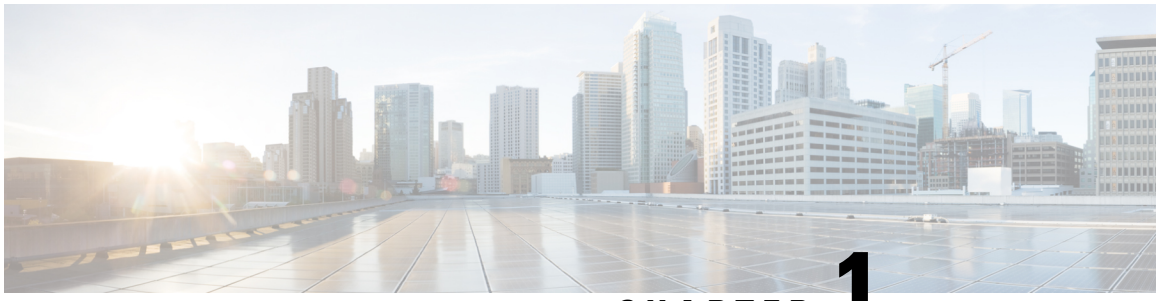
Cisco Expressway Certificate Creation and Use Deployment Guide (X14.3)

First Published: 2023-03-12

Last Modified: 2023-06-16

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

About This Guide

Important New features in software version X12.5 and later are not supported for the Cisco TelePresence Video Communication Server (VCS) product. They apply only to the Cisco Expressway Series (Expressway) product. This software version is provided to VCS for maintenance and bug fixing purposes only.

From version X12.5 onwards, this guide applies only to the Cisco Expressway Series (Expressway) product and no longer applies to the Cisco TelePresence Video Communication Server (VCS) product. Older VCS guides on Cisco.com are still valid for the VCS versions they apply to—as specified on the title page of each guide.

This deployment guide provides instructions on how to create X.509 cryptographic certificates for use with the Cisco Expressway (Expressway), and how to load them into Expressway.

This chapter explains the following:

Change History

The following table describes the information added or changed in the product.

Table 1: Change History

Release Date	Change	Reason
April 2023	Included a new sectio "Certificate Manager ECDSA Support". Updated sections "Generating a CSR" and "Creating a CSR Using Expressway".	X14.3 release
June 2020	Removed biased language from the "PKI Introduction" section.	Document correction

Release Date	Change	Reason
June 2020	Updated for X12.6	X12.6 release
February 2020	Updated the "Create a CSR using Expressway" section regarding multi-SAN certificate.	Document correction
December 2019	Updated prerequisites for deploying ACME certificate service.	Document correction
April 2019	Updates for maintenance release X12.5.2.	X12.5.2 release
January 2019	Updated for X12.5 for ACME certificate management. Other minor corrections.	X12.5 release
September 2018	Updated software version from X8.11 to X8.11.1, as version X8.11 is no longer available.	X8.11.1 release
July 2018	Updated for X8.11.	X8.11 release (withdrawn)
September 2017	Remove 999 character SAN limitation.	Fixed in X8.10 release
July 2017	Description of new warning messages for server certificate upload added. Changed UI menu path. Combined VCS and Expressway versions of document.	X8.10 release
December 2016	Clarified requirements for MRA certificates.	X8.9 release
June 2016	Updated for X8.8.	X8.8 release
November 2015	New template applied. Republished for X8.7.	
July 2015	Updated for X8.6.	
April 2015	Update for X8.5.2. Changes to CRL information, CSR generation page defaults, 999 character limit on SANs.	
January 2015	Update for X8.5.1. Introduced an option on the user interface to select the Digest algorithm . The default is set to SHA-256 (hash algorithm).	
December 2014	Re-issued for X8.5. Notes inserted over 2050 date management, and unsupported OIDs. Changed instructions in Appendix 2 "Creating a certificate request using OpenSSL".	
July 2014	Re-issued for X8.2. Recommended options changed for server certificate in Unified Communications deployments.	
June 2014	Republished for X8.2. Enhanced the server certificate requirements for Unified Communications deployments.	

Release Date	Change	Reason
December 2013	Initial release of Expressway version. (Compared to previous, VCS-only version) Updated for X8.1. Removed "Certificate generation using Microsoft OCS" appendix. Various improvements and clarifications to "Certificate generation using OpenSSL only" appendix.	

Information Not Covered in this Guide

This document does not cover the following Expressway configuration topics, which are instead covered in the *Expressway Administrator Guide*:

- How to enable certificate-based authentication on Expressway
- Details of root CAs pre-installed in Expressway
- How to configure minimum TLS versions and cipher suites
- How to test client certificates
- Managing mTLS certificates (Mobile and Remote Access deployments)
- Domain certificates and Server Name Indication for multitenancy (Hosted Collaboration Solution deployments)

PKI Introduction

Public Key Infrastructure (PKI) provides the mechanisms through which you can secure communications (encrypted and integrity protected) and verify the identities. Underlying PKI is:

- **A public/private key pair:** a public key is used to encrypt data that is sent to a server, but only the private key (kept secret by the server) can be used to decrypt it.
- **Signatures of data:** server “signs” data using a combination of a cryptographic hash of the data and the server’s private key. A client can verify the signature using server’s public key and the same hash. This ensures that the data is sent from the expected server, and is not tampered with.
- **Certificates:** a certificate is a wrapper around a public key, and provides information about the owner of the key in X.509 format, and typically includes server name and contact details.
- **A certificate chain:** Certificate Authority (CA) signs a certificate using its own private key. In turn, you can verify a certificate as signed by checking the signature against the CA’s certificate (public key). Web browsers and other clients have a list of CA certificates that they trust, and can thus verify the certificates of individual servers.

Transport Layer Security (TLS) is the standard mechanism for securing a TCP connection between hosts on a TCP/IP network. For example, secure HTTP (HTTPS) uses TLS to encrypt and verify traffic. To establish a TLS connection:

1. The client sends its capabilities (including cipher suites) and a random number to make an initial TCP connection.
2. The server responds with its choice of those capabilities, another random number, and its certificate.
3. The client verifies that the server certificate is issued (signed) by a CA that it trusts, and it is not revoked.
4. The client sends a “pre secret”, encrypted with the server’s public key.
5. This pre secret, combined with the exchanged random numbers (to prevent replay attacks), is used to generate a “shared secret”. This shared secret keeps the remaining communications of this TLS session encrypted between the client and server.

The following sections describe how these PKI components can be used with the Expressway.

Certificate Use on the Expressway Overview

Expressway needs certificates for:

- Secure HTTP with TLS (HTTPS) connectivity
- TLS connectivity for SIP signaling, endpoints and neighbor zones
- Connections to other systems such as Unified CM, Cisco TMS, LDAP servers and syslog servers

It uses its list of trusted Certificate Authority (CA) certificates and associated certificate revocation lists (CRLs) to validate other devices connecting to it.

The Expressway uses the Server Certificate and the Private key to provide a signed certificate to provide evidence that the Expressway is the device it says it is. This can be used with neighboring devices such as Microsoft Lync or Unified CM, as well as administrators using the web interface.

A certificate identifies the Expressway. It contains names by which it is known and to which traffic is routed. If the Expressway is known by multiple names for these purposes, such as if it is part of a cluster, this must be represented in the X.509 subject data, according to the guidance of RFC5922. The certificate must contain the FQDN of both the Expressway itself and of the cluster. The following lists show what must be included in the X.509 subject, depending on the deployment model chosen.

If the Expressway is not clustered:

- Subject Common Name = FQDN of Expressway
- Subject Alternate Names = leave blank*

If the Expressway is clustered, with individual certificates per Expressway:

- Subject Common Name = FQDN of cluster
- Subject Alternate Name = FQDN of Expressway peer, FQDN of cluster*

You manage the Cisco Expressway's server certificate through the Server certificate page (**Maintenance** > **Security** > **Server certificate**). This certificate is used to identify the Expressway when it communicates with client systems using TLS encryption, and with web browsers over HTTPS. You can use the Server certificate page to:

- View details about the currently loaded certificate.

- Generate a certificate signing request.
- Upload a new server certificate.

Certificate Generation Overview

X.509 certificates may be supplied from a third party, or may be generated by a certificate generator such as OpenSSL or a tool available in applications such as Microsoft Certification Authority. Third-party certificates supplied by recognized certificate authorities are recommended, although Expressway deployments in controlled or test environments can use internally generated certificates.

The Expressway also supports the Automated Certificate Management Environment (ACME), and you can configure it to automatically request and deploy certificates signed by the *Let's Encrypt*® certificate authority.

Earlier releases of Cisco Expressway supported RSA certificates only. However, Cisco Expressway X14.3 release onwards, Elliptic Curve Digital Signature Algorithm (ECDSA) certificate has been added along with the existing RSA certificate.

The certificate manager supports the generation of ECDSA certificates with different key length values.

When you update or install Cisco Expressway, the self-signed certificate is generated.

Certificate generation is usually a 3-stage process:

- Stage 1: generate a private key
- Stage 2: create a certificate request
- Stage 3: authorize and create the certificate

This document presents alternative methods of generating the root certificate, client/server certificate for the Expressway, and private key:

- Generating a Certificate Signing Request, describes how to use the Expressway itself to generate the private key and certificate request.
- Generate Certificate Using OpenSSL Only, documents the OpenSSL-only process, which could be used with a third party or internally managed CA.

For mutual TLS authentication the Expressway Server certificate must be capable of being used as a Client certificate as well, thus allowing the Expressway to authenticate as a client device to a neighboring server (see Enable AD CS to Issue Client and Server Certificates).

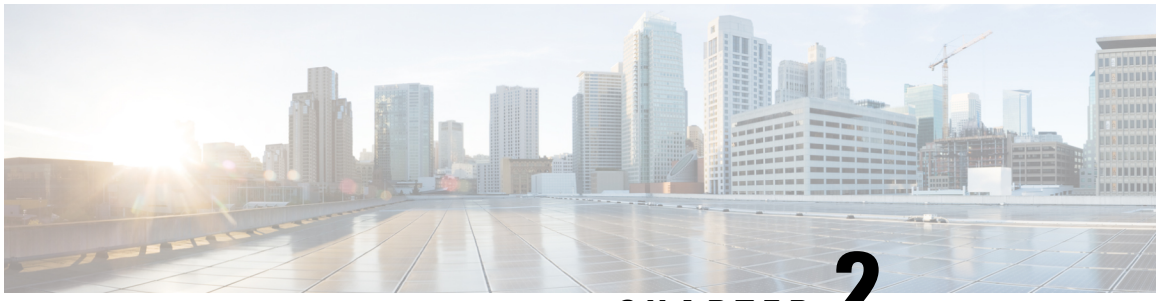
Points to be Aware

- When you generate a CSR using external systems, ensure that the CSR does not contain any unsupported OIDs. Currently, only the following Extended Validation OIDs are supported.
  - 1.3.6.1.4.1.311.60.2.1.1 jurisdictionOfIncorporationLocalityName
  - 1.3.6.1.4.1.311.60.2.1.2 jurisdictionOfIncorporationStateOrProvinceName
  - 1.3.6.1.4.1.311.60.2.1.3 jurisdictionOfIncorporationCountryName

For more information on how to verify if there are unsupported OIDs in the certificate, see the section Issues with SSH Failures and Unsupported OIDs.

.

- Wildcard certificates manage multiple subdomains and the services names they support. They can be less secure than SAN certificates and are not supported by Expressway.
- Changes are being introduced to the way that dates are handled from 2050, and certificates that have expiry dates beyond that can cause operational issues.
- The Expressway mechanism for CA certificate checking, requires the BasicConstraints extension to be present.
- We highly recommend using certificates based on RSA keys. Other types of certificate, such as those based on DSA keys, are not tested and may not work with the Expressway in all scenarios.
- Do not allow your server certificate to expire as this may cause other external systems to reject your certificate and prevent the Expressway from being able to connect to those systems.



CHAPTER 2

Generating a Certificate Signing Request

A Certificate Signing Request (CSR) contains the identity information about the owner of a private key. It can be passed to a third-party or internal certification authority for generating a signed certificate, or it can be used in conjunction with an application such as ACME, Microsoft Certification Authority, or OpenSSL. Now, Expressway supports generating CSR with Elliptic Curve Digital Signature Algorithm (ECDSA) or RSA based public key algorithm.

Note Generating a new server Certificate Signing Request (CSR) does not invalidate the existing active server certificate installed in Expressway.

