOT** accept code files that have been cosigned.
- A Cosigner's (CableLabs or MSO) CVC only – The OpenCable Host **MUST** verify that the Cosigner CVC chains up to the CableLabs CVC Root CA before accepting a code file. When the OpenCable Host configuration file contains a valid cosigner's CVC, it is used to initialize the device with a cosigner. Once validated, the name of the CVC's subject organizationName will become the code cosigner assigned to the OpenCable Host. In order for OpenCable Host to subsequently accept a code image, the cosigner in addition to the OpenCable Host manufacturer **MUST** have signed the code file.
- Both a Manufacturer's CVC and a Cosigner's CVC – The OpenCable Host **MUST** verify that both CVCs chain up to the CableLabs CVC Root before accepting a code file.

9.1.1.1 CVC Processing

To expedite the delivery of an updated CVC without requiring the OpenCable Host to process a code upgrade, the CVC **MAY** be delivered in either the configuration file or an SNMP MIB. The format of the CVC is the same whether it is in a code file, configuration file, or SNMP MIB.

9.1.1.1.1 Processing the Configuration File CVC

When a CVC is included in the configuration file, the OpenCable Host MUST verify the CVC before accepting any of the code upgrade settings it contains. At receipt of the CVC in the configuration file, the OpenCable Host must perform the following validation and procedural steps. If any of the following verification checks fail, the OpenCable Host MUST immediately halt the CVC verification process and log the error if applicable. If the OpenCable Host configuration file does not include a CVC that validates properly, the OpenCable Host MUST NOT download upgrade code files whether triggered by the configuration file or via an SNMP MIB. In addition, if the OpenCable Host configuration files does not include a CVC that validates properly, the OPENCABLE HOST is not required to process CVCs subsequently delivered via an SNMP MIB, and MUST NOT accept information from a CVC subsequently delivered via an SNMP MIB.

At receipt of the CVC in a configuration file, the OpenCable Host MUST:

1. Verify that the extendedKeyUsage extension is in the CVC and includes an OID id-kp-codeSigning.
2. Check the CVC subject organization name.
 - a) If the CVC is a Manufacturer's CVC (Type 32) [2] then:
 - i. If the organizationName is identical to the OpenCable Host's manufacturer name, then this is the manufacturer's CVC. In this case, the OpenCable Host MUST verify that the manufacturer's CVC validity start time is greater-than or equal-to the manufacturer's cvcAccessStart value currently held in the OpenCable Host.
 - ii. If the organizationName is not identical to the OpenCable Host's manufacturer name, then this CVC MUST be rejected and the error logged.
 - b) If the CVC is a Cosigner's CVC (Type 33) [2] then:
 - i. If the organizationName is identical to the OpenCable Hosts current code cosigner, then this is the current cosigner's CVC and the OpenCable Host MUST verify that the validity start time is greater-than or equal-to the cosigner's cvcAccessStart value currently held in the OpenCable Host.
 - ii. If the organizationName is not identical to the current code cosigner name, then after the CVC has been validated (and registration is complete) this subject organization name will become the OpenCable Host's new code cosigner. OpenCable Host MUST NOT accept a code file unless it has been signed by the manufacturer, and cosigned by this code cosigner.
3. Validate the CVC issuer signature using the CL CVC CA Public Key held by the OpenCable Host.
4. Update the OpenCable Host's current value of cvcAccessStart corresponding to the CVC's subject organizationName (i.e., manufacturer or cosigner) with the validity start time value from the validated CVC. If the validity start time value is greater than the OpenCable Host's current value of codeAccessStart, update the OpenCable Host's codeAccessStart value with the validity start time value. The OpenCable Host SHOULD discard any remnants of the CVC.

9.1.1.1.2 Processing the SNMP CVC

The OpenCable Host MUST process SNMP delivered CVCs when enabled to upgrade code files; otherwise, all CVCs delivered via SNMP MUST be rejected. When validating the CVC delivered via SNMP, the OPENCABLE HOST MUST perform the following validation and procedural steps. If any of the following verification checks fail, the OpenCable Host MUST immediately halt the CVC verification process, log the error if applicable, and remove all remnants of the process to that step.

