



OpenCable System Security

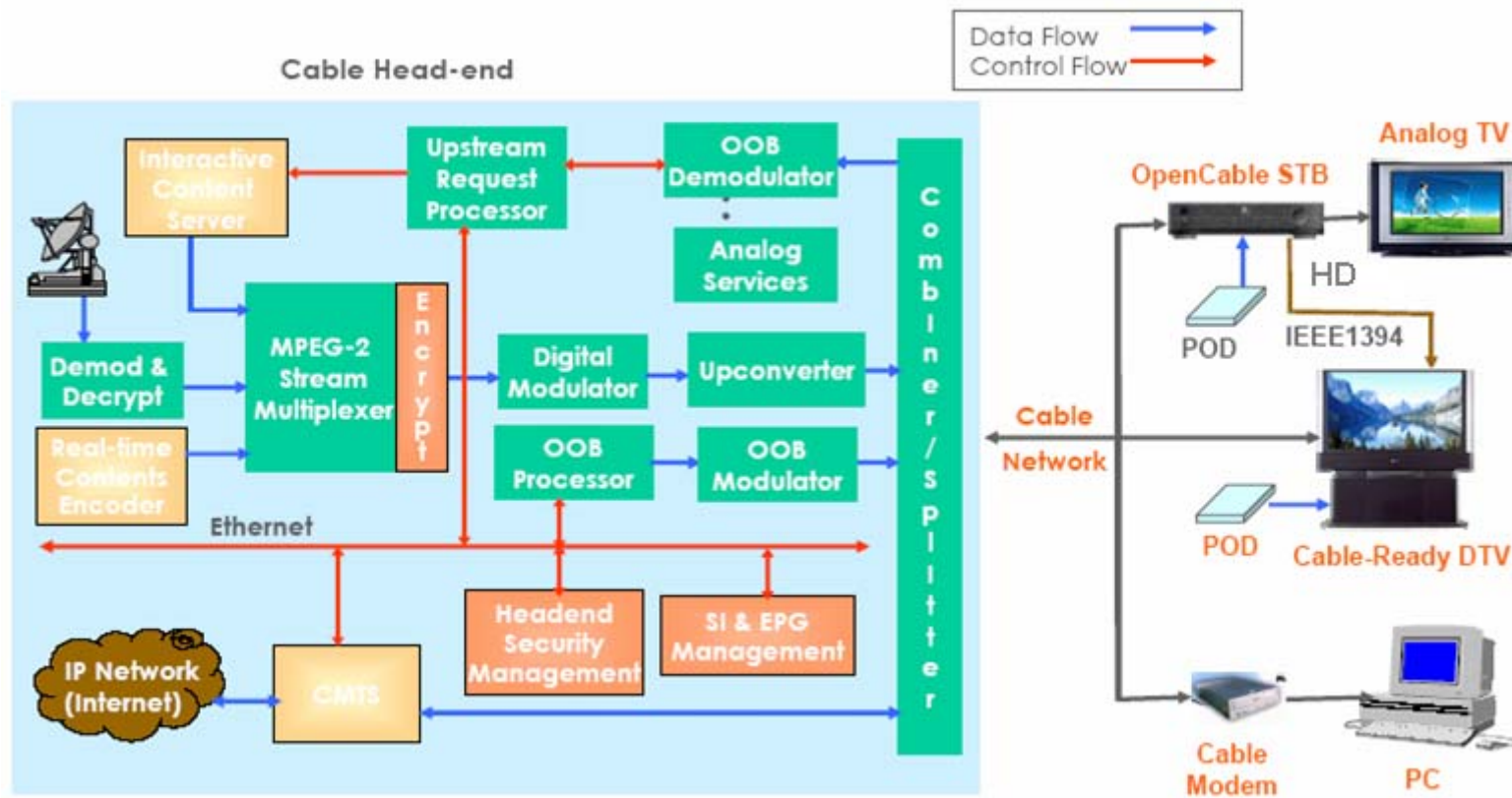
Jae Han Lee
Young Woo Ahn



Contents

- OpenCable System Overview
- Terms and Definitions
- Introduction
- Copy Protection
- OpenCable certificate profiles
- OpenCable certificate management
- References

OpenCable System Overview



- OpenCable System Overview



Terms and Definitions (1/5)

- CableLabs®
 - a nonprofit research and development consortium, is dedicated to helping its cable operator members integrate new cable telecommunications technologies into their business objectives.
- OpenCable
 - a set of specifications created by CableLabs to "Define the next-generation of advanced digital cable-ready devices".
- CableCARD
 - a detachable device distributed by cable providers such as Comcast, Time Warner or Cox. also called "Point of Deployment" (POD) module.



