CHAPTER 6

Troubleshooting PacketCable eMTA Provisioning

This chapter features information that will help you solve possible issues in a PacketCable voice technology deployment.

- Troubleshooting Tools, page 6-4
- Troubleshooting Scenarios, page 6-5
- Certificate Trust Hierarchy, page 6-9

This chapter assumes that you are familiar with the PacketCable Media Terminal Adapter (MTA) Device Provisioning Specification, PKT-SPPROV1.5-I01-050128. See the PacketCable website for details.

Provisioning PacketCable embedded MTAs (eMTAs) is a relatively complex process; however, with the right tools and 'tricks of the trade,' getting eMTAs operational can be straightforward.

This chapter assumes that Network Registrar and BAC are both in use; however, much of the information also applies for other deployments. Basic knowledge of Network Registrar (scopes, policies, basic DNS zone setup, and record entry) and BAC (Class of Service, DHCP criteria, external files, and BAC directory structure) is assumed.

The PacketCable eMTA provisioning process consists of 25 steps for the Secure flow; the Basic flow has far fewer steps. To troubleshoot eMTAs, knowledge of these 25 steps from the PacketCable provisioning specification is absolutely essential. See PacketCable Voice Configuration, page 5-1.

This section contains the following topics:

- Components, page 6-1
- Key Variables, page 6-3

Components

Before troubleshooting eMTAs, you should be familiar with the following system components.

- eMTA
- DHCP Server
- DNS Server
- KDC
- PacketCable Provisioning Server
- Call Management Server

eMTA

The eMTA is a cable modem and an MTA in one box, with a common software image. The CM and MTA each have their own MAC addresses and each performs DHCP to get its own IP address. The eMTA contains, at minimum, three certificates. One certificate is a unique MTA certificate. A second certificate identifies the MTA manufacturer. Both the device and manufacture certificates are sent by the MTA to authenticate itself to the KDC. The third certificate is a telephony root certificate used to verify the certificates sent by the KDC to the MTA. The KDC certificates will be chained from the telephony root, therefore the telephony root must reside on the MTA to validate the authenticity of the KDC certificates. The MTA portion receives its own configuration file, which it uses to identify its controlling call agent, among other things.

DHCP Server

The DOCSIS specifications mandate that cable modems negotiate their IP address using the Dynamic Host Configuration Protocol (DHCP). The MTA, like most CPE on a DOCSIS network, must use DHCP to obtain its IP address and other crucial information (DNS servers, PacketCable Option 122 for Kerberos realm name, provisioning server FQDN).

Note The cable modem portion, in addition to its normally required DHCP options, also requests, and must receive, Option 122 suboption 1, which it passes to the MTA portion as the IP address of the correct DHCP server from which to accept offers.

When using BAC with PacketCable support, be aware that a correctly configured BAC will automatically populate the ToD server, DNS servers, TFTP server, as well as the Option 122 fields; these do not need to be explicitly set in the Network Registrar policy.

DNS Server

The Domain Name System (DNS) server is fundamental in PacketCable provisioning. The PacketCable provisioning server, which is the device provisioning engine (DPE) in a BAC architecture, must have an address (A) record in the appropriate zone, because its fully qualified domain name (FQDN) is provided to the MTA in Option 122 by the DHCP server. The KDC realm must have a zone of the same name as the realm name containing a server (SRV) record that contains the FQDN of the Kerberos server.

The Kerberos server identified in the SRV record must itself have an A record in the appropriate zone. The call management server (CMS) identified in the MTA configuration file must also have an A record in the appropriate zone. Lastly, the MTAs themselves must have A records in the appropriate zone, since the CMS reaches the MTA by resolving its FQDN. Dynamic DNS (DDNS) is the preferred method of creating A records for the MTA. Refer to Cisco Network Registrar documentation for information on configuring and troubleshooting DDNS.

KDC

The KDC is responsible for authenticating MTAs. As such, it must check the MTA certificate, and provide its own certificates so that the MTA can authenticate the KDC. It also communicates with the DPE (the provisioning server) to validate that the MTA is provisioned on the network.

PacketCable Provisioning Server

The PacketCable provisioning server is responsible for communicating the location of the MTA configuration file to the MTA, and/or provisioning MTA parameters via SNMP. SNMPv3 is used for all communication between the MTA and the provisioning server. The keys used to initiate SNMPv3 communication are obtained by the MTA during its authentication phase with the KDC. Provisioning server functionality is provided by the DPE in a BAC architecture.

Call Management Server

The call management server (CMS) is essentially a soft switch, or call-agent, with additional PacketCable functionality to control, among other things, quality of service on a cable network. The MTA sends a network call signaling (NCS) restart in progress (RSIP) message to the CMS upon successful PacketCable provisioning.

Key Variables

This section describes the key variables required to provision an eMTA correctly.

- Certificates, page 6-3
- Scope-Selection Tags, page 6-4
- MTA Configuration File, page 6-4

Certificates

The MTA_Root.cer file contains the MTA root certificate (a certificate that is rooted in the official PacketCable MTA root).

You must know in advance what telephony root certificate is required for the MTAs you want to provision. Deployments in production networks use telephony certificates rooted in the PacketCable real root. There is also a PacketCable test root used in lab and testing environments.