At receipt of the CVC via SNMP, the OpenCable Host MUST:

1. Verify that the extendedKeyUsage extension is in the CVC and includes an OID id-kp-codeSigning.

2. Check the CVC subject organization name.
 - a) If the organizationName is identical to the OpenCable Hosts manufacturer name, then this is the manufacturer's CVC. In this case, the OpenCable Host MUST verify that the manufacturer's CVC validity start time is greater-than the manufacturer's cvcAccessStart value currently held in the OpenCable Host.
 - b) If the organizationName is identical to the OpenCable Hosts current code cosigner, then this is a current cosigner's CVC and the validity start time MUST be greater-than the cosigner's cvcAccessStart value currently held in the OpenCable Host.
 - c) If the organizationName is not identical to OpenCable Host's manufacturer or current cosigner's name, then the OpenCable Host MUST immediately reject this CVC.
3. Validate the CVC issuer signature using the CL CVC CA Public Key held by the OpenCable Host.
4. Update the current value of the subject's cvcAccessStart values with the validated CVC's validity start time value. If the validity start time value is greater than the OpenCable Host's current value of codeAccessStart, update the OpenCable Hosts codeAccessStart value with the validity start value. All certificate parameters EXCEPT for the validity start time are no longer needed and SHOULD be discarded.

9.1.2 Time Varying Controls

To mitigate the possibility of an OpenCable Host receiving a previous code file via a replay attack, the code files include a signing-time value in the PKCS #7 structure that can be used to indicate the time the code image was signed. The OpenCable Host MUST keep two UTC time values associated with each code-signing agent. One set MUST always be stored and maintained for the OpenCable Host's manufacturer. Additionally, if the code file is cosigned, the OpenCable Host MUST also store and maintain a separate set of time values for the cosigner.

These values are used to control code file access to the OpenCable Host by individually controlling the validity of the CVS and the CVC. These values are:

- CodeAccessStart – a 12-byte UTC time value references to Greenwich Mean Time (GMT)
- CvcAccessStart – a 12-byte UTC time value references to GMT

UTCTime values in the CVC MUST be expressed as GMT and MUST include seconds. That is, they MUST be expressed in the following form: YYMMDDHHMMSSZ. The year field (YY) MUST be interpreted as 20YY.

These values will always be referenced to GMT, so the final ASCII character (Z) can be removed when stored by the OpenCable Host as codeAccessStart and cvcAccessStart.

The OpenCable Host MUST maintain each of these time values in a format that contains equivalent time information and accuracy to the 12-character UTC format (i.e., YYMMDDHHMMSS).

The values of codeAccessStart and cvcAccessStart corresponding to the OpenCable Host's manufacturer MUST NOT decrease. The value of codeAccessStart and cvcAccessStart corresponding to the cosigner MUST NOT decrease as long as the cosigner does not change and the OpenCable Host maintains that cosigner's time-varying control values.

9.2 Transport

The download MUST be performed with TFTP or using the DSMCC Carousel.

9.3 Trigger Mechanism

The download can be initiated through either SNMP or through the configuration file, or the CVT.

9.3.1 Configuration File Initiated Download

Configuration file initiated download **MUST** be supported.

The Configuration-file-initiated software download is initiated by sending the Software Upgrade File Name in the configuration file. If the Software Upgrade File Name in the configuration file does not match the current software image of the OpenCable Host, the OpenCable Host **MUST** request the specified file via TFTP from the Software Server.

Note: The Software Server IP Address is a separate parameter. If present, the OpenCable Host **MUST** attempt to download the specified file from this server. If not present, the OpenCable Host **MUST** attempt to download the specified file from the configuration file server.