This chapter explains the following:

Creating a CSR Using Expressway

The Expressway can generate server certificate signing requests. This removes the need to use an external mechanism to generate and obtain certificate requests.

To generate a CSR:

Procedure

Step 1 Go to **Maintenance > Security > Server certificate**.

Step 2 Click **Generate CSR** to go to the **Generate CSR** page.

Step 3 Enter the required properties for the certificate:

a. From the **Additional Information** section, select the **Public key algorithm**. Choose RSA or ECDSA from the drop-down.

b. Choose the desired **Key length** (in bits) from the drop-down based on the Public key algorithm.

Note The defined Key length (in bits) for ECDSA – 256, 384, 521 and RSA – 2048 and 4096

c. See Server Certificates and Clustered Systems, if your Expressway is part of a cluster.

d. See the "Server Certificates Requirements for Unified Communications" section, if this Expressway is part of a Unified Communications solution.

e. The certificate request includes, automatically, the public key that is used in the certificate and, the client and server authentication Enhanced Key Usage (EKU) extension.

Step 4 Click **Generate CSR**. The system produces a signing request and an associated private key. The private key is stored securely on the Expressway and cannot be viewed or downloaded. You must never disclose your private key, not even to the certificate authority.

Step 5 You are returned to the **Server certificate** page. From here you can:

a. **Download** the request to your local file system so that it can be sent to a certificate authority. You are prompted to save the file (the exact wording depends on your browser).

b. View the current request (click **Show (decoded)** to view it in a human-readable form, or click **Show (PEM file)** to view the file in its raw format).

c. Use ACME to manually or automatically submit the CSR to a CA that signs ACME certificates.

Note

- Only one signing request can be in progress at any point of time. This is because the Expressway has to keep track of the private key file associated with the current request. To discard the current request and start a new request, click **Discard CSR**.
- From version X8.5.1 the user interface provides an option to set the Digest algorithm. The default is set to SHA-256, with options to change to SHA-1, SHA-384, or SHA-512.
- From version X8.10, you cannot select SHA-1.
- The Issuer and Subject fields of certificates returned by Let's Encrypt do not include attributes like State, Country, and Organisation. The Expressway UI still requires you to complete these fields in the CSR, even though the authority ignores them.

You must now use CSR to generate a signed PEM certificate file. You can pass it to a third-party or internal certification authority, or use it in conjunction with an application such as Microsoft Certification Authority (see Authorize a Request and Generate a Certificate Using Microsoft Certification Authority) or OpenSSL (see Operate as a Certificate Authority Using OpenSSL).

If you have multiple entries or FQDNs in the SAN (such as for MRA deployments), ensure that you ask for a multi-domain / multi-SAN certificate from your certificate authority, not a single certificate. Some authorities do not suggest this option unless you specifically request it.

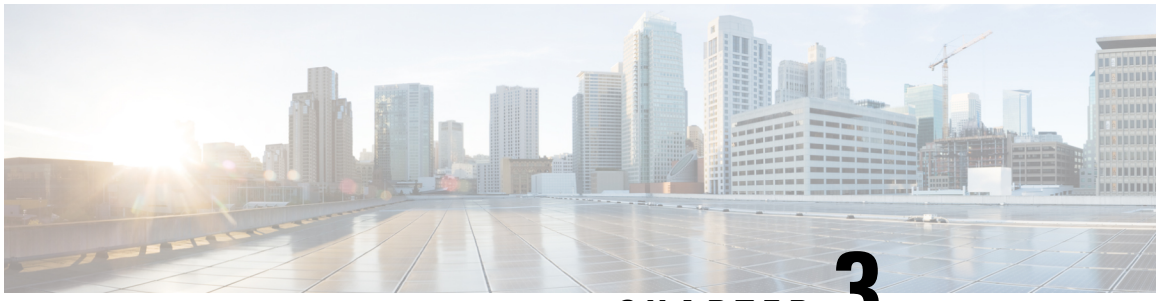
When the signed server certificate is received back from the certificate authority, upload it to the Expressway as described in Loading Certificates and Keys Onto Expressway.

Server Certificates and Clustered Systems

When a CSR is generated, a single request and private key combination is generated for that peer only.

If you have a cluster of Expressways, you must generate a separate signing request on each peer. Those requests must then be sent to the certificate authority and the returned server certificates uploaded to each relevant peer.

You must ensure that the correct server certificate is uploaded to the appropriate peer, otherwise the stored private key on each peer will not correspond to the uploaded certificate.



CHAPTER 3

Server Certificate Requirements for Unified Communications

This chapter explains the following:

Cisco Unified Communications Manager Certificates

Two Cisco Unified Communications Manager certificates are significant for Mobile and Remote Access:

- *CallManager* certificate
- *tomcat* certificate

These certificates are automatically installed on the Cisco Unified Communications Manager and by default they are self-signed and have the same common name (CN).

We recommend using CA-signed certificates. However, if you do use self-signed certificates, the two certificates must have different common names. The Expressway does not allow two self-signed certificates with the same CN. So if the *CallManager* and *tomcat* self-signed certificates have the same CN in the Expressway's trusted CA list, the Expressway can only trust one of them. This means that either secure HTTP or secure SIP, between Expressway-C and Cisco Unified Communications Manager, will fail.

Also, when generating tomcat certificate signing requests for any products in the Cisco Collaboration Systems Release 10.5.2, you need to be aware of CSCus47235. You need to work around this issue to ensure that the FQDNs of the nodes are in the certificates as Subject Alternative Name (SAN) entries. The *Expressway X8.5.3 Release Note* on the Release Notes page has details of the workarounds.

IM and Presence Service Certificates

Two IM and Presence Service certificates are significant if you use XMPP:

- *cup-xmpp* certificate
- *tomcat* certificate

We recommend using CA-signed certificates. However, if you do use self-signed certificates, the two certificates must have different common names. The Expressway does not allow two self-signed certificates with the same CN. If the *cup-xmpp* and *tomcat* (self-signed) certificates have the same CN, Expressway only trusts one of them, and some TLS attempts between Cisco Expressway-E and IM and Presence Service servers will fail. For more details, see CSCve56019.

Expressway Certificates

The Expressway certificate signing request (CSR) tool prompts for and incorporates the relevant Subject Alternative Name (SAN) entries as appropriate for the Unified Communications features that are supported on that Expressway.

The following table shows which CSR alternative name elements apply to which Unified Communications features:

Add these items as subject alternative names	**When generating a CSR for these purposes**			
	Mobile and Remote Access	Jabber Guest	XMPP Federation	Business to Business Calls
Unified CM registrations domains (despite their name, these have more in common with service discovery domains than with Unified CM SIP registration domains)	Required on Expressway-E only	—	—	—
XMPP federation domains	—	—	Required on Expressway-E only	—
IM and Presence chat node aliases (federated group chat)	—	—	Required	—
Unified CM phone security profile names	Required on Expressway-C only	—	—	—
(Clustered systems only) Expressway Cluster name	Required on Expressway-C only	Required on Expressway-C only	Required on Expressway-C only	—

Note

- You may need to produce a new server certificate for the Expressway-C if chat node aliases are added or renamed. Or when IM and Presence nodes are added or renamed, or new TLS phone security profiles are added.
- You must produce a new Expressway-E certificate if new chat node aliases are added to the system, or if the Unified CM or XMPP federation domains are modified.
- You must restart the Expressway for any new uploaded server certificate to take effect.

More details about the individual feature requirements per Expressway-C / Expressway-E are described below.

Expressway-C server certificate requirements

The Expressway-C server certificate must include the elements listed below in its list of Subject Alternative Names (SAN).

- **Unified CM phone security profile names**: The names of the **Phone Security Profiles** in Unified CM are configured for encrypted Transport Line Signaling (TLS) and are used for devices requiring remote access. Use the Fully Qualified Domain Name (FQDN) format and separate multiple entries with commas.

  It is essential to generate Certificate Signing Request (CSR) for the new node while adding a new Expressway-C node to an existing cluster of Expressway-C. It is mandated to put secure profile names as they are on CUCM, if secure registration of Mobile and Remote Access (MRA) client is needed over MRA. CSR creation on the new node will fail if "Unified CM phone security profile names" are just names or hostnames on CUCM device security profiles. This will force Administrators to change the value of "Unified CM phone security profile names" on CUCM under the **Secure Phone Profile** page.

  From X12.6, it is mandated that the Unified CM phone security profile name must be a Fully Qualified Domain Name (FQDN). It cannot be just any name or hostname or a value.

  For example, `jabbersecureprofile.domain.com`, `DX80SecureProfile.domain.com`

  Note The FQDN can comprise multiple levels. Each level's name can only contain letters, digits and hyphens, with each level separated by a period (dot). A level name cannot start or end with a hyphen, and the final level name must start with a letter.

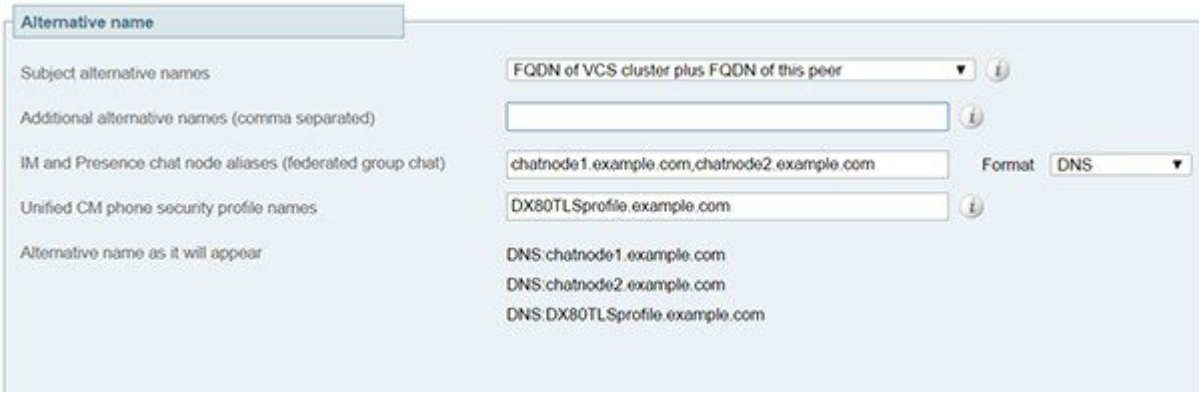
  Having the secure phone profiles as alternative names means that Unified CM can communicate via Transport Line Signaling (TLS) with the Expressway-C when it is forwarding messages from devices that use those profiles.

- **IM and Presence chat node aliases (federated group chat):** the **Chat Node Aliases** (e.g. chatroom1.example.com) that are configured on the IM and Presence servers. These are required only for Unified Communications XMPP federation deployments that intend to support group chat over TLS with federated contacts.

  The Expressway-C automatically includes the chat node aliases in the CSR, providing it has discovered a set of IM&P servers.

We recommend that you use DNS format for the chat node aliases when generating the CSR. You must include the same chat node aliases in the Expressway-E server certificate's alternative names.

Figure 1: Enter subject alternative names for security profiles and chat node aliases on the Expressway-C's CSR generator



Expressway-E server certificate requirements

The Expressway-E server certificate must include the elements listed below in its list of subject alternative names (SAN). If the Expressway-E is also known by other FQDNs, all of the aliases must be included in the server certificate SAN.

- **Unified CM registrations domains:** all of the domains which are configured on the Expressway-C for Unified CM registrations. Required for secure communications between endpoint devices and Expressway-E.

  The Unified CM registration domains used in the Expressway configuration and Expressway-E certificate, are used by Mobile and Remote Access clients to lookup the *_collab-edge* DNS SRV record during service discovery. They enable MRA registrations on Unified CM, and are primarily for service discovery.

  These service discovery domains may or may not match the SIP registration domains. It depends on the deployment, and they do not have to match. One example is a deployment that uses a .local or similar private domain with Unified CM on the internal network, and public domain names for the Expressway-E FQDN and service discovery. In this case, you need to include the public domain names in the Expressway-E certificate as SANs. There is no need to include the private domain names used on Unified CM. You only need to list the edge domain as a SAN.

  Select the *DNS* format and manually specify the required FQDNs. Separate the FQDNs by commas if you need multiple domains. You may select *CollabEdgeDNS* format instead, which simply adds the prefix collab-edge. to the domain that you enter. This format is recommended if you do not want to include your top level domain as a SAN (see example in following screenshot).