The KDC certificates used by the KDC to authenticate itself to the MTA must be rooted in the same telephony root that is stored on the MTA (PacketCable real or test root). Most MTA vendors support test images that have Telnet and/or HTTP login capabilities so that you can determine which telephony root is enabled, and change the root used (in most cases, you can only select between the PacketCable real or test root).

The most common scenario has the KDC loaded with certificates (from the *BAC_home*/kdc/solaris/packetcable/certificates directory) as follows:

- CableLabs_Service_Provider_Root.cer
- Service_Provider.cer
- Local_System.cer
- KDC.cer
- MTA_Root.cer

The first four certificates comprise the telephony certificate chain. The MTA_Root.cer file contains the MTA root used by the KDC to validate the certificates sent by the MTA.

Note Refer to Using the PKCert.sh Tool, page 13-5, for information on installing and managing KDC certificates.

To determine if you are using PacketCable test root, open the CableLabs_Service_Provider_Root.cer file in Windows, and validate that the Subject OrgName entry is **O = CableLabs**, and/or check that the Subject Alternative name reads **CN=CABLELABS GENERATED TEST ROOT FOR EQUIPMENT TEST PURPOSES ONLY**.

The KDC certificate (KDC.cer) contains the realm name to use. The realm name that BAC (and the corresponding DNS zone) is configured to use must match this realm name. Additionally, the MTA configuration file realm org name must match the organization name as seen in the telephony root.

The KDC certificate has a corresponding private key that must be installed in the *BAC_home*/kdc/solaris directory. Usually it is named KDC_private_key.pkcs8 or KDC_private_key_proprietary.. When changing certificates, you must also change the private key.

Scope-Selection Tags

In most scenarios, BAC is involved in processing all DHCP requests from scopes with scope-selection tags that match selection criteria specified in the DHCP Criteria page of the BAC administrator user interface. Client class can also be used to tie scopes to BAC processing; ensure you make this association before you attempt to provision devices.

MTA Configuration File

The MTA configuration file contains the location of the CMS. Additionally, it must contain an entry for Realm Name. This value must match that of the certificate chain in use.

Certain table entries within the MTA configuration file are indexed by the realm name delivered to the MTA in Option 122. This realm name entry in the MTA configuration file must match that delivered in Option 122. For example, if **DEF.COM** was the realm name delivered in Option 122, MTA configuration file entries in the pktcMtaDevRealm table would be indexed with a suffix made up of the ASCII-coded character values (in dot delimited decimal format when using the Cisco Broadband Configurator) of the realm name, for example 68.69.70.46.67.79.77. There are many free ASCII conversion pages available on the web to make this conversion easier.

Troubleshooting Tools

The 25 eMTA Secure provisioning steps contained in the PacketCable MTA Device Provisioning Specification are shown in Figure 5-1. This section describes:

- Logs, page 6-5
- Ethereal, SnifferPro, or Other Packet Capture Tools, page 6-5

Logs

These log files are used to maintain the following information:

- The Network Registrar has two logs (name_dhcp_1_log and name_dns_1_log), which contain the most recent logging entries from Network Registrar. Look in these files for DHCP- or DNS-related problems.
- The *BAC_home*/kdc/logs/kdc.log file shows all KDC interactions with MTAs, and KDC interactions with the DPE.
- The *BAC_data*/dpe/logs/dpe.log file shows the major steps related to SNMPv3 interaction with the MTA. You can also use the **show log** CLI command if you are working with the hardware DPE.

Note Turning on the tracing of snmp, registration server and registration server detail messages, using the command line interface, helps to troubleshoot potential PacketCable problems. For information on the appropriate troubleshooting commands, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

Ethereal, SnifferPro, or Other Packet Capture Tools

A packet capture tool is indispensable when troubleshooting the eMTAs. The Ethereal version, as packaged by CableLabs, includes numerous packet decoders specific to PacketCable. These include the Kerberos AS and AP packets.

- If you suspect that a specific failure is DHCP-related, capture packets while filtering on packets sourced from, or destined to, the CMTS cable interface IP address and the DHCP server IP address.
- If you suspect that a specific failure is related to any of the 25 steps occurring after DHCP, filter all packets to and from the eMTA IP address. This provides a very concise, easy-to-follow trace of provisioning steps 5 through 25, as shown in Figure 5-1.

Troubleshooting Scenarios

The scenarios listed in Table 6-1 are possible failures involving eMTAs.

Table 6-1 Troubleshooting Scenarios

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The KDC does not start.	The KDC certificate does not correspond to the private key.	Ensure that you have matching certificates and private key.
	The KDC license expired or is missing.	Restore KDC license to *BPR_HOME/kdc* directory.