In case where the OpenCable Host reaches the maximum number of retries (max retries = 3) resulting from multiple losses of power or multiple resets during a configuration-file-initiated upgrade, the OpenCable Host's status **MUST** adhere to the following requirements after it is provisioned:

- docsDevSwAdminStatus **MUST** be allowProvisioningUpgrade{2}
- docsDevSwFilename **MUST** be the filename of the software that failed the upgrade process
- docsDevSwServer **MUST** be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus **MUST** be other{5}
- docsDevSwCurrentVer **MUST** be the current version of software that is operating on the OpenCable Host

If the OpenCable Host exhausts required number of TFTP retries by issuing a total of 16 consecutive retries, it **MUST** fall back to last known working image and proceed to an operational state and adhere to the following requirements:

- docsDevSwAdminStatus **MUST** be allowProvisioningUpgrade{2}
- docsDevSwFilename **MUST** be the filename of the software that failed the upgrade process
- docsDevSwServer **MUST** be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus **MUST** be failed{4}
- docsDevSwCurrentVer **MUST** be the current version of the software that is operating on the OpenCable Host.

After the OpenCable Host has completed the configuration-file-initiated secure software upgrade, the OpenCable Host **MUST** reboot and become operational with the correct software image. After the OpenCable Host is provisioned the:

- docsDevSwAdminStatus **MUST** be allowProvisioningUpgrade{2}
- docsDevSwFilename **MAY** be the filename of the software currently operating on the OpenCable Host device
- docsDevSwServer **MAY** be the address of the TFTP server containing the software that is currently operating on the OpenCable Host

- docsDevSwOperStatus MUST be completeFromProvisioning{2}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the OpenCable Host

9.3.2 SNMP Initiated Download

SNMP initiated download MUST be supported.

From the provisioning server:

1. Set docsDevSwServer to the address of the TFTP server for software upgrades.
2. Set docsDevSwFilename to the file pathname of the software upgrade image.
3. Set docsDevSwAdminStatus to Upgrade-from-mgt. docsDevSwAdminStatus MUST persist across resets/reboots until over-written from an SNMP manager or via the configuration file.

The default state of docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2} until it is over-written by ignoreProvisioningUpgrade{3} following a successful SNMP-initiated software upgrade or otherwise altered by the management station. docsDevSwOperStatus MUST persist across resets to report the outcome of the last software upgrade attempt.

If an OpenCable Host suffers a loss of power or resets during SNMP-initiated upgrade, the OpenCable Host MUST resume the upgrade without requiring manual intervention and when the OpenCable Host resumes the upgrade process:

1. docsDevSwAdminStatus MUST be Upgrade-from-mgt{1}
2. docsDevSwFilename MUST be the filename of the software image to be upgrade
3. docsDevSwServer MUST be the address of the TFTP server containing the software upgrade image to be upgraded
4. docsDevSwOperStatus MUST be inProgress{1}
5. docsDevSwCurrentVers MUST be the current version of the software that is operating on the OpenCable Host

In case where the OpenCable Host reaches the maximum number of retries (max retries = 3) resulting from multiple losses of power or resets during an SNMP-initiated upgrade, the OpenCable Host's status MUST adhere to the following requirements after it is registered:

1. docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
2. docsDevSwFilename MUST be the filename of the software that failed the upgrade process
3. docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
4. docsDevSwOperStatus MUST be other{5}
5. docsDevSwCurrentVer MUST be the same version of the software that is operating on the OpenCable Host

If the OpenCable Host exhausts the required number of TFTP retries by issuing a total of 16 consecutive retries, the OpenCable Host MUST fall back to the last known working image and proceed to an operational state and adhere to the following requirements:

1. docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
2. docsDevSwFilename MUST be the filename of the software that failed the upgrade process

3. docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
4. docsDevSwOperStatus MUST be failed{4}
5. docsDevSwCurrentVer MUST be the current version of software that is operating on the OpenCable Host

After the OpenCable Host has completed the SNMP-initiated secure software upgrade, the OpenCable Host MUST reboot and become operational with the correct software image. After the device is operational, it MUST adhere to the following requirements:

1. Set its docsDevSwAdminStatus to ignoreProvisioningUpgrade{3}
2. Set its docsDevOperStatus to completeFromMgt{3}
3. Reboot