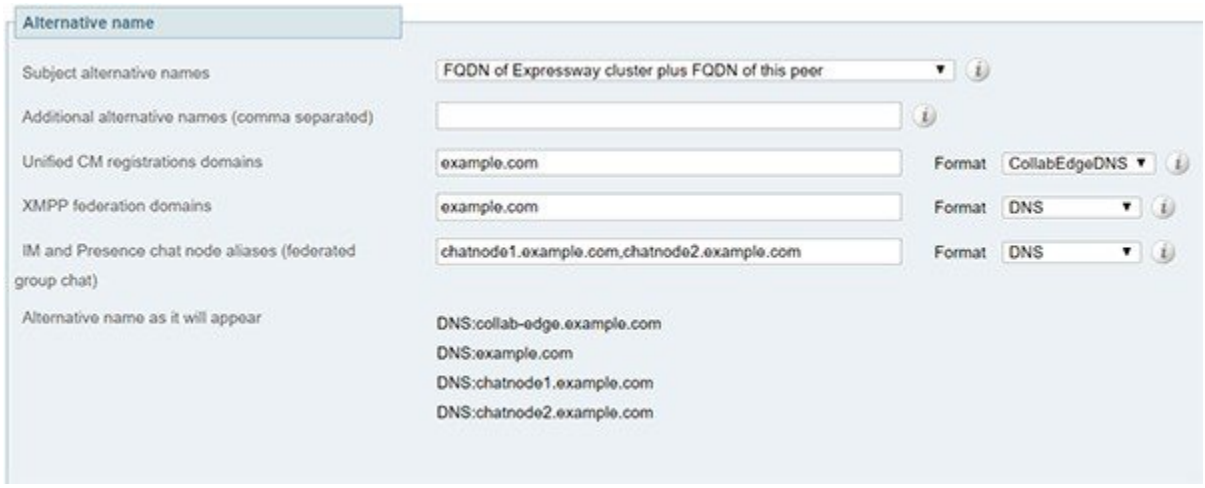
- **XMPP federation domains:** the domains used for point-to-point XMPP federation. These are configured on the IM&P servers and should also be configured on the Expressway-C as domains for XMPP federation.

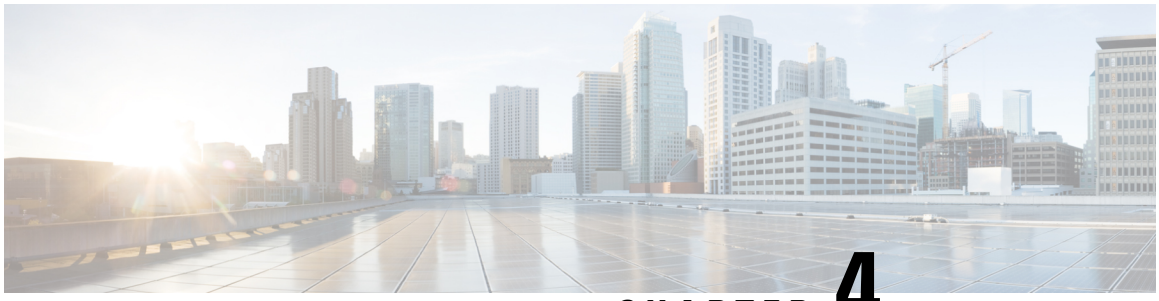
  Select the *DNS* format and manually specify the required FQDNs. Separate the FQDNs by commas if you need multiple domains. Do not use the *XMPPAddress* format as it may not be supported by your CA, and may be discontinued in future versions of the Expressway software.

- **IM and Presence chat node aliases (federated group chat):** the same set of **Chat Node Aliases** as entered on the Expressway-C's certificate. They are only required for voice and presence deployments which will support group chat over TLS with federated contacts.

You can copy the list of chat node aliases from the equivalent **Generate CSR** page on the Expressway-C.

Figure 2: Enter subject alternative names for Unified CM registration domains, XMPP federation domains, and chat node aliases, on the Expressway-E's CSR generator





CHAPTER 4

Using ACME on Expressway-E

From X12.5, the Cisco Expressway Series supports the ACME protocol (Automated Certificate Management Environment) which enables automatic certificate signing and deployment to the Cisco Expressway-E from a certificate authority such as Let's Encrypt. The main benefit of this feature is to generate low-cost server certificates to identify the Expressway-E, thereby reducing the cost of Expressway-based deployments like MRA (Mobile and Remote Access).

Due to the underlying validation mechanism this feature is most likely to be useful for MRA deployments. For Business to Business (B2B) applications, it's not always practical to include your primary domain in ACME certificates.

The configuration process is simple. You enter some information on the Cisco Expressway-E to create a certificate signing request (CSR), then the Expressway's ACME client interacts with the certificate authority to request the certificate. Expressway downloads the certificate and you click a button to deploy it. After this manual step, you can schedule renewal so that the certificate does not expire—because ACME certificates are deliberately short-lived.

One compromise of the ACME protocol is that it requires an inbound HTTP connection to port 80 on the Cisco Expressway-E. You can manage this risk with the Expressway's security features or, for highly secure environments, you can disable ACME and use the traditional CSR procedure with your preferred certificate authority.

No Jabber Guest support with ACME.

Currently, Expressway does not support ACME with Jabber Guest deployments.

This chapter explains the following:

ACME Deployment Overview

1. Deploy ACME Certificate Service
2. Configure ACME Certificate Service on Expressway-E
3. Generate a Certificate Signing Request for ACME

4. Sign the CSR using ACME Provider
5. [Optional] Check the Signed ACME Certificate
6. Deploy the ACME Certificate
7. Enable Automated Renewal of the ACME Certificate

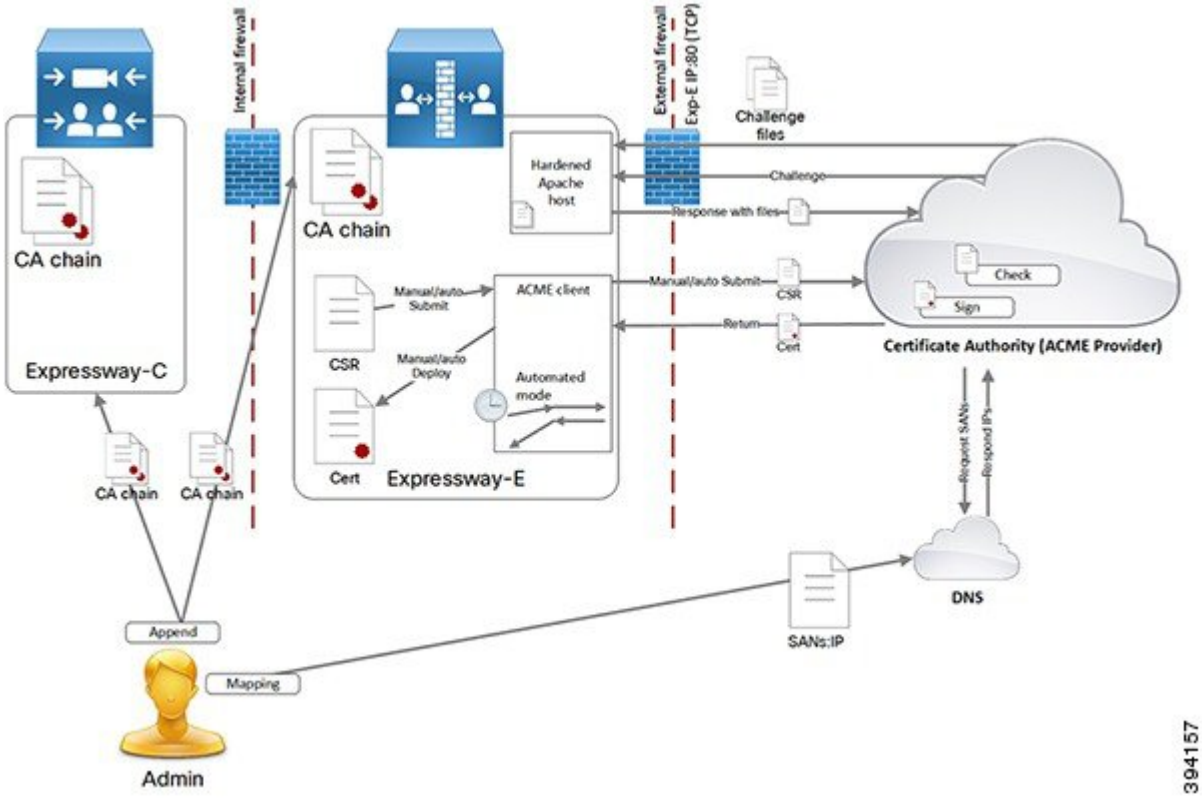
For clustered deployments, ACME **must be enabled individually on each peer** rather than at cluster level. Most certificate operations are performed per node.

How ACME Works

ACME is a client server protocol that enables automated certificate management of web hosts. The Expressway-E has an ACME client that interacts with an ACME provider, which is under the control of a certificate authority.

We currently work with the Let's Encrypt authority to generate server certificates.

We also use ACME to generate domain certificates for SNI (multitenancy), for which the process is essentially the same as the server certificate process. Multitenancy is only supported for HCS deployments and more information about using ACME with SNI is available in the Certificate Management and Service Discovery area of the Collaboration Knowledge Portal.



The ACME Certificate Service on the Expressway-E is a different method of requesting and applying server certificates to Expressway-E than the method described in the other parts of this document.

The essential signing process is:

- Define request > Submit to CA > CA generates and signs the certificate > Apply certificate.
- The ACME certificate service follows this process, but it removes the cost and some of the manual effort.
- One caveat about the process is that the CA has to interrogate the submitting host to verify that it controls the domains in the CSR.

Common Configuration

These tasks are always required when using the ACME Certificate Service:

1. Create a CSR on the Expressway-E.
2. Configure DNS with the domains from your CSR and map them to the Expressway-E public IP address.
3. Each domain must have an A record, not just the FQDNs.
4. Configure the ACME client with the provider details and your email address.

Lets encrypt verification process

For Let’s Encrypt to verify that all the domains requested in the CSR are under the control of requestor, it performs a challenge for each one. It provides files, containing random strings, that the requestor must be able to serve on port 80 for each domain in the CSR.

Let's Encrypt only issues the certificate after it successfully reads all the challenge files.

This is how it works when you manually control the process:

Procedure

Step 1 You initiate the signing process:

a. The ACME client opens an HTTPS connection to Let’s Encrypt and uploads the CSR.

b. Let’s Encrypt responds with a list of challenge files, one for each domain in the CSR.

c. The client places the challenge files on all the peers in the Expressway-E cluster.

d. Each Expressway-E peer starts a virtual Apache host, configured to serve only the challenge files.

e. The client notifies Let’s Encrypt it is ready to serve the challenge files.

f. Let’s Encrypt attempts to retrieve the challenge files.

g. The client polls Let’s Encrypt to see if the challenge process was successful.

h. If the challenge exchange was successful, then the client downloads the signed certificate, stores it in a staging area, and notifies you that the certificate is ready to deploy.

i. The Expressway-E peers close down the virtual Apache hosts.

Step 2 You initiate the deployment process:

a. The Expressway-E copies the staged certificate over the existing server certificate.

b. It copies the private key associated with the CSR over the existing private key.

c. Expressway-E signals to other internal processes that they need to reload the server certificate. (You do not need to restart the Expressway-E.)

The Expressway-E now presents the ACME certificate when making TLS connections.

Frequent expiry and low impact renewal

Let's Encrypt certificates are only valid for 90 days, by design. This means you need to renew your certificates more frequently, which we address in the ACME Certificate Service by:

- Providing an automated renewal mode, that fetches a new certificate when two-thirds of the validity period has expired.

  There is no notification at the two-thirds time if the service is not in automated mode. You are responsible for submitting a new signing request. Let's Encrypt sends expiration warning emails to the account that you use to configure the ACME client on Expressway-E.

- Removing the need to restart the Expressway-E when you use ACME Certificate Service to deploy a new certificate (either automated or manual deployment).

  The Expressway processes that use the certificate can load the new certificate without restarting. Expressway-E does not drop TLS connections, and presents the new certificate for new connection attempts.

  There is no interruption of service for Mobile Remote Access clients.

Note If you use a different method to upload a new server certificate, you must restart the Expressway-E. That behavior is unchanged with the introduction of the ACME Certificate Service.

Automated renewal mode

You can schedule a particular time, on one or more days of the week, when you configure automated renewal. The schedule is only used for deploying the certificate, not for requesting a new one.