Table 6-1 Troubleshooting Scenarios (continued)

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The MTA device does not appear in the BAC Devices page.	An incorrect cable helper address may have been configured.	Fix the helper address.
	The scope-selection tags do not match the DHCP criteria selected in the BAC user interface.	Verify that the MTA scope-selection tags match those in the PacketCable DHCP criteria created, in BAC, for the relevant MTAs.
	The Network Registrar extension point is not properly installed.	Re-install the Network Registrar extension point. Refer to the *Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1*.
	The cable modem portion did not receive Option 122.	Verify that the tags on the scope of the cable modem portion match the DOCSIS DHCP criteria configure for BAC.
The MTA device does not accept the DHCP offer and continually cycles through the DHCP flow.	There are invalid DHCP options configured.	Check that scope policy includes DNS server option, and/or check that the *cnr_ep.properties* file includes entries for primary and secondary DNS servers.
	The DHCP offer may have come from a DHCP server different from the one indicated in the cable modem portion's Option 122 suboption 1.	Check the *cnr_ep.properties* file to ensure that the main and backup DHCP servers are set correctly.
Both the *kdc.log* file and the ethereal trace indicate that the MTA device never contacts the KDC.	An incorrect DNS server is specified in the *cnr_ep.properties* file or the MTA scope policy, or both.	Check or correct *cnr_ep.properties* DNS servers.
	A zone is missing or has been incorrectly set up for the Kerberos realm.	Make sure a zone with same name as realm is created and contains an 'SRV' record of format '_kerberos._udp 0 0 88 *KDC FQDN*'.
	There is a missing or incorrect KDC 'A' record entry.	Ensure that an 'A' record exists for the FQDN contained in the Kerberos zone's 'SRV' record.
	The DPE FQDN cannot be resolved.	Ensure that the provFQDNs entry in *dpe.properties* has the correct FQDN and IP of the DPE.

Table 6-1 Troubleshooting Scenarios (continued)

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The KDC reports failure during the Kerberos AS-Request.	The MTA certificate does not match the MTA root used by KDC.	Verify that the *MTA_Root.cer* is correct by comparing the *MTA_Root.cer* against that used on a working system. If it is correct, the MTA itself could have a certificate problem. This situation is extremely rare and if this is the case, contact the MTA manufacturer.
	FQDN lookup by KDC to Prov Server failed. The device may not yet be provisioned in BAC.	Verify that the device appears. It should be given both a Class of Service and a DHCP criteria.
	A clock skew error. See PacketCable Checklists, page 3-6, for additional information.	Ensure that all BAC network elements are clock-synced via NTP. Refer to the *Broadband Access Center DPE CLI Reference, 2.7.1*.
	A mismatch may exist between the KDC and the DPE. **Note** If other devices are provisioned correctly, this is probably not the cause of the problem.	Check that these three entries exist in the *BPR_HOME/kdc/solaris/keys* directory: • mtafqdnmap,dpe.abc.com@DEF.COM • mtaprovsrvr,dpe.abc.com@DEF.COM • krbtgt,DEF.COM@DEF.COM The DPE FQDN and realm name on your system will be different from this example. Contents of these entries must match the entry in either the dpe.properties ‘KDCServiceKey’ entry, or the keys generated using the KeyGen utility.
The KDC reports success at the AS-Request/Reply (steps 9 and 10 in Figure 5-1) but the MTA device never moves past step 9.	There is a certificate mismatch between the telephony root loaded or enabled on the MTA, and that loaded on the KDC.	Check certificates on MTA and KDC.
	Although highly unlikely, it is possible that there is a corrupted telephony certificate chain. **Note** If other devices are provisioned correctly, this is not the cause of the problem.	Ensure that the correct certificate is loaded or enabled on MTA. If no devices can be provisioned correctly, try a different certificate on the KDC.
Failure at AP Request/Reply (step 14 in Figure 5-1).	A clock skew error. See PacketCable Checklists, page 3-6, for additional information.	Ensure that all BAC network elements are clock-synced via NTP. Refer to the *Broadband Access Center DPE CLI Reference*.
	Cannot resolve Prov Server FQDN.	Make sure that the provisioning server (DPE) has a correct DNS entry. Ensure that dpe.properties provFQDNs entry has the correct FQDN and IP of the provisioning server (DPE).
	There is no route from the MTA to the DPE.	Correct the routing problem.
The MTA device never issues a TFTP request for a configuration file.	There is no route to the TFTP server running on the DPE.	Correct the routing problem.

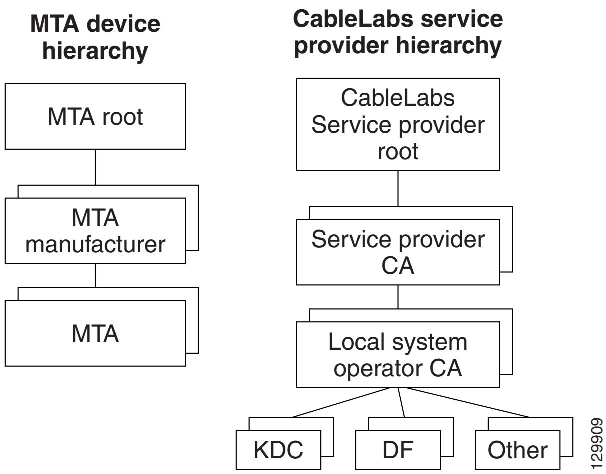
Table 6-1 Troubleshooting Scenarios (continued)