Terms and Definitions (2/5)

- Certification Authority
 - An entity authorized to issue, manage, revoke, and renew Certificates
- CableLabs Manufacturer Root CA
 - The X.509 certificate authority controlled by CableLabs to issue CA Certificates.
- CableLabs Device CA
 - An X.509 certificate authority authorized by the CableLabs Manufacturer Root CA to issue Card or Host Certificates.



Terms and Definitions (3/5)

- Certificate
 - a message that, at least, states a name or identifies the CA, the Subscriber, and the Certificate's Validity Period, contains the Subscriber's public key, contains a Certificate serial number, and is digitally signed by a CA.
- 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.



Terms and Definitions (4/5)

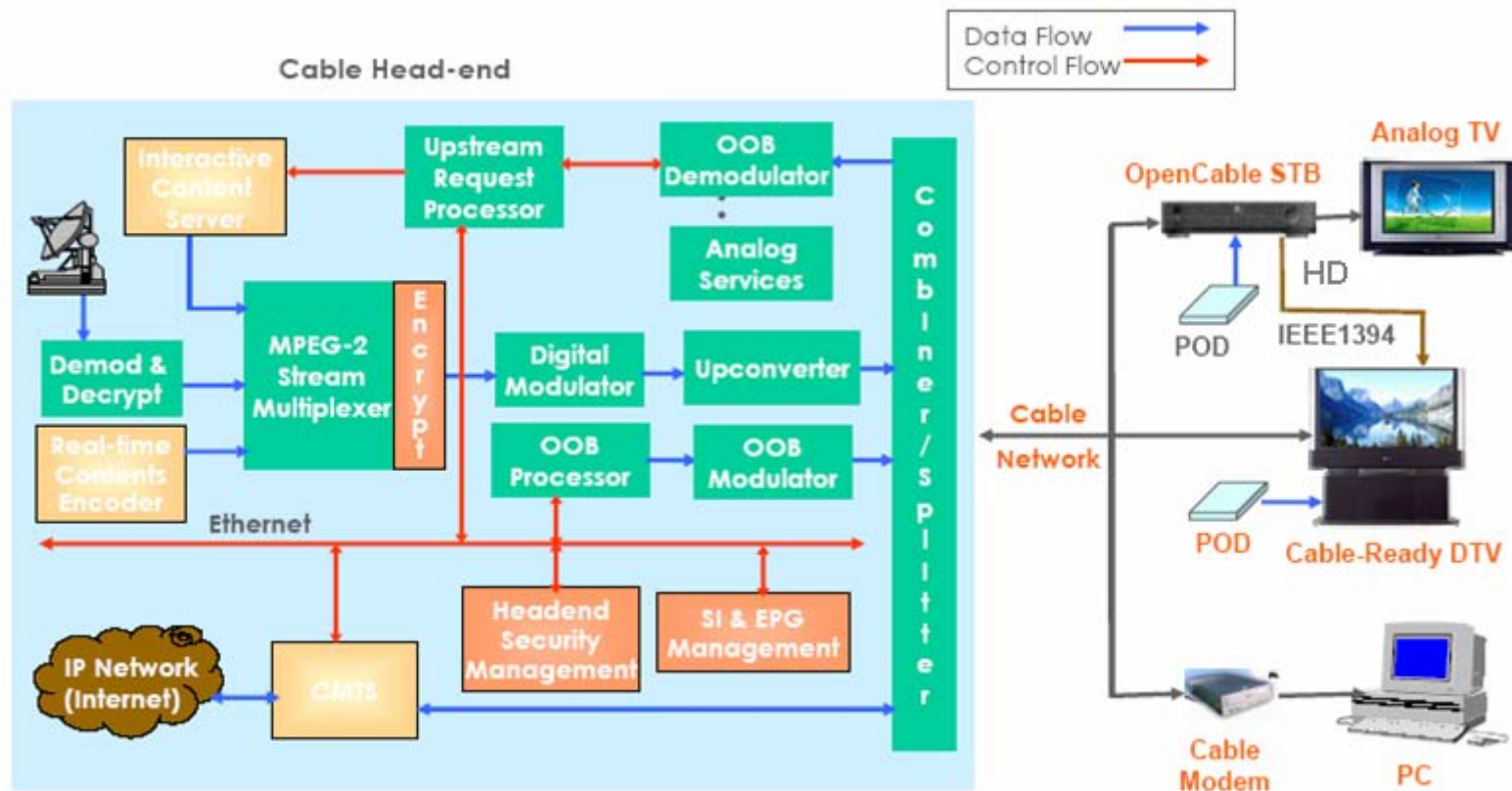
- CA Certificate
 - an X.509 version 3 certificate of the CA that issues Device Certificates for device identity authentication
- Device Certificate
 - an X.509 version 3 certificate used for Card and Host identity authentication. It is issued by a CableLabs CA.