When you put the service in automated mode, the service requests and receives an initial certificate, then deploys the certificate at the next scheduled opportunity. When two-thirds of that certificate's validity period has elapsed, the ACME Certificate Service automatically resubmits the stored CSR to get a new certificate.

There are two automated resubmission opportunities per day. These are deliberately at random times to improve security of the challenge process. At these times, the Expressway-E must accept requests on port 80, so it is better that they are unpredictable.

After the successful automated signing, the ACME Certificate Service automatically deploys the staged certificate at the next scheduled opportunity. This takes a few seconds and does not impact running processes that use the certificate.

More about the virtual Apache host

Let's Encrypt needs to verify that the certificate requestor controls the domain names in the CSR, using the challenge and validation process described above. Let's Encrypt must be able to access port 80 on all peers in the cluster because, when a domain resolves to several IP addresses, Let's Encrypt will connect to any one of them, at random.

It is impractical to try and restrict access to the Expressway-E port based on the source address, because Let's Encrypt does not have a concise list or CIDR containing all their servers.

To reduce the risk of malicious access, the Apache virtual host only runs during the challenge phase, and is also restricted to allow HTTP access only to the challenge files.

Apache is configured to listen on port 80 (if it is not already listening on that port) and forwards (only) ACME challenge traffic to the virtual Apache host.

The virtual host only listens on one unprivileged port on its localhost interface. The virtual host is hardened in the usual way. It denies: directory browsing, symbolic links, all options, and usage of .htaccess files. Due to this, the HTTP to HTTPS redirect is only supported if the Web Administration port for Expressway E is configured as the default 443 port.

If the Expressway-E is configured to redirect port 80 to 443:

- We add an exception to the 80 to 443 redirect rule for ACME challenge traffic. This exception is added automatically in the background and cannot be manually configured.
- The exception filters only on GET requests to the required paths (.well-known/acme-challenge/).

  Therefore, only GET requests on port 80 to specific file paths will reach the virtual host. All other requests are redirected to port 443 as normal.

If port 80 is not enabled on Expressway-E:

- We configure Apache to listen on port 80.
- We add a rule to redirect GET requests, on port 80, for the ACME challenge files, to the virtual Apache host.
- All other requests return HTTP error 404 (not found).

The challenge process can last a few minutes, depending on the number of domains in the CSR and the number of peers in the Expressway cluster.

When the challenge is complete:

- We remove the challenge files.
- We remove the exception to the 80 to 443 redirection rule.
- We stop Apache from listening on port 80, if it was not configured to allow redirection to 443.
- We stop the Apache virtual host.

Deploy ACME Certificate Service

Prerequisites

- Check the Let's Encrypt terms and conditions with your legal representative.
- Configure DNS with any mappings to Expressway-E that you need as CN or SAN in your certificate.
- Create an email account to use with the Let's Encrypt CA.
- Append the Let's Encrypt root CA certificate to Expressway trust stores.
- Append the Let's Encrypt intermediate CA certificate to Expressway trust stores.
- Enable TCP 80 inbound from the internet to your Expressway-Es' public addresses.
- Ensure that all domains on the SAN have a valid A record (not just the FQDNs). If the record of a **domain** is already used by another web server, you can configure the *collab-edge* domain on the CSR and configure an A record for it.

Append Let's Encrypt Root CA Certificate to Expressway Trust Stores

Let's Encrypt is a relatively new CA, so their own CA root certificate is cross signed by the established IdenTrust CA. Follow these steps to make sure that all your Expressways trust the Internet Security Research Group Root X1:

Procedure

Step 1 Go to https://letsencrypt.org/certs/isrgrootx1.pem.

Step 2 For each Expressway-E (and traversal Expressway-C) in the deployment you are securing with certificates signed by Let's Encrypt:

a. Sign on to the Expressway's web interface.

b. Go to **Maintenance** > **Security** > **Trusted CA certificate**.

c. In the **Upload** section of the page, select the certificate file you created.

d. Click **Append CA certificate**.

The trusted CA certificate list should now include the Internet Security Research Group root certificate.

Append Let's Encrypt Intermediate CA Certificate to Expressway Trust Stores

Procedure

Step 1 Go to https://letsencrypt.org/certs/lets-encrypt-r3.pem.

Step 2 For each Expressway-E (and traversal Expressway-C) in the deployment you are securing with certificates signed by Let's Encrypt:

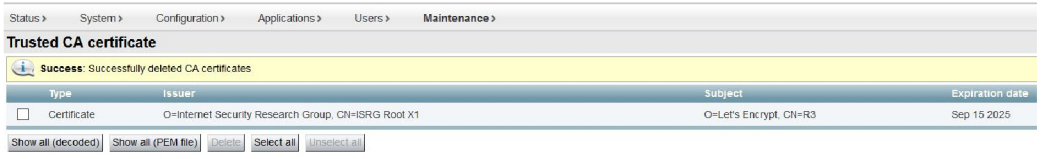
a. Sign on to the Expressway's web interface.

b. Go to **Maintenance** > **Security** > **Trusted CA certificate**.

c. In the **Upload** section of the page, select the certificate file you created.

d. Click **Append CA certificate**.

The trusted CA certificate list should now include both the Internet Security Research Group root certificate and the Let's Encrypt CA certificate.



Configure ACME Certificate Service on Expressway-E

Procedure

Step 1 Sign on to the Expressway-E and go to **Maintenance** > **Security** > **Server certificate**.

Step 2 Scroll down to the **ACME Certificate Service** section.

Step 3 Select the ACME **Provider** from the drop-down list.

This is the CA that signs your certificate. Currently, we only work with Let's Encrypt®.

Step 4 Enter an **Admin Email** address to use with the provider.

This should be a real address, so that you can receive communication from your ACME provider if necessary.

This address is your account name with the provider, and is linked to all certificate signing requests you make with this provider.

Step 5 Read the terms and conditions.

You may want to save a copy for your legal representatives to review, if they have not yet done that.

Step 6 Click **Accept Terms & Conditions**.

The ACME client on Expressway-E creates an account with your chosen provider.

The ACME Certificate Service on Expressway-E client is now ready to interact with your ACME provider.

Configure ACME for Each Domain Certificate

The ACME service on the Expressway-E, from version X12.5, can request and deploy domain certificates (used with SNI).

When you go to **Maintenance** > **> Security** > **Domain certificates**, the list of domains has an ACME column that shows the status of the ACME service for each domain.

Click **View/Edit** next to the domain name to enable the ACME service.

The process of configuring ACME service for domain certificates is the same as it is for the server certificate, only from a different place in the Expressway-E interface.

Generate a Certificate Signing Request for ACME

The process of creating your CSR is no different when you are using the ACME client. Follow the guidance in Generating a Certificate Signing Request.

Sign the CSR using ACME Provider

When you have a CSR saved on Expressway-E, and configured the ACME service, then you can submit the CSR to your ACME provider to verify and sign.

Procedure

Step 1 Go to **Maintenance** > **Security** > **Server certificate**.

Step 2 Scroll down to the ACME Service Configuration.

Step 3 Click **Sign CSR with ACME Provider**.

The ACME client on Expressway-E submits the saved CSR to the chosen provider.

Step 4 Wait a few minutes for the signing process to complete.

The provider checks DNS for CN and SAN attributes in your CSR, to verify that they match up with the Expressway-E address from which it received the signing request. The provider signs and returns the certificate, which the ACME client stores on Expressway-E, waiting for you to deploy it.

Step 5 Manually refresh the **Server certificate** page.

You see a success banner when the certificate is signed and ready to use.

[Optional] Check the Signed ACME Certificate

Procedure

Step 1 Go to **Maintenance** > **Security** > **Server certificate** and down to the **ACME Certificate Service** section.

The **Status** field shows that you have a signed certificate ready to deploy.

Step 2 In the **Pending ACME Certificate** field, click **Show (decoded)**.

Step 3 Verify the details are as you expected. If they are not, you may have to discard the pending cert and generate a new CSR.

Note It's possible that Let's Encrypt CA may ignore some of the attributes you provided in the CSR.

Deploy the ACME Certificate

Procedure

Step 1 Go to **Maintenance** > **Security** > **Server certificate** and down to the **ACME Certificate Service** section.

The **Status** field shows that you have a signed certificate ready to deploy.

Step 2 Click **Deploy Pending Cert**.

The Expressway-E starts using this certificate in transactions that require it to authenticate itself to the other party. There is no need to restart the Expressway-E.

Enable Automated Renewal of the ACME Certificate

ACME certificates are deliberately short-lived as a security precaution. At the time of writing, the validity period is 90 days from the date of issue.

The ACME Certificate Service on Expressway-E monitors the certificate validity, and warns you when two-thirds of the validity period has elapsed. You can manually respond by following the procedure outlined in previous topics.

To avoid this frequent task, you can use the automated renewal option to have the ACME Certificate Service renew and deploy your certificate for you.

Procedure

Step 1 Go to **Maintenance** > **Security** > **Server certificate** and down to the **ACME Certificate Service** section.

Step 2 Change the **ACME Automated Scheduler** field to *On*.

Step 3 Select one or more **Schedule Days** and a **Schedule Time**.

When two-thirds of the certificate's validity has elapsed, the ACME Certificate Service attempts to renew and deploy the server certificate at the given time on the next selected day.

Step 4 Click **Save**.

The **Status** shows that the service is in Automated mode. The next time it renews and deploys the certificate, it updates the **Last Deploy Status** and **Last Sign Status**.

Revoke an ACME Certificate

These are some of the reasons why you might want to revoke an ACME certificate on your Expressway-E:

- The Expressway-E has been compromised.
- You factory reset the Expressway-E.
- The purpose of the Expressway-E changed.
- The ACME account is no longer valid.

To revoke an ACME certificate, you need to prove to the provider that you own the Expressway-E's DNS address and that you control the original entries in the certificate. To do that you need to repeat the signing CSR process used for the certificate, but you do not need to redeploy the resulting certificate.

You should deploy a new certificate before you revoke the original certificate. Keep a copy of the certificate you want to revoke.

Caution Do not revoke a certificate that is in use, because that will interrupt all services that use this certificate.

Procedure

Step 1 Take a backup of the current certificate.

This precaution helps if you inadvertently overwrite your current certificate with one that you intend to revoke.

Step 2 Copy the certificate that you want to revoke into a temporary location on the Expressway-E. Remember the path to the location.

If you do not have a copy of the certificate you want to revoke, you may be able to retrieve it from https://crt.sh/.

Step 3 Create a CSR that contains all the domain names of the certificate you want to revoke. See Generate a Certificate Signing Request for ACME.

Step 4 Submit the CSR to be signed by the ACME provider that signed the original certificate. See Sign the CSR using ACME Provider.

You should now have a new, pending certificate that has matching SAN entries to the certificate you want to revoke.

This process has proved that you are entitled to revoke the original certificate.

Step 5 Sign in (as an administrator) to the CLI of the Expressway-E.

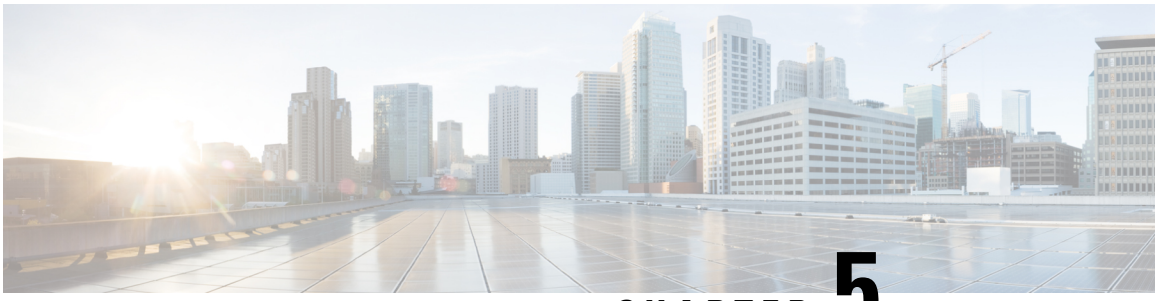
Step 6 Run the acmerevoke command in one of the following ways:

- If the default provider signed the certificate: `xcommand Acmerevoke "/path_to_cert_to_be_revoked"`
- Otherwise, specify the provider that signed the certificate: `xcommand Acmerevoke CertPath:"/path_to_cert_to_be_revoked" Provider:"ACME_Provider_Name"`

  (The same provider that signed the certificate must also revoke the certificate.)

The provider responds with 200 OK if it successfully revoked the certificate.