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The MTA device never receives the TFTP configuration file.	The configuration file is not cached at the DPE.	Wait until the next provisioning attempt, at which time the file should be cached. If this fails, reset the MTA.
	A conflicting TFTP server option is included in the network registrar MTA scope policy.	Since BAC inserts the DPE address for the TFTP server, you can safely remove this option from the policy.
The MTA device receives a configuration file but the DPE fails to receive the SNMP Inform (step 25 in Figure 5-1) as seen in the *dpe.log* file.	One of: • An internal conflict in the configuration file. • A conflict with Realm origin of the telephony certificate chain. • A conflict with the Realm Name provided in Option 122.	Ensure that the MTA configuration file is consistent.
The MTA device reports success (step 25 in Figure 5-1) although an RSIP is not sent.	The MTA cannot resolve the IP address of the CMS FQDN given in the MTA configuration file.	Verify that a DNS entry exists for the CMS.
	The MTA cannot reach the IP address(es) of the CMS. This is an indication that no route is configured.	Resolve all routing problems.
The MTA device reports success (step 25 in Figure 5-1), although it proceeds to contact the KDC again for CMS service.	The MTA configuration file points to an incorrect cable modem.	Correct the configuration file, or reconfigure the Cisco BTS 10200 to use the FQDN listed in the configuration file.
	The MTA configuration file has its pktcMtaDevCmsIPsecCtrl value missing, or it is set to 1. This means it will perform secure NCS call signaling, or it will use an ASCII suffix that does not match that of the CMS FQDN.	Correct the configuration file. If you intend to perform secure signaling, take the necessary steps to configure the KDC and the BTS for support.
The MTA device reports success (step 25 in Figure 5-1), RSIPs, but gets no response or gets an error in response from the soft switch.	The MTA is unprovisioned or has been incorrectly provisioned on the Cisco BTS 10200.	Provision MTA on the Cisco BTS 10200.
	An eMTA DNS entry does not exist.	Place an entry in the correct DNS zone for the eMTA. Dynamic DNS is the preferred method. Refer to Cisco Network Registrar documentation for information on enabling DDNS.

Certificate Trust Hierarchy

There are two certificate hierarchies affiliated with BAC PacketCable, the MTA Device Certificate Hierarchy and the CableLabs Service Provider Certificate Hierarchy, as shown in Figure 6-1.

Figure 6-1 PacketCable Certificate Hierarchy



Before implementing PacketCable in BAC, you should thoroughly familiarize yourself with these technology documents:

- *RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile*
- *DOCSIS Baseline Privacy Plus Interface Specification, SP-BPI+-I11-040407, April 7, 2004*

Note While Euro PacketCable uses the security specifications from PacketCable [PKT-SP-SEC-I08-030415], some changes are needed in relation to the digital certificates that are used in a Euro PacketCable environment. To keep Euro PacketCable and PacketCable as alike as possible, Euro PacketCable uses all PacketCable security technology, including new revision of the security specifications [PKTSP-SEC-I08-030415].

The elements of the Euro PacketCable certificates that are different from the PacketCable certificates are indicated in the tables below.

For Euro PacketCable, the Euro PacketCable certificates are the only valid certificates, any requirements that are stated in [PKT-SP-SEC-I08-030415] for PacketCable which refer to PacketCable Certificates are changed to the corresponding requirements for the Euro PacketCable certificates.

Euro PacketCable-compliant eMTAs must have the Euro-DOCSIS root CVC CA's public key stored in the CM's nonvolatile memory instead of the DOCSIS CVC CA's public key. Euro PacketCable-compliant standalone MTAs must have the tComLabs CVC Root Certificate and the tComLabs CVC CA certificate stored in nonvolatile memory. The CVC of manufacturers are verified by checking the certificate chain.

Certificate Validation

PacketCable certificate validation in general involves validation of an entire chain of certificates. For example, when the Provisioning Server validates an MTA Device certificate, the following chain of certificates is validated:

MTA Root Certificate + MTA Manufacturer Certificate + MTA Device Certificate

The signature on the MTA Manufacturer Certificate is verified with the MTA Root Certificate and the signature on the MTA Device Certificate is verified with the MTA Manufacturer Certificate. The MTA Root Certificate is self-signed and is known in advance to the Provisioning Server. The public key present in the MTA Root Certificate is used to validate the signature on this same certificate.

Usually the first certificate in the chain is not explicitly included in the certificate chain that is sent over the wire. In the cases where the first certificate is explicitly included it must already be known to the verifying party ahead of time and must *not* contain any changes to the certificate with the possible exception of the certificate serial number, validity period, and the value of the signature. If changes other than these exist in the CableLabs Service Provider Root certificate that was passed over the wire in comparison to the known CableLabs Service Provider Root certificate, the device making the comparison must fail the certificate verification.

The exact rules for certificate chain validation must fully comply with RFC 2459, where they are referred to as Certificate Path Validation. In general, X.509 certificates support a liberal 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 2459 recommends that certificate authorities restrict the encoding of name fields so that an implementation can declare a match or mismatch using simple binary comparison.

PacketCable security follows this recommendation. Accordingly, the DER-encoded tbsCertificate.issuer field of a PacketCable 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.

The sections below specify the required certificate chain, which must be used to verify each certificate that appears at the leaf node (at the bottom) in the PacketCable certificate trust hierarchy illustrated in Figure 6-1.