The OpenCable Host MUST properly use ignoreProvisioningUpgrade status to ignore software upgrade values that may be included in the configuration file and become operational with the correct software image. After the device is operational, it MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be ignoreProvisioningUpgrade{3}
- docsDevSwFilename MAY be the filename of the software currently operating on the OpenCable Host
- docsDevSwServer MAY be the address of the TFTP server containing the software that is currently operating on the OpenCable Host
- docsDevSwOperStatus MUST be completeFromMgt{3}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the OpenCable Host

In the case where the OpenCable Host successfully downloads (or detects during download) an image that is not intended for the OpenCable Host, the OpenCable Host MUST adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process
- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of the software that is operating on the OpenCable Host

In the case where the OpenCable Host determines that the download image is damaged or corrupted, it MUST reject the image. The OpenCable Host MAY re-attempt to download if the MAX number of TFTP sequence retries has not been reached. If the OpenCable Host chooses not to retry and the MAX number of TFTP sequence retry has not been reached, the OpenCable Host MUST fall back to the last known working image and proceed to an operational state, generate appropriate event notification and adhere to the following requirements:

- docsDevSwAdminStatus MUST be allowProvisioningUpgrade{2}
- docsDevSwFilename MUST be the filename of the software that failed the upgrade
- docsDevSwServer MUST be the address of the TFTP server containing the software that failed the upgrade process

- docsDevSwOperStatus MUST be other{5}
- docsDevSwCurrentVer MUST be the current version of software that is operating on the OpenCable Host

9.3.3 CVT Initiated Download

This is described in the OpenCable CableCARD Interface Specification [16].

9.4 Image File Structure

For secure software download, the code download file is a file built using a PKCS #7 compliant structure that has been defined in a specific format for use with OpenCable Host. The code file MUST comply with [14] and MUST be DER encoded. The code file MUST match the structure shown in Table 12.

When downloading the CA Certificates (e.g., a CableLabs CVC CA Certificate and/or a CableLabs Device CA Certificate) as a part of the Code File, the certificates MAY be contained in the fields specified in Table 12, separated from the actual code image contained in the CodeImage field.

Table 12 – Code File Structure

Code File	Description
PKCS #7 Digital Signature {	
ContentInfo	
ContentType	SignedData
SignedData ()	EXPLICIT signed-data content value: includes CVS and X.509 compliant CVCs
} end PKCS #7 Digital Signature	
SignedContent {	
DownloadParameters {	Mandatory TLV format (Type 28). (Length is zero if there is no sub-TLVs.)
MfgCACerts ()	Optional TLV for one or more DER-encoded CableLabs Manufacturer CA Certificate(s) each formatted according to CableLabs Device CA-Certificate TLV format (Type 17) Refer to Section 9.4.2.1.
clabCVCRootCACert()	Optional TLV for one DER-encoded certificate formatted according to the CableLabs CVC Root CA CA-Certificate TLV format (Type 51). Refer to Section 9.4.2.2.
clabCVCCACertificate ()	Optional TLV for one DER-encoded certificate formatted according to the CableLabs CVC CA-Certificate TLV format (Type 52). Refer to Section 9.4.2.3.
}	
CodeImage ()	Upgrade code image.
} end SignedContent	

9.4.1 Signed Data

The code download file will contain the information in a PKCS #7 Signed Data content type as shown below in Table 13. Though maintaining compliance to [14], the structure used has been restricted in format to ease the processing performed by the OpenCable Host to validate the signature. The PKCS #7 SignedData MUST be DER encoded and exactly match the structure shown below except for any change required for DER encoding (e.g., the ordering of SET OF attributes). The OpenCable Host MUST reject the PKCS #7 signature if the PKCS #7 Signed Data does not match the DER encoded structure.