Step 7 Delete any saved copies of the revoked certificate.



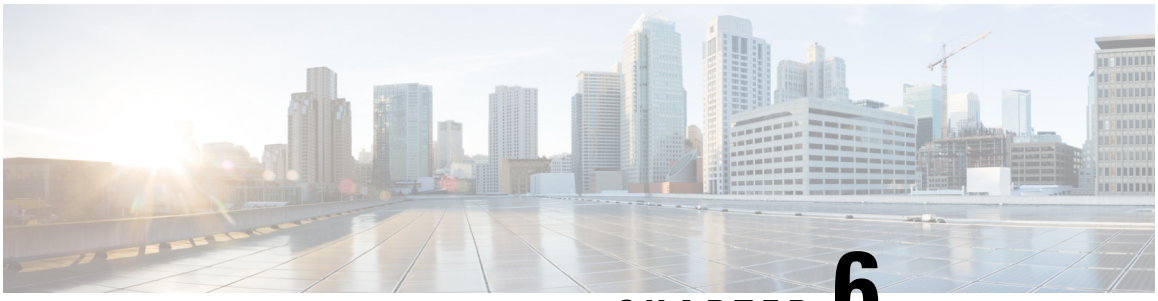
CHAPTER 5

View the Currently Uploaded Certificate

The **Server certificate data** section shows information about the server certificate currently loaded on the Expressway.

To view the currently uploaded server certificate file, click **Show (decoded)** to view it in a human-readable form, or click **Show (PEM file)** to view the file in its raw format.

Note To replace the currently uploaded server certificate with the Expressway's original certificate, click **Reset to default server certificate**.



CHAPTER 6

Loading Certificates and Keys Onto Expressway

The Expressway uses standard X.509 certificates. The certificate information must be supplied to the Expressway in PEM format. Typically three elements are loaded:

- The server certificate (which is generated by the certificate authority, identifying the ID of the certificate holder, and should be able to act as both a client and server certificate).
- The private key (used to sign data sent to the client, and decrypt data sent from the client, encrypted with the public key in the server certificate). This must only be kept on the Expressway and backed up in a safe place – security of the TLS communications relies upon this being kept secret.
- A list of certificates of trusted certificate authorities.

Note New installations of Expressway software (from X8.1 onwards) ship with a temporary trusted CA, and a server certificate issued by that temporary CA. We strongly recommend that you replace the server certificate with one generated by a trusted certificate authority, and that you install CA certificates for the authorities that you trust.

Note On Expressway-C and Expressway-E, we recommend that you do not upload multiple CA certificates with the same common name. This is because the endpoints may fail to log in if Expressway is configured to authenticate endpoints using an external IdP.

Warning Warning messages that may be displayed

From X8.10, the upload mechanism for server certificates (**Maintenance** > **Security** > **Server certificate**) displays a warning if the certificate fails to meet certain criteria. Cases when the warning is displayed include:

- Certificate does not have an acceptable level of security.
- Certificate is missing a common name (CN) attribute. An alarm is also raised in this case. Because some Expressway services don't work without the common name (MRA, Jabber Guest, and the Web Proxy for Cisco Meeting Server).
- The certification authority (CA) or certificate revocation list (CRL) is not recognized.

The certificate upload is not prevented.

This chapter explains the following:

Loading a Server Certificate and Private Key Onto Expressway

The Expressway's server certificate is used to identify the Expressway when it communicates with client systems using TLS encryption, and with web browsers over HTTPS.

As well as these instructions, a video demonstration of the process provided by Cisco TAC engineers is available on the Expressway/VCS Screencast Video List page.

Note We recommend you install the CA certificate first before installing the server certificate. Otherwise, the server certificate will fail to load.

To upload a server certificate:

1. Go to **Maintenance** > **Security** > **Server certificate**.
2. Use the **Browse** button in the **Upload new certificate** section to select and upload the **server certificate** PEM file.

Note Make sure to upload the Server certificate file with a valid FQDN.

   a. If you upload a certificate with a hostname or IP in the SAN field, the upload fails with the error "File upload failed.: Subject alternative name must be a valid FQDN".

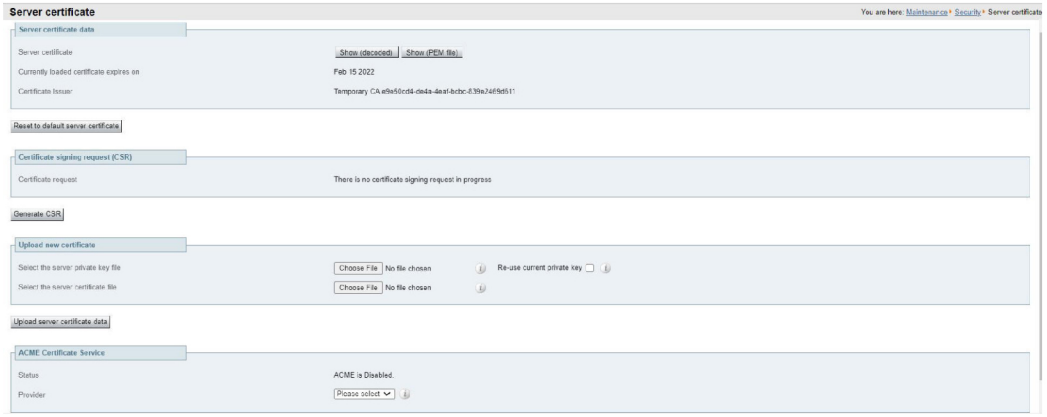
   b. If you upload a certificate with a hostname or IP in the CN (common name) field, the upload fails with the error "File upload failed.: Common name must be a valid FQDN".

3. If you used an external system to generate the Certificate Signing Request (CSR) you must also upload the **server private key** PEM file that was used to encrypt the server certificate. (The private key file will have been automatically generated and stored earlier if the Expressway was used to produce the CSR for this server certificate.)
   - The **server private key** PEM file must not be password protected.
   - You cannot upload a server private key if a certificate signing request is in progress.
4. Click **Upload server certificate data.**
   - When you generate a CSR in X7, the application puts **csr.pem** and **privkey_csr.pem** into **/tandberg/persistent/certs**.
   - When you generate a CSR in X8, the application puts **csr.pem** and **privkey.pem** into **/tandberg/persistent/certs/generated_csr**.

Re-use current private key check box - According to your local security requirements, check the **Re-use current private key** check box if you don't want a new private key. You may want to do this if you are extending the validity of your current certificate or re-issuing a previously generated CSR.

5. Use the **Provider** drop-down list in the **ACME Certificate Service** section to select trusted ACME clients used for signing of CSRs.

If you want to upgrade from X7 and have an unsubmitted CSR, then we recommend you to discard the CSR before upgrade, and then regenerate the CSR after upgrade.

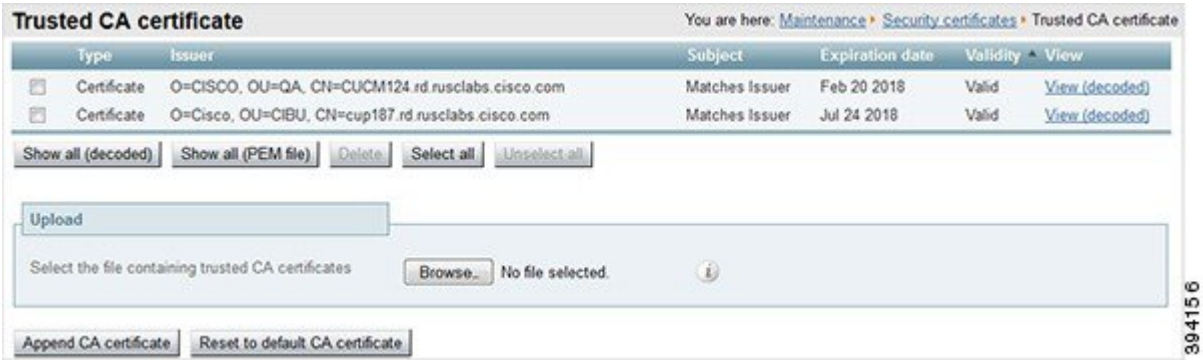


456940

Managing the Trusted CA Certificate List

The Trusted CA certificate page (**Maintenance** > **Security** > **Trusted CA certificate**) allows you to manage the list of certificates for the Certificate Authorities (CAs) trusted by this Expressway. When a TLS connection to Expressway mandates certificate verification, the certificate presented to the Expressway must be signed by a trusted CA in this list and there must be a full chain of trust (intermediate CAs) to the root CA.

- To upload a new file containing one or more CA certificates, **Browse** to the required PEM file and click **Append CA certificate**. This will append any new certificates to the existing list of CA certificates. If you are replacing existing certificates for a particular issuer and subject, you have to manually delete the previous certificates.
- To replace all of the currently uploaded CA certificates with the system's original list of trusted CA certificates, click **Reset to default CA certificate**.
- To view the entire list of currently uploaded trusted CA certificates, click **Show all (decoded)** to view it in a human-readable form, or click **Show all (PEM file)** to view the file in its raw format.
- To view an individual trusted CA certificate, click **View (decoded)** in the row for the specific CA certificate.
- To delete one or more CA certificates, tick the box(es) next to the relevant CA certificate(s) and click **Delete**.



Note this Recommendation

The maximum number of Certificate Authority (CAs) that can be uploaded/supported to the Expressway *trust store* is 1000.

Changing an Existing Server Certificate

Important This procedure on “Changing an Existing Server Certificate” does not apply to server certificates generated through “Let's Encrypt” certificate authority.

Before you begin

Generate Certificate Signing Request (CSR) before changing the server certificate. For more information, see Generating a Certificate Signing Request.

Note Set the Transport Line Signaling (TLS) verify mode to *Permissive* before changing the server certificate. This will protect against any errors encountered during certificate changes. Revert the TLS verify mode to *Enforce* after the changes.

Procedure

Step 1 Add the new Trusted CA certificate on all nodes in the cluster.

Step 2 If "TLS Verification mode" in **System** > **Clustering** is set to *Enforce*, change the "TLS Verification mode" to *Permissive*. Click **Save**.

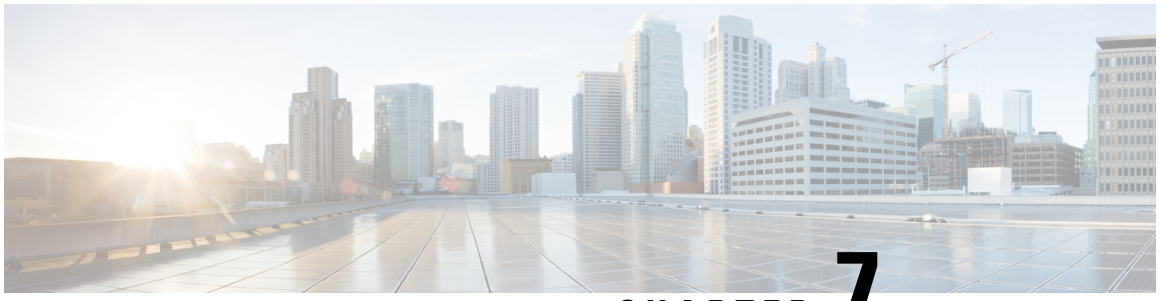
Step 3 Update the Server Certificate on all the nodes in the cluster.

Step 4 Restart the nodes one at a time.

Note Allow each node to recover before restarting the next node.

Step 5 If you change the "TLS Verification mode" from *Enforce* to *Permissive* in **step 2**, change it back to *Enforce*.

Step 6 Delete any unwanted CA certificates if they are no longer required.



CHAPTER 7

Manage Certificate Revocation Lists (CRLs)

Certificate revocation list files (CRLs) are used by the Expressway to validate certificates presented by client browsers and external systems that communicate with the Expressway over TLS/HTTPS. A CRL identifies those certificates that have been revoked and can no longer be used to communicate with the Expressway.

We recommend that you upload CRL data for the CAs that sign TLS/HTTPS client and server certificates. When enabled, CRL checking is applied for every CA in the chain of trust.

This chapter explains the following:

Certificate Revocation Sources

The Expressway can obtain certificate revocation information from multiple sources:

- Automatic downloads of CRL data from CRL distribution points.
- Through OCSP (Online Certificate Status Protocol) responder URIs in the certificate to be checked (SIP TLS only).
- Manual upload of CRL data.
- CRL data embedded within the Expressway's **Trusted CA certificate** file.