Validity period nesting is not checked and intentionally not enforced. Thus, the validity period of a certificate need not fall within the validity period of the certificate that issued it.

MTA Device Certificate Hierarchy

The device certificate hierarchy exactly mirrors that of the DOCSIS1.1/BPI+ hierarchy. It is rooted at a CableLabs-issued PacketCable MTA Root certificate, which is used as the issuing certificate of a set of manufacturer certificates. The manufacturer certificates are used to sign the individual device certificates.

The information contained in the following tables contains the PacketCable-specific values for the required fields according to RFC 2459. These PacketCable specific values must be followed according to Table 6-2, except that Validity Periods should be as given in the respective tables. If a required field is not specifically listed for PacketCable, then follow the guidelines in RFC 2459.

MTA Root Certificate

This certificate must be verified as part of a certificate chain containing the MTA Root Certificate, MTA Manufacturer Certificate, and the MTA Device Certificate.

Table 6-2 lists the values relevant to the MTA Root Certificate.

Table 6-2 MTA Root Certificate

MTA Root Certificate		
Subject Name Form	**PacketCable** C=US O=CableLabs OU=PacketCable CN=PacketCable Root Device Certificate Authority	**Euro PacketCable** C=BE O=tComLabs OU=Euro-PacketCable CN=Euro-PacketCable Root Device Certificate Authority
Intended Usage	This certificate is used to sign MTA Manufacturer Certificates and is used by the KDC. This certificate is not used by the MTAs and thus does not appear in the MTA MIB.	
Signed By	Self-signed	
Validity Period	20+ years. It is intended that the validity period is long enough that this certificate is never reissued.	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true, pathLenConstraint=1)	

MTA Manufacturer Certificate

This certificate must be verified as part of a certificate chain containing the MTA Root Certificate, MTA Manufacturer Certificate, and the MTA Device Certificate. The state/province, city and manufacturer's facility are optional attributes. A manufacturer may have more than one manufacturer's certificate, and there may exist one or more certificates per manufacturer. All certificates for the same manufacturer may be provided to each MTA either at manufacture time or during a field update. The MTA must select an appropriate certificate for its use by matching the issuer name in the MTA Device Certificate with the subject name in the MTA Manufacturer Certificate. If present, the authorityKeyIdentifier of the device certificate must match the subjectKeyIdentifier of the manufacturer certificate as described in RFC 2459. The CompanyName field that is present in O and CN may be different in the two instances.

Table 6-3 lists the values relevant to the MTA Manufacturer Certificate.

Table 6-3 MTA Manufacturer Certificates

MTA Manufacturer Certificate		
Subject Name Form	**PacketCable** C=US O=CableLabs OU=PacketCable CN=PacketCable Root Device Certificate Authority	**Euro PacketCable** C=*Country of Manufacturer* O=*Company Name* [stateOrProvinceName = *State/Province*] [localityName=*City*] OU=Euro-PacketCable [organizationalUnitName= *Manufacturing Location*] CN=*Company Name* Euro-PacketCable CA
Intended Usage	This certificate is issued to each MTA manufacturer and can be provided to each MTA as part of the secure code download as specified by the PacketCable Security Specification (either at manufacture time, or during a field update). This certificate appears as a read-only parameter in the MTA MIB. This certificate along with the MTA Device Certificate is used to authenticate the MTA device identity (MAC address) during authentication by the KDC.	
Signed By	MTA Root Certificate CA	
Validity Period	20 years	
Modulus Length	2048	
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)	

MTA Device Certificate

This certificate must be verified as part of a certificate chain containing the MTA Root Certificate, MTA Manufacturer Certificate and the MTA Device Certificate. The state/province, city and manufacturer's facility are optional attributes. The MAC address must be expressed as six pairs of hexadecimal digits separated by colons, for example, "00:60:21:A5:0A:23". The alpha hexadecimal characters (A-F) must be expressed as uppercase letters. The MTA device certificate should not be replaced or renewed.

Table 6-4 lists the values relevant to the MTA Device Certificate.

Table 6-4 MTA Device Certificates

MTA Device Certificate		
Subject Name Form	**PacketCable** C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*], OU=PacketCable [OU=*Product Name*] [OU=*Manufacturer's Facility*] CN=*MAC Address*	**Euro PacketCable** C=*Country of Manufacturer* O=*Company Name* [ST=*State/Province*] [L=*City*] OU=Euro-PacketCable [OU=*Product Name*] [OU=*Manufacturing Location*] CN=*MAC Address*
Intended Usage	This certificate is issued by the MTA manufacturer and installed in the factory. The provisioning server cannot update this certificate. This certificate appears as a read-only parameter in the MTA MIB. This certificate is used to authenticate the MTA device identity (MAC address) during provisioning.	
Signed By	MTA Manufacturer Certificate CA	
Validity Period	At least 20 years	
Modulus Length	1024, 1536 or 2048	
Extensions	keyUsage[c,o](digitalSignature, keyEncipherment) authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*)	

MTA Manufacturer Code Verification Certificates

Code Verification Certificate (CVC) specification for eMTAs must be identical to the DOCSIS 1.1 CVC, specified in DOCSIS specification SP-BPI+-I11-040407.