Terms and Definitions (5/5)

- RSA
 - A public key cryptographic system invented by Rivest, Shamir, and Adelman.
- DFAS
 - Dynamic Feedback Arrangement Scrambling Technique, a component of the encryption algorithm
- Certification Revocation List (CRL)
 - A periodically issued list, digitally signed by a CA, of identified Certificates that have been revoked prior to their expiration dates.

Introduction



- OpenCable System Overview



Introduction

- OpenCable systems consist of three main parts;
 - Headend
 - CableCARD (Point-Of-Deployment)
 - Host device (set-top box or digital TV)
- In digital Cable systems, a conditional access scrambling system and the copy protection system protect high value movies and video programs.

Introduction



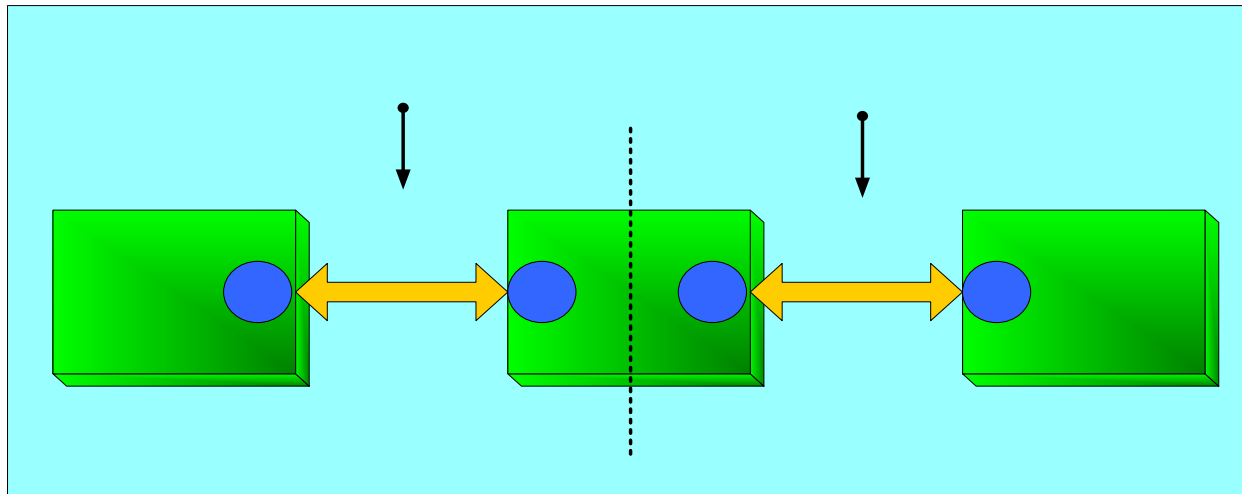
- POD is connected to the Host through the PCMCIA interface.



Introduction

- A properly authorized CableCARD security device
 - Previously referred to as a Point of Deployment (POD) module
 - eliminates the scrambling.
 - rescrambles the content before sending it to the Host devices.

Copy Protection



- Conditional Access Service : the protection of content by service provider
 - Copy Protection Service : preventing the duplication of the content
- Protection for illegal service**



Copy Protection

- This mechanism is analyzed into three steps.
- First Step
 - The CableCARD certifies the Host Certificate and then obtains the Host ID.
 - If the certification in the CableCARD fails, it notifies the CA (Conditional Access) system in the Headend.
 - If it succeeds, POD transmits POD ID and Host ID to the Headend CA system.