Table 13 – PKCS #7 Signed Data

PKCS #7 Field	Description
SignedData {	
Version	version = 1
DigestAlgorithmIdentifiers	SHA-1
ContentInfo	
ContentType	data (SignedContent is concatenated at the end of the PKCS #7 structure)
certificates {	CableLabs Code Verification Certificates
MfgCVC	(REQUIRED for all code files)
CosignerCVC	(OPTIONAL; required for cosignatures)
} end certificates	
SignerInfo {	
MfgSignerInfo {	(REQUIRED for all code images)
Version	version = 1
IssuerAndSerialNumber	
IssuerName	
CountryName	US
OrganizationName	CableLabs
CommonName	CableLabs CVC CA
CertificateSerialNumber	<CableLabs CVC CA serial number>
DigestAlgorithm	SHA-1
AuthenticatedAttributes	
ContentType	data (contentType of signedContent)
SigningTime	UTC Time (GMT), YYMMDDHHMMSSZ
MessageDigest	Digest of the content as defined in PKCS #7
DigestEncryptionAlgorithm	RsaEncryption
EncryptedDigest	
} end mfg signer info	
CoSignerInfo {	(OPTIONAL; required for cosignatures)
Version	version = 1
IssuerAndSerialNumber	
IssuerName	
CountryName	US
OrganizationName	CableLabs/MSO
CommonName	CableLabs/MSO CVC
CertificateSerialNumber	<Cosigner CVC serial number>
DigestAlgorithm	SHA-1
AuthenticatedAttributes	
ContentType	data (contentType of signedContent)
SigningTime	UTC Time (GMT), YYMMDDHHMMSSZ
MessageDigest	Digest of the content as defined in PKCS #7
DigestEncryptionAlgorithm	RsaEncryption
EncryptedDigest	
} end CoSignerInfo	
} end SignerInfo	
} end SignedData	

9.4.2 Signed Content

The signed content field of the code file contains the code image and the download parameters field, which possibly contains additional optional items - CA Certificates (e.g., a CableLabs CVC CA Certificate and/or a CableLabs Device CA Certificate).

The final code image is in a format compatible with the destination OpenCable Host. In support of the PKCS #7 signature requirements, the code content is typed as data; i.e., a simple octet string. The format of the final code image is not specified here and will be defined by each manufacturer according to their requirements.

Each manufacturer SHOULD build their code with additional mechanisms that verify an upgrade code image is compatible with the destination OpenCable Host.

If included in the signed content field, the certificate fields are intended to replace the CableLabs CVC Root CA or the CableLabs CVC CA Certificate(s) currently stored in the OpenCable Host. If the code download and installation is successful, the OpenCable device MUST update its currently stored certificates with the ones contained in the MfgCACerts, clabCVCRootCACert and clabCVCCACertificate fields of the signed content field in the new Code File.

9.4.2.1 CableLabs Device CA-Certificate TLV Format

The attribute is a string attribute containing an X.509 CA Certificate, as defined in [14].

Type	Length	Value
17	Variable	X.509 CableLabs CableLabs Device CA Certificate (DER-encoded ASN.1)

9.4.2.2 CVC Root CA CA-Certificate TLV Format

The attribute is a string attribute containing an X.509 CA Certificate, as defined in [14].

Type	Length	Value
51	Variable	X.509 CableLabs CVC Root CA Certificate (DER-encoded ASN.1)

9.4.2.3 CVC CA-Certificate TLV Format

The attribute is a string attribute containing an X.509 CA Certificate, as defined in [14].

Type	Length	Value
52	Variable	X.509 CableLabs CVC CA Certificate (DER-encoded ASN.1)

9.4.3 Code Signing Keys

The PKCS #7 digital signature uses the RSA Encryption Algorithm with SHA-1 [14]. The OpenCable Host MUST be able to verify code file signatures.

9.5 Image Validation

The OpenCable Host MUST perform the verification checks presented in this section. If any of the verification checks fail, or if any portion of the code file is rejected due to invalid formatting, the OpenCable Host MUST immediately halt the download process, log the error if applicable, remove all remnants of the process to that step, and continue to operate with its existing code. The verification checks can be made in any order, as long as all of the applicable checks presented in this section are made.

1. The OpenCable Host MUST validate the manufacturer's signature information by verifying that the PKCS #7 signingTime value is:
 - a) strictly greater-than the manufacturer's codeAccessStart value currently held in the OpenCable Host.
 - b) equal-to or greater-than the manufacturer's CVC validity start time.
 - c) less-than or equal-to the manufacturer's CVC validity end time.