CableLabs Service Provider Certificate Hierarchy

The Service Provider Certificate Hierarchy is rooted at a CableLabs issued CableLabs Service Provider Root certificate. That certificate is used as the issuing certificate of a set of service provider's certificates. The service provider's certificates are used to sign an optional local system certificate. If the local system certificate exists then that is used to sign the ancillary equipment certificates, otherwise the ancillary certificates are signed by the Service Provider's CA.

The information contained in Table 6-5 contains the specific values for the required fields according to RFC 2459. These specific values must be followed. If a required field is not specifically listed then the guidelines in RFC 2459 must be followed exactly.

CableLabs Service Provider Root Certificate

Before any Kerberos key management can be performed, an MTA and a KDC need to perform mutual authentication using the PKINIT extension to the Kerberos protocol. An MTA authenticates a KDC after it receives a PKINIT Reply message containing a KDC certificate chain. In authenticating the KDC, the MTA verifies the KDC certificate chain, including KDC's Service Provider Certificate signed by the CableLabs Service Provider Root CA.

Table 6-5 lists the values relevant to the CableLabs Service Provider Root Certificate.

Table 6-5 CableLabs Service Provider Root Certificates

CableLabs Service Provider Root Certificate		
Subject Name Form	**PacketCable** C=US O=CableLabs CN=CableLabs Service Provider Root CA	**Euro PacketCable** C=BE O=tComLabs CN=tComLabs Service Provider Root CA
Intended Usage	This certificate is used to sign Service Provider CA certificates. This certificate is installed into each MTA at the time of manufacture or with a secure code download as specified by the PacketCable Security Specification and cannot be updated by the Provisioning Server. Neither this root certificate nor the corresponding public key appears in the MTA MIB.	
Signed By	Self-signed	
Validity Period	20+ years. It is intended that the validity period is long enough that this certificate is never reissued.	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true)	

Service Provider CA Certificate

This is the certificate held by the service provider, signed by the CableLabs Service Provider Root CA. It is verified as part of a certificate chain that includes the CableLabs Service Provider Root Certificate, Telephony Service Provider Certificate, optional Local System Certificate and an end-entity server certificate. The authenticating entities normally already possess the CableLabs Service Provider Root Certificate and it is not transmitted with the rest of the certificate chain.

The fact that a Service Provider CA Certificate is always explicitly included in the certificate chain allows a Service Provider the flexibility to change its certificate without requiring reconfiguration of each entity that validates this certificate chain (for example, MTA validating a PKINIT Reply). Each time the Service Provider CA Certificate changes, its signature must be verified with the CableLabs Service Provider Root Certificate. However, a new certificate for the same Service Provider must preserve the same value of the OrganizationName attribute in the SubjectName. The *Company* field that is present in O and CN may be different in the two instances.

Table 6-6 lists the values relevant to the CableLabs Service Provider CA Certificate.

Table 6-6 CableLabs Service Provider CA Certificates

CableLabs Service Provider Root Certificate		
Subject Name Form	**PacketCable** C=*Country* O=*Company* CN=*Company* CableLabs Service Provider CA	**Euro PacketCable** C=*Country* O=*Company* CN=*Company* tComLabs Service Provider CA
Intended Usage	This certificate is used to sign Service Provider CA certificates. This certificate is installed into each MTA at the time of manufacture or with a secure code download as specified by the PacketCable Security Specification and cannot be updated by the Provisioning Server. Neither this root certificate nor the corresponding public key appears in the MTA MIB.	
Signed By	Self-signed	
Validity Period	20+ years. It is intended that the validity period is long enough that this certificate is never reissued.	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign cRLSign), subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true)	

Local System CA Certificates

A Service Provider CA may delegate the issuance of certificates to a regional Certification Authority called Local System CA (with the corresponding Local System Certificate). Network servers are allowed to move freely between regional Certification Authorities of the same Service Provider. Therefore, the MTA MIB does not contain any information regarding a Local System Certificate (which might restrict an MTA to KDCs within a particular region).

Table 6-7 lists the values relevant to the Local System CA Certificate.

Table 6-7 Local System CA Certificates

Local System CA Certificate		
Subject Name Form	**PacketCable** C=*Country* O=*Company* OU=*Local System Name* CN=*Company* CableLabs Local System CA	**Euro PacketCable** C=*Country* O=*Company* OU=*Local System Name* CN=*Company* tComLabs Local System CA
Intended Usage	A Service Provider CA may delegate the issuance of certificates to a regional Certification Authority called Local System CA (with the corresponding Local System Certificate). Network servers are allowed to move freely between regional Certification Authorities of the same Service Provider. Therefore, the MTA MIB does not contain any information regarding a Local System Certificate (which might restrict an MTA to KDCs within a particular region).	

Table 6-7 Local System CA Certificates (continued)

Local System CA Certificate	
Signed By	Service Provider CA Certificate
Validity Period	20 years.
Modulus Length	1024, 1536, 2048
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)

Operational Ancillary Certificates

All these are signed by the either the Local System CA or by the Service Provider CA. Other ancillary certificates may be added to this standard at a later time.