Limitations and Usage Guidelines

The following limitations and usage guidelines apply:

- When establishing SIP TLS connections, the CRL data sources are subject to the **Certificate revocation checking** settings on the **SIP configuration** page.
- Automatically downloaded CRL files override any manually loaded CRL files (except for when verifying SIP TLS connections, when both manually uploaded or automatically downloaded CRL data may be used).
- When validating certificates presented by external policy servers, the Expressway uses manually loaded CRLs only.

- When validating TLS connections with an LDAP server for remote login account authentication, the Expressway only uses CRL data that has been embedded into the **Trusted CA certificate** (**Tools** > **Security** > **Trusted CA certificate**).

  For LDAP connections, Expressway does not download the CRL from Certificate Distribution Point URLs in the server or issuing CA certificates. Also, it does not use the manual or automatic update settings on the **CRL management** page.

Automatic CRL Updates

We recommend you to configure the Expressway for automatic CRL updates. This ensures that the latest CRLs are available for certificate validation.

To configure the Expressway for automatic CRL updates:

Procedure

Step 1 Go to **Maintenance** > **Security** > **CRL management**.

Step 2 Set **Automatic CRL updates** to *Enabled*

Step 3 Enter the set of **HTTP(S) distribution points** from where the Expressway can obtain CRL files.

- you must specify each distribution point on a new line
- only HTTP(S) distribution points are supported; if HTTPS is used, the distribution point server itself must have a valid certificate
- PEM and DER encoded CRL files are supported
- the distribution point may point directly to a CRL file or to ZIP and GZIP archives containing multiple CRL files
- the file extensions in the URL or on any files unpacked from a downloaded archive do not matter as the Expressway will determine the underlying file type for itself; however, typical URLs could be in the format:
  - http://example.com/crl.pem
  - http://example.com/crl.der
  - http://example.com/ca.crl
  - https://example.com/allcrls.zip
  - https://example.com/allcrls.gz

Step 4 Enter the **Daily update time** (in UTC). This is the approximate time of day when the Expressway will attempt to update its CRLs from the distribution points.

Step 5 Click **Save**.

Manual CRL Updates

You can upload CRL files manually to the Expressway. Certificates presented by external policy servers can only be validated against manually loaded CRLs.

To upload a CRL file:

Note Ensure that the CRL file size is less than 16 MB.

Procedure

Step 1 Go to **Maintenence** > **Security** > **CRL management**.

Step 2 Click **Browse** and select the required file from your file system. It must be in PEM encoded format.

Step 3 Click **Upload CRL file**.

This uploads the selected file and replaces any previously uploaded CRL file.

Click **Remove revocation list** if you want to remove the manually uploaded file from the Expressway.

If a certificate authority's CRL expires, all certificates issued by that CA will be treated as revoked.

Online Certificate Status Protocol (OCSP)

The Expressway can establish a connection with an OCSP responder to query the status of a particular certificate.The Expressway determines the OCSP responder to use from the responder URI listed in the certificate being verified. The OCSP responder sends a status of “good”, “revoked” or “unknown” for the certificate.

The benefit of OCSP is that there is no need to download an entire revocation list. OCSP is supported for SIP TLS connections only.

Outbound communication from the Expressway-E is required for the connection to the OCSP responder. Check the port number of the OCSP responder you are using (port 80 or 443) and ensure that outbound communication is allowed to that port from the Expressway-E.

Configure Revocation Checking for SIP TLS Connections

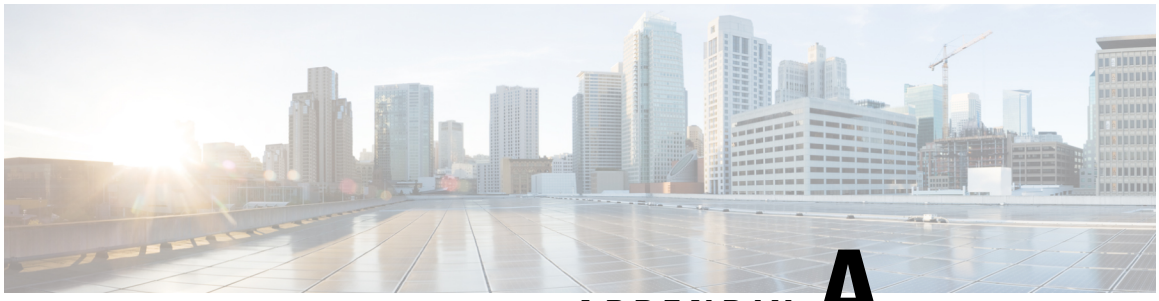
You must configure how certificate revocation checking is managed for SIP TLS connections.

Procedure

Step 1 Go to **Configuration** > **SIP**.

Step 2 Scroll down to the **Certificate revocation checking** section and configure the settings accordingly:

Field	Description	Usage tips
Certificate revocation checking mode	Controls whether revocation checking is performed for certificates exchanged during SIP TLS connection establishment.	We recommend that revocation checking is enabled.
Use OCSP	Controls whether the Online Certificate Status Protocol (OCSP) may be used to perform certificate revocation checking.	To use OCSP: • The X.509 certificate to be checked must contain an OCSP responder URI. • The OCSP responder must support the SHA-256 hash algorithm. If it is not supported, the OCSP revocation check and the certificate validation will fail.
Use CRLs	Controls whether Certificate Revocation Lists (CRLs) are used to perform certificate revocation checking.	CRLs can be used if the certificate does not support OCSP.
Allow CRL downloads from CDPs	Controls whether the download of CRLs from the CDP URIs contained in X.509 certificates is allowed.	
Fallback behavior	Controls the revocation checking behavior if the revocation status cannot be established, for example if the revocation source cannot be contacted. *Treat as revoked*: treat the certificate as revoked (and thus do not allow the TLS connection). *Treat as not revoked*: treat the certificate as not revoked. Default: Treat as not revoked	*Treat as not revoked* ensures that your system continues to operate in a normal manner if the revocation source cannot be contacted, however it does potentially mean that revoked certificates are accepted.



APPENDIX A

Troubleshooting

This chapter explains the following:

SIP TLS Negotiation Failures on Neighbor and Traversal Zones

If **TLS verify mode** is enabled, the neighbor system's FQDN or IP address, as specified in the **Peer address** field of the zone's configuration, is used to verify against the certificate holder's name in the X.509 certificate presented by that system. (The name must be in the SAN attribute of the certificate.) The certificate itself must also be valid and signed by a trusted certificate authority.

So when certificates have been generated with peer or cluster FQDNs, ensure that the zone's **Peer address** fields are configured with FQDNs rather than IP addresses.

Certificates with Key Length of 8192 Bits

SIP TLS zones may fail to become active if certificates use a key length of 8192 bits. We recommend using certificates with a key length of 4096 bits.

Service Failures when Using Mobile and Remote Access

Unified Communications mobile and remote access services can fail due to certificate errors if you upload a private key file that does not contain a trailing newline character.

Ensure that the private key file contains a trailing newline character.

Issues with SSH Failures and Unsupported OIDs

If you experience unknown ssh failures such as ssh tunnels failing to establish, please verify there are no unknown OIDs in the certificate. This can be done by checking that there are no undecoded numerical entries in the CN of the Issuer & Subject fields (from the GUI: **Maintenance** > **Security** > **Server Certificate** > **Show(decoded)** or from the console: `'openssl x509 -text -noout -in /tandberg/persistent/certs/server.pem')`

```
Invalid

subject=CN=blahdeblah,OU=IT

Security,O=BigBang,L=Washington,ST=District of

Columbia,C=US,1.3.6.1.4.1.6449.1.2.1.5.1 = #060C2B06010401B2310102010501

Valid

subject=CN=blahdeblah,OU=IT

Security,O=BigBang,L=Washington,ST=District of

Columbia,C=US,jurisdictionOfIncorporationLocalityName=Dover
```

CUCM Cipher Interop with Expressway

Servers during Transport Layer Security (TLS) handshake send Rivest Shamir Adleman (RSA)/Elliptic Curve Digital Signature Algorithm (ECDSA) ciphers. Expressway, as a client, can accept these ciphers.

Note Fresh install of Expressway comes default with ECDSA ciphers.

Expressway can negotiate an ECDSA cipher request.

Remember

- Certificate cipher with RSA, UCM sends either a *CallManager* or a *Tomcat* certificate.
- Certificate cipher with ECDSA, UCM sends either a *CallMananager-ECDSA* or a *Tomcat-ECDSA* certificate.
- Users must sequentially upload, to Expressway-C, signed Unified Call Manager (UCM) certificates as trusted Certificate Authority (CA) to verify the received certificate from UCM.

Reference Information

- **For Cipher Configuration:** Configuring ECDSA followed by RSA ciphers.

  ```
  ECDHE-ECDSA-AES128-GCM-SHAdefault:ECDHE-ECDSA-AES128-SHAdefault:ECDHE-ECDSA-
  AES128-SHA:ECDHE-ECDSA-AESdefault-GCM-SHA384:ECDHE-ECDSA-AESdefault-
  AES128-SHA:ECDHE-ECDSA-AESdefault-GCM-SHA384:ECDHE-ECDSA-AESdefault-
  GCM-SHA384
  ```

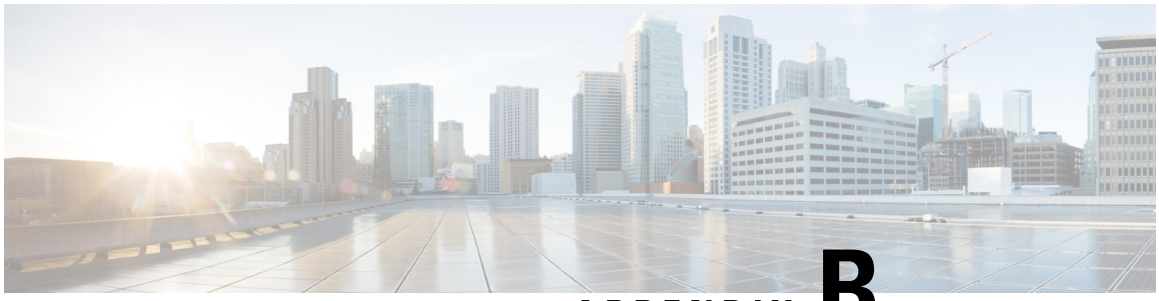
- **For Configuration in Expressway**

Add the below Ciphers under **Maintenance** > **Security** > **Ciphers**.

Note The following cipher change is required to send ECDSA as a high preference.

```
EECDH:EDH:HIGH:-
AES256+SHA:!MEDIUM:!LOW:!3DES:!MD5:!PSK:!eNULL:!aNULL:!aDH
```

Troubleshooting



APPENDIX B

Generate Certificate Using OpenSSL Only

This section describes the process for generating a private key and certificate request for the Expressway using OpenSSL. This is a generic process that relies only on the free OpenSSL package and not on any other software. It is appropriate when certificates are required to interface with neighboring devices for test purposes, and provide output to interact with Certificate Authorities.

The output for the certificate request generation process is given to a Certificate Authority which may be internal or external to the organization, and which is used to produce the X.509 certificates required by the Expressway to authenticate itself with neighboring devices.

This section also briefly describes how OpenSSL is used to manage a private Certificate Authority, but does not intend to be comprehensive. Various components of these processes are used when interfacing with third party CAs.

OpenSSL and Mac OS X or Linux

OpenSSL is already installed on Mac OS X, and is usually installed on Linux.

OpenSSL and Windows

If you do not have OpenSSL already installed, this is available as a free download from http://www.openssl.org/related/binaries.html.

Choose the relevant 32 bit or 64 bit OpenSSL - the 'Light' version is all that is needed.

If you receive a warning while installing OpenSSL that C++ files cannot be found, load the "Visual C++ Redistributables" also available on this site and then re-load the OpenSSL software.

This chapter explains the following:

Create a Certificate Request Using OpenSSL

This process creates a private key and certificate request for the server that is validated by a CA. This could be a CA that is created and managed locally, or a third-party CA.

Note

- This method to create a CSR should only be used if you have a good knowledge of working with OpenSSL as there is a potential for entering incorrect commands (especially with numerous SAN entries). Missing relevant SAN entries would require recreating the certificate at a later date.
- From version X8.5.1, the user interface provides an option to set the Digest algorithm. The default is set to SHA-256, with options to change to SHA-1, SHA-384, or SHA-512.

To generate the CSR from the command line with OpenSSL use these instructions:

Procedure

Step 1 SSH to the Expressway and log in as root.