2. The OpenCable Host MUST validate the manufacturer's CVC by verifying that the:
 - a) CVC subject organizationName is identical to the manufacturer name currently stored in the OpenCable Hosts memory.
 - b) CVC validity start time is equal-to or greater-than the manufacturer's cvcAccessStart value currently held in the OpenCable Host.
 - c) extendedKeyUsage extension is in the CVC and includes an OID id-kp-codeSigning.
3. The OpenCable Host MUST validate the certificate signature of each CVC contained in the Signed Data using the CableLabs CVC CA Public Key held by the OpenCable Host. Verification of the signature will authenticate the source of the public code verification key (CVK) and confirm trust in the key. Once trust has been established in the manufacturer's and cosigner's CVK, the remaining certificate parameters except for the validity start time are no longer needed and SHOULD be discarded.
4. The OpenCable Host MUST verify the manufacturer's code file signature.
 - a) The OpenCable Host MUST perform a new SHA-1 hash over the SignedContent. If the value of the messageDigest doesn't match the new hash, the OpenCable Host MUST consider the signature on the code file as invalid.
 - b) If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process MUST be rejected and SHOULD be immediately discarded.
5. If the manufacturer signature verifies and a cosigning agent signature is required:
 - a) The OpenCable Host MUST validate the cosigner's signature information by verifying that the:
 - i. cosigner's signature information is included in the code file.
 - ii. PKCS #7 signingTime value is strictly greater-than the corresponding codeAccessStart value currently held in the OpenCable Host.
 - iii. PKCS #7 signingTime value is equal-to or greater-than the corresponding CVC validity start time.
 - iv. PKCS #7 signingTime value is less-than or equal-to the corresponding CVC validity end time.
 - b) The OpenCable Host MUST validate the cosigner's CVC, by verifying that the:
 - i. CVC subject organizationName is identical to the cosigner's organization name currently stored in the OpenCable Hosts memory.
 - ii. CVC validity start time is equal-to or greater-than the cvcAccessStart value currently held in the OpenCable Host for the corresponding subject organizationName.
 - iii. extendedKeyUsage extension is in the CVC.
 - c) The OpenCable Host MUST validate the certificate signature using the CL CVC CA certificate held by the OpenCable Host. Verification of the signature will authenticate the source of the co-signer's public code verification key (CVK) and confirm trust in the key. Once trust has been established in the cosigner's CVK, the remaining certificate parameters except for the validity start time are no longer needed and SHOULD be discarded.
 - d) The OpenCable Host MUST verify the cosigner's code file signature.

- e) The OpenCable Host MUST perform a new SHA-1 hash over the SignedContent. If the value of the messageDigest doesn't match the new hash, the OpenCable Host MUST consider the signature on the code file as invalid.
 - f) If the signature does not verify, all components of the code file (including the code image), and any values derived from the verification process MUST be rejected and SHOULD be immediately discarded.
6. The OpenCable Host MUST verify that the downloaded software image is appropriate for itself.
 7. If the manufacturer's, and optionally the cosigner's, signature has been verified, the code image can be trusted and installation may proceed. Before installing the code image, all other components of the code file and any values derived from the verification process except the PKCS #7 signingTime values and the CVC validity start values SHOULD be immediately discarded.
 8. If the code installation is unsuccessful, the OpenCable Host MUST reject the PKCS #7 signingTime values and CVC validity start values it just received in the code file.
 9. When the code installation is successful, the OpenCable Host MUST update the manufacturer's time varying controls with the values from the manufacturer's signature information and CVC:
 - a) Update the current value of codeAccessStart with the PKCS #7 signingTime value.
 - b) Update the current value cvcAccessStart with the CVC validity start value.
 10. When the code installation is successful and if the code file was cosigned, the OpenCable Host MUST update the co- signer's time-varying controls with the values from the cosigner's signature information and CVC:
 - a) Update the current value of codeAccessStart with the PKCS #7 signingTime value.
 - b) Update the current value of cvcAccessStart with the CVC validity start value.