KDC Certificate

This certificate must be verified as part of a certificate chain containing the CableLabs Service Provider Root Certificate, Service Provider CA Certificate and the Ancillary Device Certificates. The PKINIT specification requires the KDC certificate to include the subjectAltName v.3 certificate extension, the value of which must be the Kerberos principal name of the KDC.

Table 6-8 lists the values relevant to the KDC Certificate.

Table 6-8 KDC Certificates

Key Distribution Center Certificate		
Subject Name Form	**PacketCable** C=*Country* O=*Company*, [OU=*Local System Name*] OU= CableLabs Key Distribution Center CN=*DNS Name*	**Euro PacketCable** C=*Country* O=*Company* [OU=*Local System Name*] OU=tComLabs Key Distribution Center CN=*DNS Name*
Intended Usage	To authenticate the identity of the KDC server to the MTA during PKINIT exchanges. This certificate is passed to the MTA inside the PKINIT replies and is therefore not included in the MTA MIB and cannot be updated or queried by the Provisioning Server.	
Signed By	Service Provider CA Certificate or Local System Certificate	
Validity Period	20 years	
Modulus Length	1024, 1536 or 2048	
Extensions	keyUsage[c,o](digitalSignature)authorityKeyIdentifier[n,m](keyIdentifier= *subjectKeyIdentifier value from CA certificate*)subjectAltName[n,m]	

Delivery Function (DF)

This certificate must be verified as part of a certificate chain containing the CableLabs Service Provider Root Certificate, Service Provider CA Certificate and the Ancillary Device Certificates. This certificate is used to sign phase 1 IKE intra-domain exchanges between DFs (which are used in Electronic Surveillance). Although Local System Name is optional, it is required when the Local System CA signs this certificate. The IP address must be specified in standard dotted-quad notation, for example, 245.120.75.22.

Table 6-9 lists the values relevant to the DF Certificate.

Table 6-9 DF Certificates

DF Certificate		
Subject Name Form	**PacketCable** C=*Country* O=*Company* [OU=*Local System Name*] OU=PacketCable Electronic Surveillance CN=*IP address*	**Euro PacketCable** C=*Country* O=*Company* [OU=*Local System Name*] OU=Euro-PacketCable Electronic Surveillance CNe=*IP address*
Intended Usage	To authenticate IKE key management, used to establish IPsec Security Associations between pairs of DFs. These Security Associations are used when a subject that is being legally wiretapped forwards the call and event messages containing call info have to be forwarded to a new wiretap server (DF).	
Signed By	Service Provider CA Certificate or Local System CA Certificate	
Validity Period	20 years	
Modulus Length	2048	
Extensions	keyUsage[c,o](digitalSignature) authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA certificate*) subjectAltName[n,m] (dNSName=*DNSName*)	

PacketCable Server Certificates

These certificates must be verified as part of a certificate chain containing the CableLabs Service Provider Root Certificate, Service Provider Certificate, Local System Operator Certificate (if used) and the Ancillary Device Certificates. These certificates are used to identify various servers in the PacketCable system. For example, they may be used to sign phase 1 IKE exchanges or to authenticate a PKINIT exchange. Although the Local System Name is optional, it is REQUIRED when the Local System CA signs this certificate. 2IP address values must be specified in standard dotted decimal notation, for example, 245.120.75.22. DNS Name values must be specified as a fully qualified domain name (FQDN), for example, device.packetcable.com.

Table 6-10 lists the values relevant to the PacketCable Server Certificate.

Table 6-10 PacketCable Server Certificates

PacketCable Server Certificates		
Subject Name Form	**PacketCable** C=*Country* O=*Company* OU=PacketCable OU=[*Local System Name*] OU=*Sub-System Name* CN=*Server Identifier*[:*Element ID*] The value of *Server Identifier* must be the server's FQDN or its IP address, optionally followed by a colon (:) and an Element ID with no white space before or after the colon. *Element ID* is the identifier that appears in billing event messages. It must be included in the certificate of every server that is capable of generating event messages. This includes a CMS, CMTS and MGC. [8] defines the Element ID as an 5-octet right-justified space-padded ASCII-encoded numerical string. When converting the Element ID for use in a certificate, spaces must be converted to ASCII zeroes (0x48). For example, a CMTS with Element ID 311 and IP address 123.210.234.12 will have a common name "123.210.234.12: 00311". The value of *Sub-System Name* must be one of the following: • For Border Proxy: bp • For Cable Modem Termination System: cmts • For Call Management Server: cms • For Media Gateway: mg•For Media Gateway Controller: mgc • For Media Player: mp • For Media Player Controller: mpc • For Provisioning Server: ps • For Record Keeping Server: rks • For Signaling Gateway: sg	**Euro PacketCable** C=*Country* O=*Company* OU=Euro-PacketCable [OU=*Local System Name*] OU=*Sub-system Name* CN=*Server Identifier*[:*Element ID*] Please refer to [PKT-SP-SEC-IO8-030415] for additional specifications on the commonName.
Intended Usage	These certificates are used to identify various servers in the PacketCable system. For example, they may be used to sign phase 1 IKE exchanges or to authenticate a device in a PKINIT exchange.	