Copy Protection

- Second Step
 - The Headend CA system saves the Host ID and the POD ID and check if the Host ID is in the CRL (Certification Revocation List).
 - If the Host ID is not in the CRL, the private CA system Host ID validation message is sent to the POD.
 - POD compares the Host ID saved in it with the received validate Host ID. If it is true, the process of creating the certification key is carried out; otherwise, “Mismatched ID” is shown to the user, CA descramble service is made only for the content with EMI “00”



Copy Protection

- Third Step
 - POD creates the key (AuthKeyP) from the Headend validated IDs and other data, and compares it with AuthKeyH.
 - If the two keys are equal, POD and Host save them in the non-volatile memory (NVRAM), and then finish the key exchange procedure.
 - If the keys are different, POD send the contents with EMI bits 01, 10, 11 in pass-through mode to the Host.

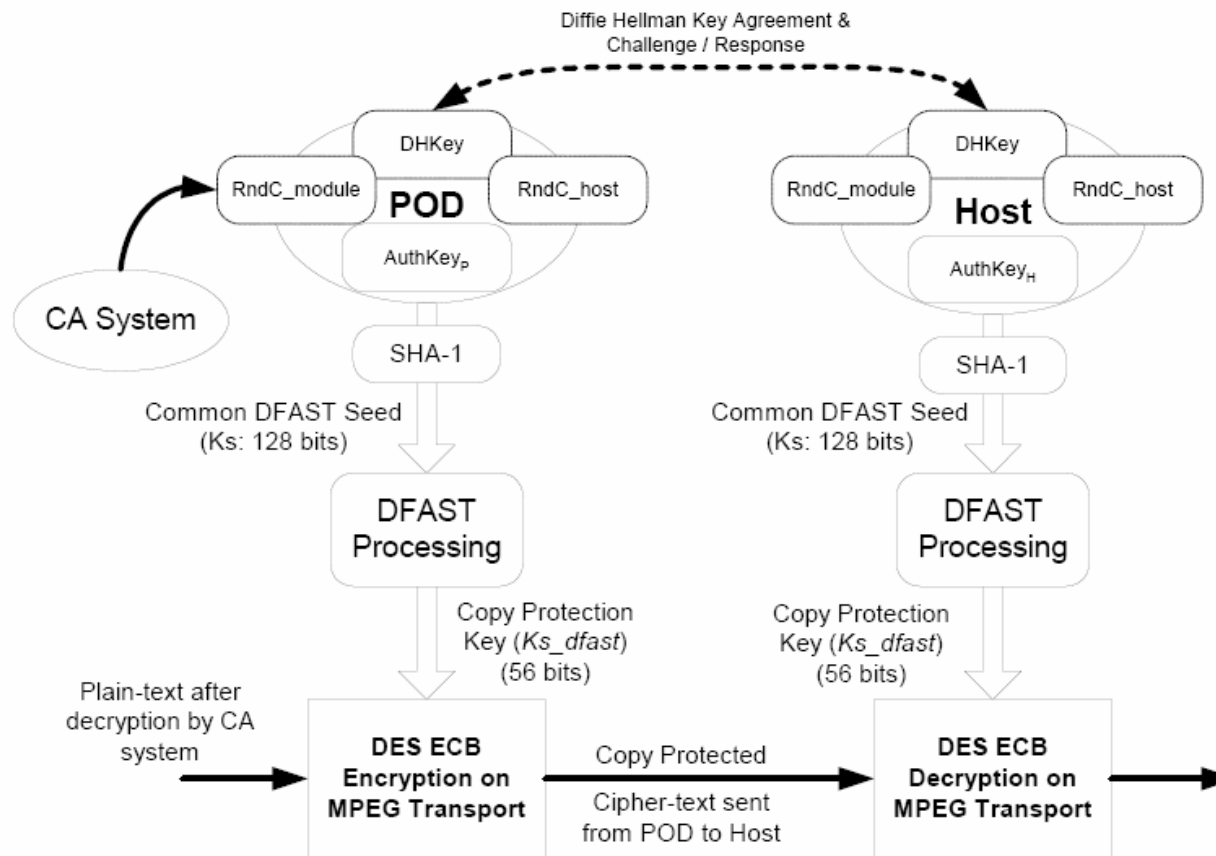


Copy Protection

Bits Value	EMI (Encryption Mode Indicator)	
	Copy Protection	Value
00	Copying is permitted	Not high
01	No future copying is permitted	High
10	One generation copy is permitted	High
11	Copying is prohibited	High