9.6 Error Codes

Error codes are defined to reflect the failure states possible during the secure software download code verification process. These error codes are managed by the device manufacturer in storing these codes and displaying them through diagnostic screens.

1. Improper code file controls.
 - a) CVC subject organizationName for manufacturer does not match the OpenCable Hosts manufacturer name.
 - b) CVC subject organizationName for code cosigning agent does not match the OpenCable Hosts current code cosigning agent.
 - c) The manufacturer's PKCS #7 signingTime value is equal-to or less-than the codeAccessStart value currently held in the OpenCable Host.
 - d) The manufacturer's PKCS #7 validity start time value is less-than the cvcAccessStart value currently held in the OpenCable Host.
 - e) The manufacturer's CVC validity start time is less-than the cvcAccessStart value currently held in the OpenCable Host.
 - f) The manufacturer's PKCS #7 signingTime value is less-than the CVC validity start time.
 - g) Missing or improper extendedKeyUsage extension in the manufacturer CVC.
 - h) The cosigner's PKCS #7 signingTime value is equal-to or less-than the codeAccessStart value currently held in the OpenCable Host.

- i) The cosigner's PKCS #7 validity start time value is less-than the cvcAccessStart value currently held in the OpenCable Host.
 - j) The cosigner's CVC validity start time is less-than the cvcAccessStart value currently held in the OpenCable Host.
 - k) The cosigner's PKCS #7 signingTime value is less-than the CVC validity start time.
 - l) Missing or improper extended key-usage extension in the cosigner's CVC.
2. Code file manufacturer CVC validation failure.
 3. Code file manufacturer CVS validation failure.
 4. Code file cosigner CVC validation failure.
 5. Code file cosigner CVS validation failure.
 6. Improper Configuration File CVC format (e.g., Missing or improper key usage attribute)
 7. Configuration File CVC validation failure.
 8. Improper SNMP CVC format.
 - a) CVC subject organizationName for manufacturer does not match the OpenCable Hosts manufacturer name.
 - b) CVC subject organizationName for code cosigning agent does not match the OpenCable Hosts current code cosigning agent.
 - c) The CVC validity start time is less-than or equal-to the corresponding subject's cvcAccessStart value currently held in the OpenCable Host.
 - d) Missing or improper key usage attribute.
 9. SNMP CVC validation failure.

Appendix I Revision History

OC-SP-SEC-I02-030707 contains modifications from the following ECNs.

ECN	Date Accepted	Summary
03-0392	4/1/03	Correct commas in X.509 Certificate profiles and validity periods.
03-0406	6/2/03	Add subKeyIdentifier to the Application and CableLabs CVC certificates for OCAP.
03-0407	6/16/03	Clarify the KeyUsage extension in each certificate profile.
03-0405	6/23/03	Protection and Handling of Critical Security Parameters.

OC-SP-SEC-I03-031121 contains modifications from the following ECNs. Version I03 also contains editorial changes, updating POD to CableCARD.

ECN	Date Accepted	Summary
03-0465	10/20/03	Change Manufacturer CA to CableLabs Device CA in Certificate Management Architecture

OC-SP-SEC-I04-040402 contains modifications from the following ECNs.

ECN	Date Accepted	Summary
SEC-N-03.0527-4	2/19/04	Protect CCI Parameter
SEC-N-03.0547-2	2/20/04	Correct RFC 3280 compliance
SEC-N-04.0555-4	2/20/04	Device Certificate Validation

OC-SP-SEC-I05-040831 contains modifications from the following ECNs.

ECN	Date Accepted	Summary
SEC-N-04.0596-4	6/10/04	Typo correction in the description on the CableLabs Application Code Verification Certificate
SEC-N-04.0631-2	6/24/04	Secure software download correction in Security Spec
SEC-N-04.0642-5	8/9/04	Clarification of Certificate Profile - part 1
SEC-N-04.0643-5	8/9/04	Clarification of Certificate Profile - part 2

OC-SP-SEC-I06-060413 contains modifications from the following ECNs.

ECN	Date Accepted	Summary
SEC-N-05.0826-5	2/3/06	Certificate Profile Changes