Table 6-10 PacketCable Server Certificates (continued)

PacketCable Server Certificates	
Signed By	Telephony Service Provider Certificate or Local System Certificate
Validity Period	Set by MSO policy
Modulus Length	2048
Extensions	keyUsage[c,o](digitalSignaturekeyEncipherment) authorityKeyIdentifier[n,m](keyIdentifier=*subjectKeyIdentifier value from CA cert*) subjectAltName[n,m](dNSName=*DNSName* | iPAddress=*IP AddressName*) The keyUsage tag is optional. When it is used it must be marked as critical. Unless otherwise described below, the subjectAltName extension must include the corresponding name value as specified in the CN field of the subject.

The CN attribute value for CMS certificates must be the Element ID. The subjectAltName extension must include either the IP Address or the FDQN of the CMS. The CN attribute value for CMTS certificates must be the Element ID. The subjectAltName extension must include either the IP Address or the FDQN of the CMTS.

The CN attribute value for MGC certificates must be the Element ID. The subjectAltName extension must include either the IP Address or the FDQN of the MGC.

Certificate Revocation

Out of scope for PacketCable at this time.

Code Verification Certificate Hierarchy

The CableLabs Code Verification Certificate (CVC) 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 does not apply to eMTAs. Refer to section 11 for more information.

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 F4 (65537 decimal).

CableLabs Code Verification Root CA Certificate

This certificate must be validated as part of the certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA, and the Code Verification Certificates. See Certificate Validation, page 6-10, for additional information on how to validate certificates.

Table 6-11 lists the values relevant to the CableLabs Code Verification Root CA Certificate.

Table 6-11 CableLabs Code Verification Root CA Certificates

CableLabs Code Verification Root CA Certificate		
Subject Name Form	**PacketCable** C=US O=CableLabs CN=CableLabs CVC Root CA	**Euro PacketCable** C = BE O = tComLabs CN = tComLabs CVC Root CA
Intended Usage	This certificate is used to sign Code Verification CA Certificates. This certificate must be included in the S-MTAs nonvolatile memory at manufacture time.	
Signed By	Self-signed	
Validity Period	20+ years	
Modulus Length	2048	
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectkeyidentifier [n,m] basicConstraints [c,m](cA=true)	

CableLabs Code Verification CA Certificate

The CableLabs Code Verification CA Certificate must be validated as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, CableLabs Code Verification CA Certificate and the Code Verification Certificate. See Certificate Validation, page 6-10, for additional information on how to validate certificates. There may be more than one CableLabs Code Verification CA. A S-MTA must support one CableLabs CVC CA at a time.

Table 6-12 lists the values relevant to the CableLabs Code Verification CA Certificate.

Table 6-12 CableLabs Code Verification CA Certificates

CableLabs Code Verification CA Certificate		
Subject Name Form	**PacketCable** C=US O=CableLabs CN=CableLabs CVC CA	**Euro PacketCable** C = BE O = tComLabs CN = tComLabs CVC CA
Intended Usage	This certificate is issued to CableLabs by the CableLabs Code Verification Root CA. This certificate issues Code Verification Certificates. This certificate must be included in the S-MTAs nonvolatile memory at manufacture time.	
Signed By	CableLabs Code Verification Root CA	
Validity Period	Set by CableLabs policy	

Table 6-12 CableLabs Code Verification CA Certificates (continued)

CableLabs Code Verification CA Certificate	
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectKeyIdentifier [n,m] authorityKeyIdentifier [n,m] basicConstraints [c,m](cA=true, pathLenConstraint=0)

Manufacturer Code Verification Certificate

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.

Table 6-13 lists the values relevant to the Manufacturer Code Verification Certificate.

Table 6-13 Manufacturer Code Verification Certificates

Manufacturer Code Verification Certificate		
Subject Name Form	**PacketCable** C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*] CN=*Company Name* Mfg CVC	**Euro PacketCable** C=*Country* O=*Company Name* [ST=*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	tComLabs Code Verification CA Certificate
Validity Period	Set by CableLabs policy	
Modulus Length	1024, 1536, 2048	
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m]	

The Company Name in the Organization may be different than the Company Name in the Common Name.

Service Provider Code Verification Certificate

The Service Provider Code Verification Certificate must be validated as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA Certificate, and the Service Provider Code Verification Certificate. Refer to Certificate Validation, page 6-10, for additional information on how to validate certificates.

Table 6-14 lists the values relevant to the Service Provider Code Verification Certificate.

Table 6-14 Service Provider Code Verification Certificates

Service Provider Code Verification Certificate		
Subject Name Form	C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*] CN=*Company Name* Service Provider CVC	C=*Country* O=*Company Name* [ST=*State/Province*] [L=*City*] CN=*Company Name* Service Provider CVC
Intended Usage	The CableLabs Code Verification CA issues this certificate to each authorized Service Provider. It is used in the policy set by the cable operator for secure software download.	
Signed By	CableLabs Code Verification CA	tComLabs Code Verification CA Certificate
Validity Period	Set by CableLabs policy	
Modulus Length	1024, 1536, 2048	
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m]	

The Company Name in the Organization may be different than the Company Name in the Common Name.

Certificate Revocation Lists for CVCs

The S-MTA is not required to support Certificate Revocation Lists (CRLs) for CVCs.