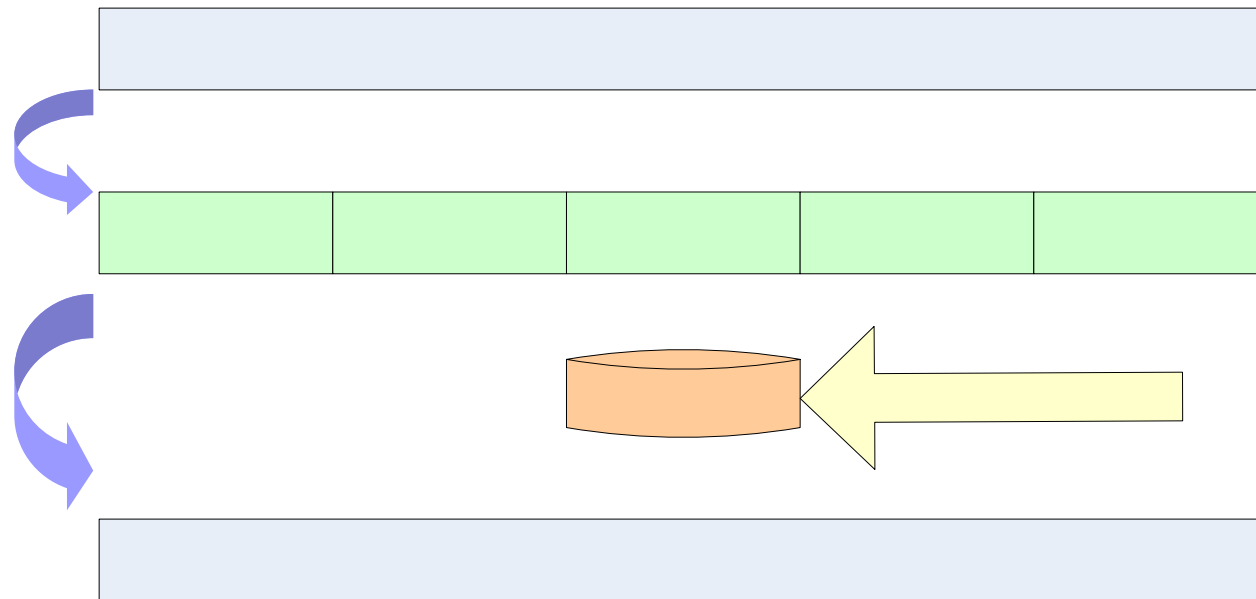
- EMI bits are included in the one-byte copy control information, which is transferred from Headend to POD by a private secure delivery means.

Copy Protection



- The basic key negotiation process for POD copy protection [3]

Copy Protection : DES-ECB



Plain Text (MF

- Cipher Text is generated by applying 56bit key to the plain text split into 64bit blocks.

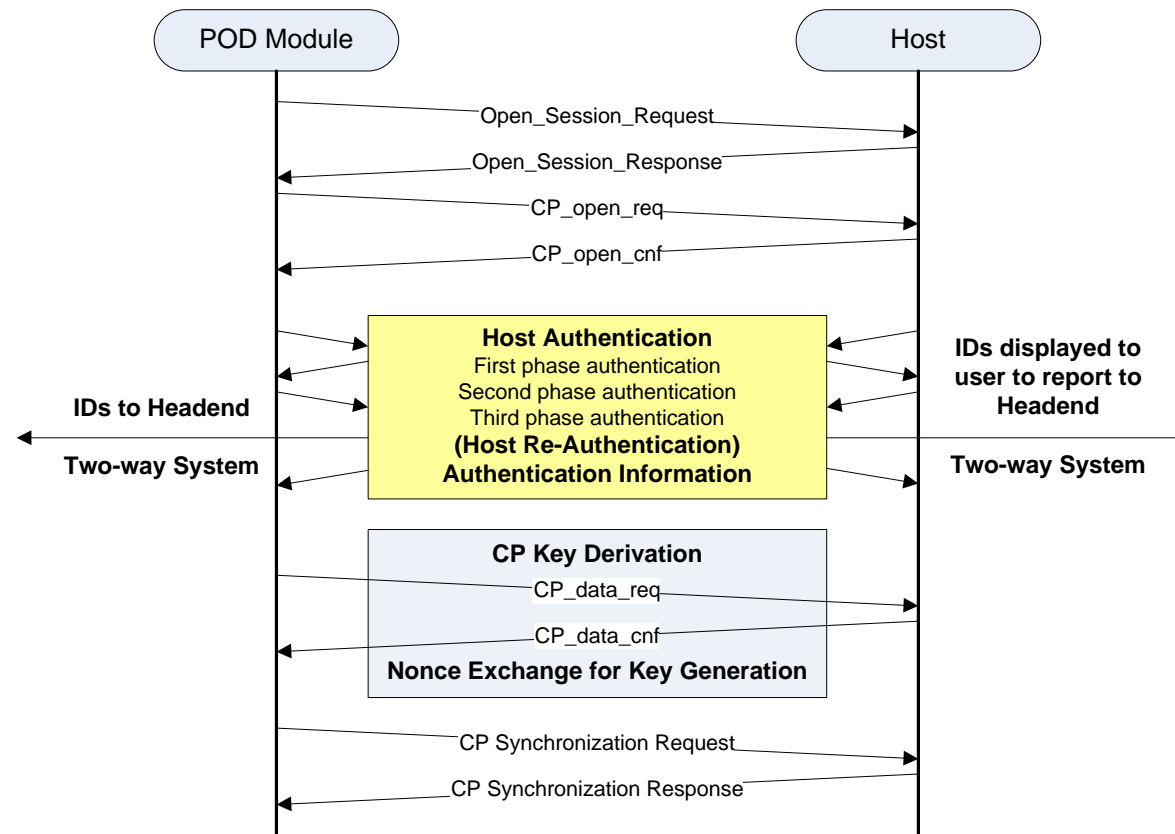
Split into sequence block

64bits

64bits

6

Copy Protection



- POD CPS Authorization and CP Key Derivation



Copy Protection

- Authentication Key Generation
 - During the Host authentication process, Diffie–Hellman public keys are exchanged between POD and Host
 - Diffie–Hellman shared secret key (DHKey) :
$$DHKeyP = (DH_pubKeyH)^x \text{ mod } n =$$
$$(DH_pubKeyP)^y \text{ mod } n = DHKeyH$$
 - Both POD and Host calculate a 160 bit value called the Authentication Key or AuthKey, which is calculated based on the 64 bit POD ID, the 40 bit Host ID, and the DHKey.



Copy Protection

- The POD and the Host computes its authentication key by applying the SHA-1 function
 - $AuthKeyP = \text{SHA-1} [DHKey | \text{Host_ID} | \text{POD_ID}]$
 - $AuthKeyH = \text{SHA-1} [DHKey | \text{Host_ID} | \text{POD_ID}]$
- The Host transfers $AuthKeyH$ to the POD, where the POD compares it against $AuthKeyP$.
- The POD proceeds with authentication if and only if $AuthKeyP$ is identical to $AuthKeyH$.



Copy Protection

- The need for authentication Key generation occurs only once (per Host-POD pair) when the POD and Host are first connected.
- The resulting *AuthKeyP* and *AuthKeyH* values and Diffie-Hellman Secret Key (DHKey) then need to be stored in non-volatile memory, and are used to generate transmission keys later in the key derivation step.



Copy Protection

- Copy Protection Key Generation
 - The POD and the Host compute the SHA-1 key, K_s , based on the *AuthKey*, shared secret DH key (*DHKey*), and the random numbers exchanged in the key generation
 - $K_s = \text{SHA-1} [\textit{AuthKeyP} \mid \textit{DHKey} \mid N_{\textit{Host}} \mid N_{\textit{module}}]_{\text{MSB128}}$
 - $K_s = \text{SHA-1} [\textit{AuthKeyH} \mid \textit{DHKey} \mid N_{\textit{Host}} \mid N_{\textit{module}}]_{\text{MSB128}}$
 - $\text{CP-Key} = K_{s_dfast} = \text{DFAST} [K_s]$
 - The 160-bit SHA-1 output is truncated to its 128 MSB's, left-most bits, to generate a seed, K_s , with the proper length for the DFAST engine.



Copy Protection

Key or Variable	Size (bits)	Description
Nonces (N_{Host} , N_{module})	64 bits each	Random numbers used to refresh the CP-Key.
Authentication Keys ($AuthKey_H$, $AuthKey_P$)	160 bits each	Results from the Host authentication process. It is a long-term key, and is stored in a non-volatile memory.
Shared Diffie-Hellman Key ($DHKey$)	1024 bits	The 1024 bit shared DH secret key. It is a long-term key, and is stored in non-volatile memory.
SHA-1 Key (K_s)	128 bits	The most significant 128 bits of the 160 bit SHA-1 output, where the SHA-1 input is the $DHKey$, Authentication Key, and nonces from POD and Host.
Copy Protection Key (K_{s_dfast})	56 bits	DFAST output, final encryption and decryption key

- Length of Keys and Parameters Used in the Key Generation



Certificate Profile

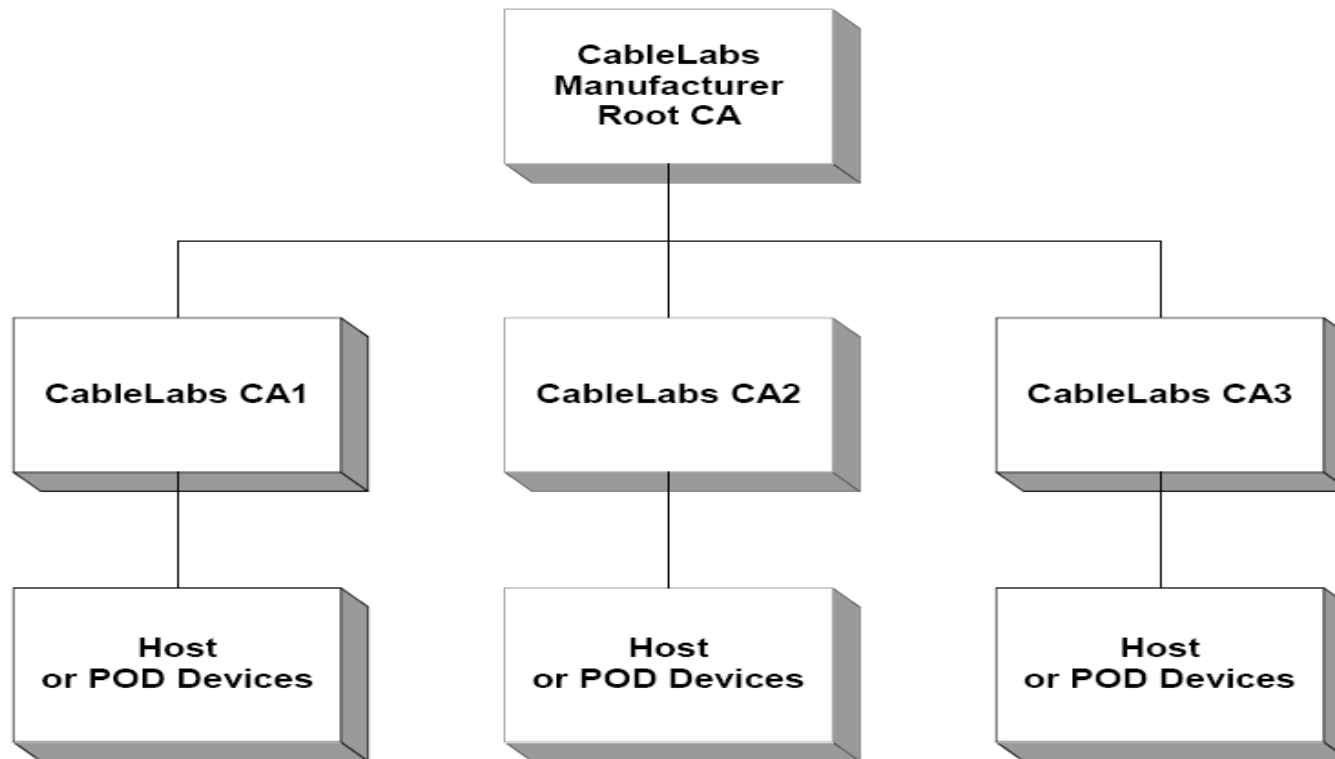
- CableCARD Copy Protection
 - Employs X.509 Version 3 certificates for authenticating key exchanges between the POD and Host
- RSA Public Keys are used throughout the hierarchy.
- The signature algorithm
 - SHA-1 with RSA Encryption



Certificate Profile

- A three-level hierarchy of trust supporting three types of X.509 version3 certificate:
 - A single, self-signed CableLabs Manufacturer Root CA certificate
 - CableLabs Device CA certificates
 - CableCARD and Host certificates
- The CableLabs Manufacturer Root CA serves as the root CA.
 - The root CA issues certificates to the device CAs.
 - The CableLabs Device CAs issue device certificates for CableCARD and Host devices.

Certificate Profile



- CableLabs Device Certificate Hierarchy [1]



Certificate Profile

- CableLabs
 - Responsible for maintaining the CableLabs Manufacturer Root CA and each CableLabs Device CA.
- CableLabs Manufacturer Root CA
 - In charge of generating and distributing to MSO's Certificate Revocation Lists (CRL), which identifies revoked CableLabs Device certificates.



Certificate Profile

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)

- CableLabs Manufacturer Root CA Certificate [1]



Certificate Profile

- CableLabs Device CA certificates
 - Must be issued and signed by the CableLabs Manufacturer Root CA.
 - The state/province and city are optional attributes.



Certificate Profile

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)

- CableLabs Device CA Certificate [1]



Certificate Profile

- Device certificates
 - Must be issued and signed by the CableLabs Device CA.
 - Must have a validity period greater than the operational lifetime of the specific device, because they are permanently installed.



Certificate Profile

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)

- Device Certificate [1]



Certificate Management

- Host and CableCARD Certificate Storage and Management
 - Host or CableCARD certificates must be stored in non-volatile memory.
 - The CableLabs Device Root CA's (RSA) public key must be placed into non-volatile memory.
 - The CableLabs Device CA certificate must be stored in the cable device's non-volatile memory.
 - The CableLabs Manufacturer Root CA certificate must be loaded into both CableCARD and Host devices at manufacture time.

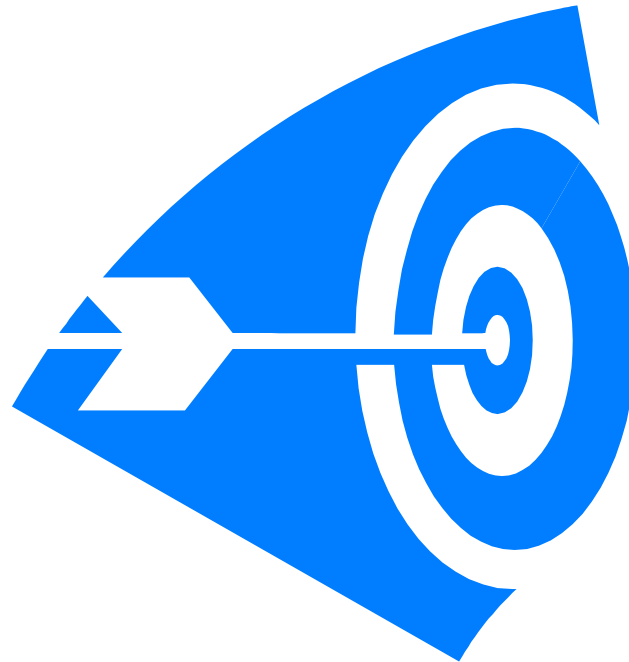


References

- [1] OC-SP-SEC-I05-040831, OpenCable System Security Specification, August 31, 2004, Cable Television Laboratories, Inc., www.opencable.com
- [2] OC-SP-CCCP-IF-I14-040831, OpenCable CableCARD Copy Protection Interface Specification, August 31, 2004, Cable Television Laboratories, Inc.
- [3] SCTE 41 2003 (Formerly DVS 301) POD Copy Protection Standard, Society of Cable Telecommunications Engineers, <http://www.scte.org/documents/pdf/ANSISCTE412004.pdf>



Questions





Appendix - Certificate Format

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.

- X.509 Basic Certificate Fields [1]