Step 2 Make a new directory to do the work in - `mkdir /tmp/certtemp`

Step 3 Move in to this directory - `cd /tmp/certtemp`

Step 4 Copy the Open SSL configuration file we use for CSR to this directory, as we need to edit it **(Note: Keep the dot at the end)** - `cp /etc/openssl/csrreq.cnf`

Step 5 Open the file for editing – `vi csrreq.cnf`

Step 6 Find the line "`default_md = sha1`" and edit it so that it reads "`default_md = sha256`"

Step 7 Uncomment the line "`# req_extensions = v3_req`" by removing the **#** at the start of it

Step 8 Make sure that the line "`extendedKeyUsage=serverAuth, clientAuth`" is present within the section `[v3_req]`

Step 9 Find the line "`subjectAltName = ${ENV::CSR_ALT_NAME}`" and replace it such that it lists what you want in the Subject Alternative Names in the certificate e.g. "`subjectAltName = DNS:peer1vcs.example.com,DNS:peer2vcs.example.com,DNS:ClusterFQDN.example.com`". Make sure you add all the additional relevant entries. For MRA this may comprise:

a. Expressway E: `DNS:<CM domain name>, DNS:<XMPP federation domain>, DNS:<federation chat alias 1>, DNS:<federation chat alias 2>`, etc.

b. Expressway C: `DNS:<secure profile name 1>, DNS:<secure profile name 2>`, etc.

Step 10 Now save the file and exit.

Step 11 Run the following OpenSSL command to generate a new CSR and Private key for the VCS "`openssl req -nodes -newkey rsa:4096 -keyout privatekey.pem -out myrequest.csr -config csrreq.cnf`" changing the `rsa:nnnn` if required. (nnnn = keylength, recommended number is 4096).

Step 12 The console displays output similar to the following example, where you are required to enter information. You do not need to populate all of them, but some fields are required:

- Country
- State and Province
- Locality name
- Organization name
- Common name
- Email address - optional, can leave blank

- A challenge password - optional, can leave blank
- An optional company name - optional, can leave blank

```
Generating a 4096 bit RSA private key
............++
..............................................++
writing new private key to 'privatekey.pem'
-----
You are about to be asked to enter information that will be incorporated into your certificate
request.
What you are about to enter is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:Berkshire
Locality Name (eg, city) []:Reading
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Cisco
Organizational Unit Name (eg, section) []:CIBU
Common Name (eg, YOUR name) []:exp01.example.com
Email Address []:
```

When you have completed the fields, you will have two new files, **myrequest.csr** and **privatekey.pem**.

Step 13 (Optional) If you want to validate the DNS entries have been entered correctly into the request, the **myrequest.csr** file can be decoded using the command: `openssl req -text -noout -in myrequest.csr`

Step 14 Submit the CSR to your chosen Certificate Authority, who will provide the public certificate.

Step 15 Upload the public certificate to the VCS via **Maintenance** > **Security** > **Server certificate** webpage, "**Select the server certificate file**" entry box.

Step 16 Upload the **privatekey.pem** to the VCS via **Maintenance** > **Security** > **Server certificate** webpage, "**Select the server private key file**" entry box.

The **privatekey.pem** should be kept safe.

Operate as a Certificate Authority Using OpenSSL

A major deployment is to make use of a third-party certificate authority, or already have one internal to an organization's IT department. However, you can use OpenSSL to manage certificates in a private certificate authority as outlined below.

If you have already configured OpenSSL to act as a CA, go to section Create Signed Certificate Using OpenSSL.

Configure OpenSSL Act as CA

OpenSSL is powerful software, and when operating as a CA, requires a number of directories and databases to be configured for tracking issued certificates.

The list of directories and files can be found in the openssl configuration file under the section `[ CA_default ]`. By default, create the required files/directories:

- A **demoCA** directory in the current directory, with 3 subdirectories **certs**, **newcerts**, and **private**.
- An empty file called **index.txt** in the **demoCA** directory.
- A file called **serial** in the **demoCA** directory, storing a 2-digit number, such as "10".

For example, use the commands:

```
mkdir demoCA

cd demoCA

mkdir certs

mkdir newcerts

mkdir private

touch index.txt

echo 10 > serial
```

Create Certificate Authority Using OpenSSL

This process creates a private key and certificate of a Certificate Authority (CA), which is used to validate other certificates. Note that this will not be trusted by devices outside of those on which it is explicitly installed.

From a command prompt:

Procedure

Step 1 Ensure that you are in the **demoCA** directory.

Step 2 For Windows: copy **openssl.cfg** from the directory where OpenSSL is installed to the **demoCA** directory and rename it as **openssl_local.cfg**.

For Mac OS X: copy **/System/Library/OpenSSL/openssl.cnf** to the **demoCA** directory and rename it as **openssl_local.cfg**.

Step 3 Use a text editor to edit the **openssl_local.cfg** file that was created by the above copy command. Make the following modifications to the `[CA_default]` section:

a. Ensure that the line `copy_extensions = copy` does not have a # at the beginning of the line. Delete the # if it is there. If the line remains commented out, it will strip attributes in the CSR and, SSL Server and SSL Client attributes will not appear in the certificate.

b. Change `policy = policy_match` to `policy = policy_anything`

c. Change `dir = ./demoCA` to `dir = .`

d. Optionally, change `default_days = 365` (1 year validity of the generated certificate) to `default_days = 3650` (10 years, or choose another suitable value).

e. Save the file.

Step 4 Generate a private key for the CA by running the following command:

```
openssl genrsa -aes256 -out private/cakey.pem 4096
```

This prompts for a password to encrypt the private key: choose a strong password and record it in a safe place. The cakey.pem file is used to create the CA certificate and to sign other certificates and must also be kept secure.

Step 5 Generate the CA certificate by running the following command.

For Windows: `openssl req -new -x509 -days 3650 -key private/cakey.pem -config openssl_local.cfg -sha1 -extensions v3_ca -out cacert.pem`

For OS X: `openssl req -new -x509 -days 3650 -key private/cakey.pem -config openssl_local.cfg -sha1 -extensions v3_ca -out cacert.pem`

Step 6 Enter a passphrase for the key, and then enter the data requested, including:

- Country
- State or Province
- Locality name
- Organization name
- Organizational unit
- Common name - this is typicaly the name of the contact person for this CA
- Email address - optional, can leave blank

After you enter the requested data, the operation is complete and the certificate authority certificate **cacert.pem** is now available.

Create Signed Certificate Using OpenSSL

This process signs the server certificate with the generated CA key, using previously generated certificate request.

From a command prompt:

Procedure

Step 1 Ensure that you are in the **demoCA** directory.

Step 2 Ensure that the certificate request file **(certcsr.pem)** is available:

- If the certificate request is created using the Expressway (recommended process):

  Copy the file downloaded from the Expressway into the **demoCA** directory and rename it as **certcsr.pem**.

- If the certificate request is created using OpenSSL:

  Copy the previously generated certificate request into the **demoCA** directory and then covert it to PEM format by running the following command:

  ```
  openssl req -in certcsr.der -inform DER -out certcsr.pem -outform PEM
  ```

Step 3 Generate a signed server certificate by running the following command:

```
openssl ca -config openssl_local.cfg -cert cacert.pem -keyfile private/cakey.pem -in
certcsr.pem -out certs/server.pem -md sha1
```

If you receive a "failed to update database TXT_DB error number 2" error message, you can remove the contents of the index.txt file and then rerun the command.

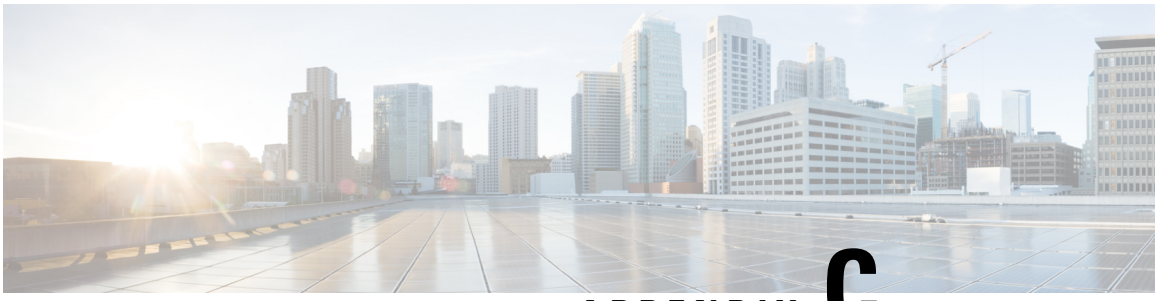
Step 4 You will be prompted to enter the password for the CA's private key.

The signed certificate for the server is now available as **demoCA/certs/server.pem**.

Create Self-Signed Certificates Using OpenSSL

We do not recommend creating self-signed certificates. They will not work in Unified Communications deployments.

Instead, you should create a Certificate Authority using OpenSSL as described above.



APPENDIX C

Convert a DER Certificate File to PEM Format

A private key, root (CA) certificate and the server / client certificate can be generated using third-party tools (or purchased from a certificate authority), and may be generated as PEM (required format, extension .pem) or DER (extension .cer) format files.

Certificates must be in PEM format for use on the Expressway. Conversion from DER to PEM format is done either using OpenSSL or Windows, as documented in the following sections.

Convert a DER certificate file to a PEM file using OpenSSL

To convert from DER to PEM format, on a system running openssl, execute the command:

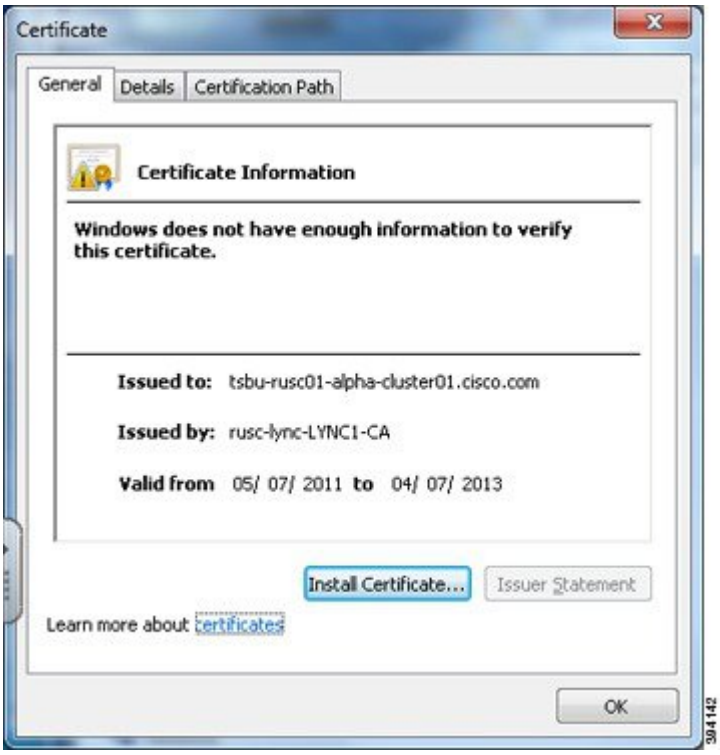
```
openssl x509 -in <filename>.cer -inform DER -out <filename>.pem -outform PEM
```

Convert a DER certificate file to a PEM file using Microsoft Windows

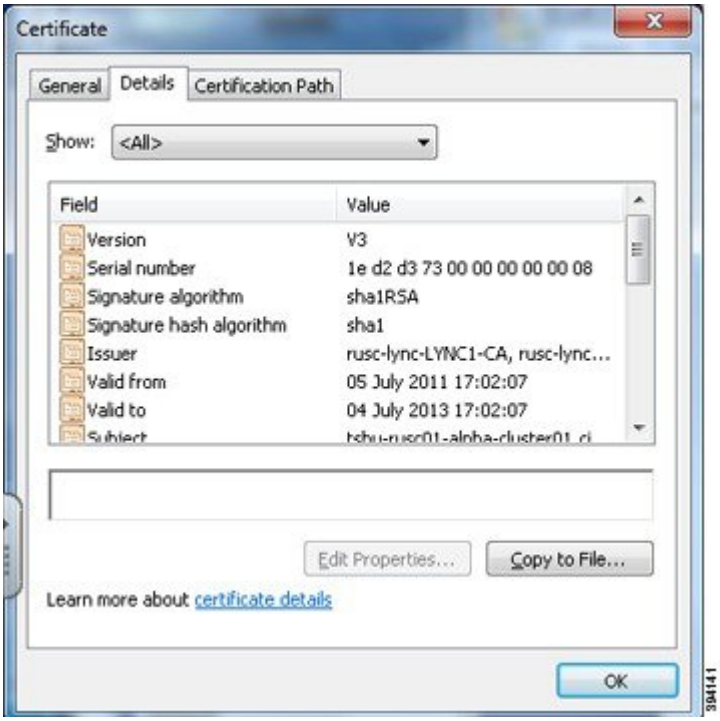
To convert from DER to PEM format using Microsoft Windows:

1. Double click the DER file to convert (this will likely have a '.cer' extension)

Convert a DER Certificate File to PEM Format



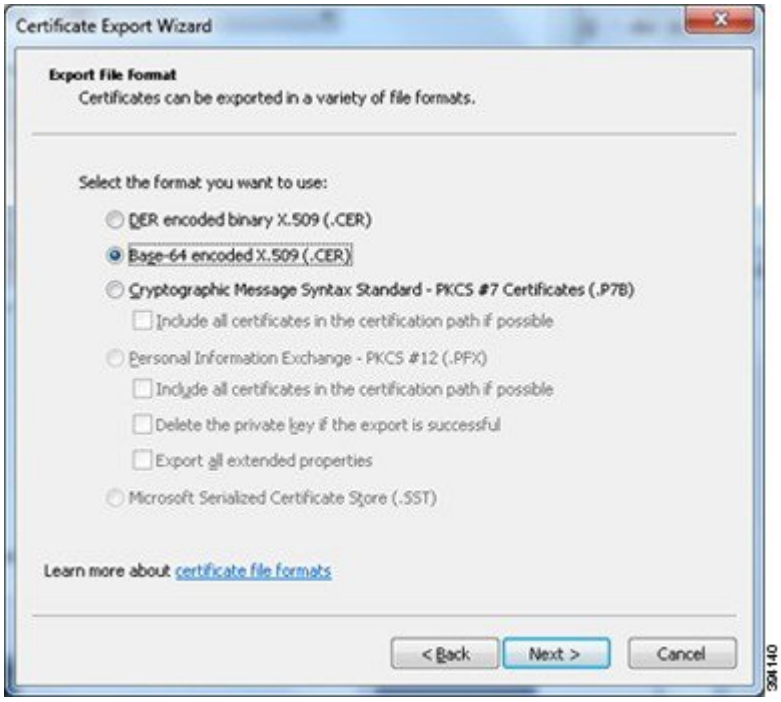
2. Select the **Details** tab



3. Click **Copy to File…**
4. On the **Welcome** page, click **Next**

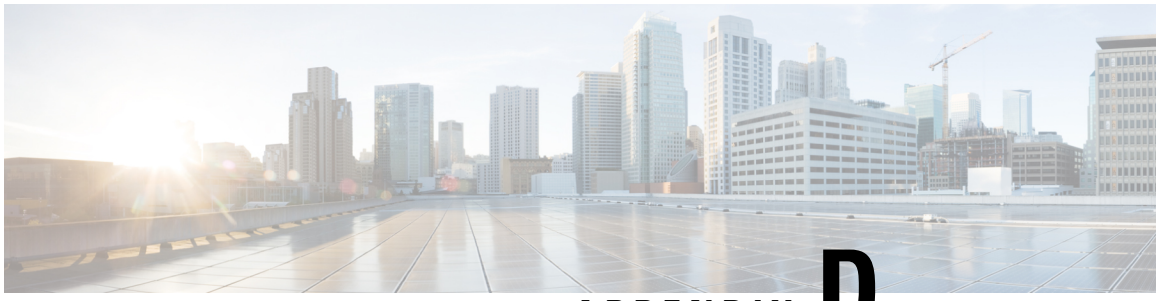
Convert a DER Certificate File to PEM Format

5. Select *Base-64 encoded X.509 (.CER)* and click **Next**



6. Click **Browse** and select required destination for file (e.g. **server.pem**) and then click **Next**
7. Click **Finish**
8. Change the filename from **server.pem.cer** to **server.pem**
9. This is used in the Loading Certificates and Keys Onto Expressway section of this document.

Convert a DER Certificate File to PEM Format



APPENDIX D

Decode Certificates

This section describes some methods to decode and view the content of certificates.

OpenSSL

A PEM file (e.g. **cert.pem**) can be decoded by the following command:

```
openssl x509 -text -in cert.pem
```

A DER file (e.g. **cert.cer**) can be decoded by the following command:

```
openssl x509 -text -inform DER -in cert.cer
```

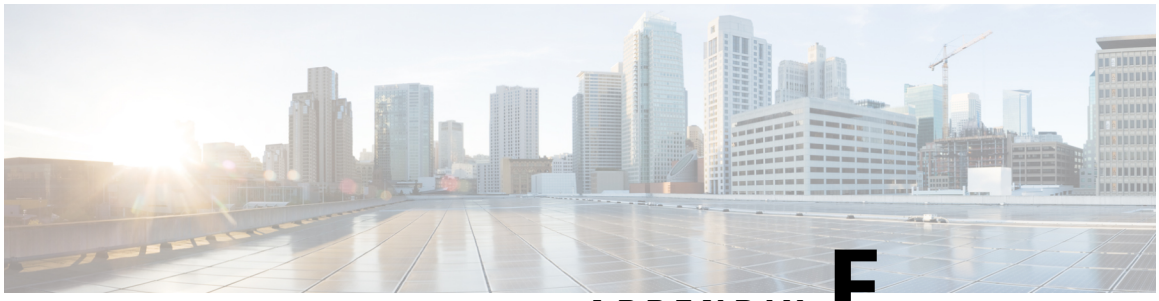
Firefox

In Firefox, you can view the certificate in use for a website by clicking the **Security Information** button on the address bar, and then clicking **More Information** followed by **View Certificate**.

Internet Explorer

In Internet Explorer, you can view the certificate in use for a website by clicking the lock icon to the right of the address bar. A **Website Identification** dialog appears. Click the **View Certificates** link at the bottom.

Decode Certificates



APPENDIX E

Enable AD CS to Issue Client and Server Certificates

Note The CA component of Microsoft Active Directory Certificate Services (AD CS) must be able to issue a certificate that can be used for authentication of the Expressway as client or server.

AD CS in Windows Server 2008 Standard R2 (and later) can issue these types of certificates, if you create a certificate template for them. **Earlier versions of Windows Server Standard Edition are not suitable.**

The default "Web Server" certificate template in AD CS creates a certificate for Server Authentication. The server certificate for the Expressway also needs Client Authentication if you want to configure a neighbor or traversal zone with mutual authentication (where **TLS verify mode** is enabled).

To set up a certificate template with both Server and Client authentication:

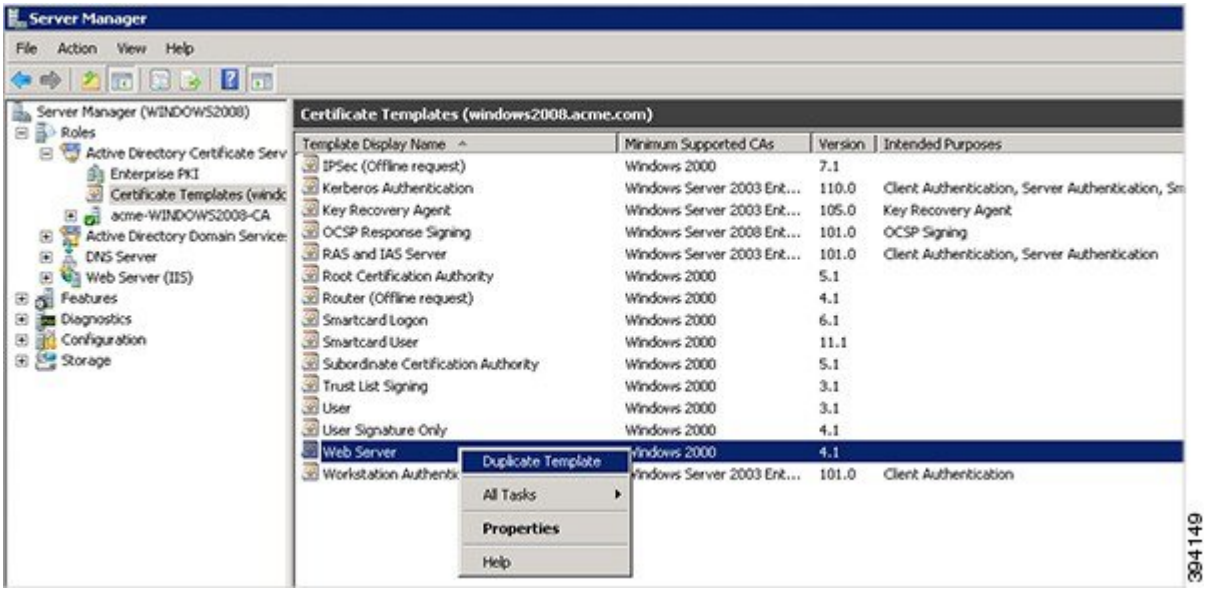
1. In Windows, launch **Server Manager** (**Start** > **Administrative Tools** > **Server Manager**).

   (Server Manager is a feature included with server editions of Windows.)

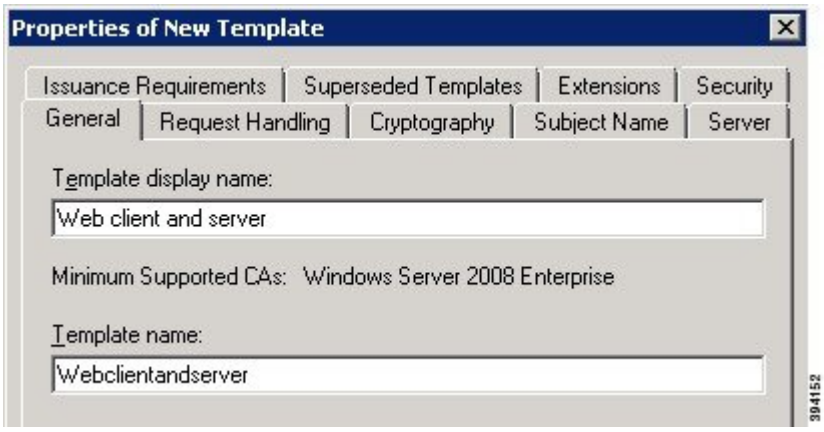
2. Expand the **Server Manager** navigation tree to **Roles** > **Active Directory Certificate Services** > **Certificate Templates (<domain>)**.

3. Right-click on **Web Server** and select **Duplicate Template**.

Enable AD CS to Issue Client and Server Certificates

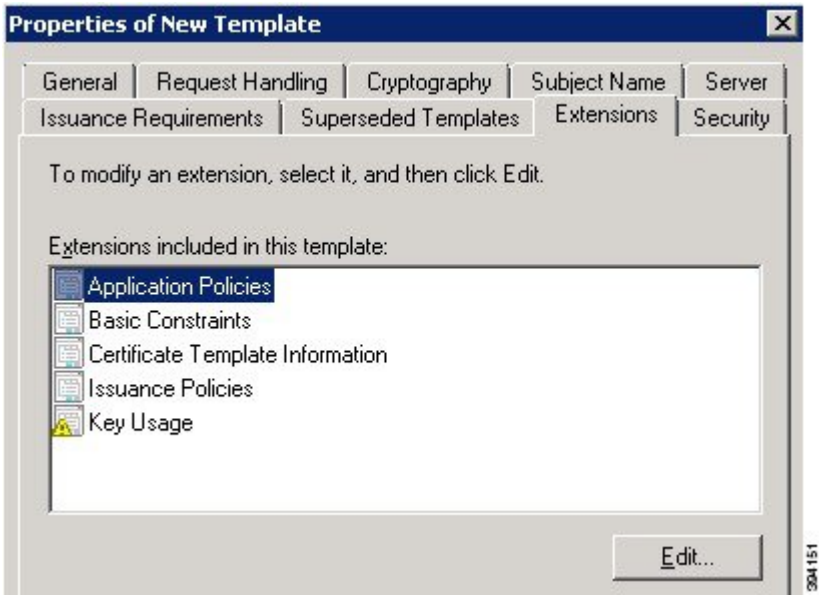


4. Select **Windows Server 2003 Enterprise** and click **OK**.
5. On the **General** tab, enter the **Template display name** and **Template name**, for example `Web client and server` and `Webclientandserver`.

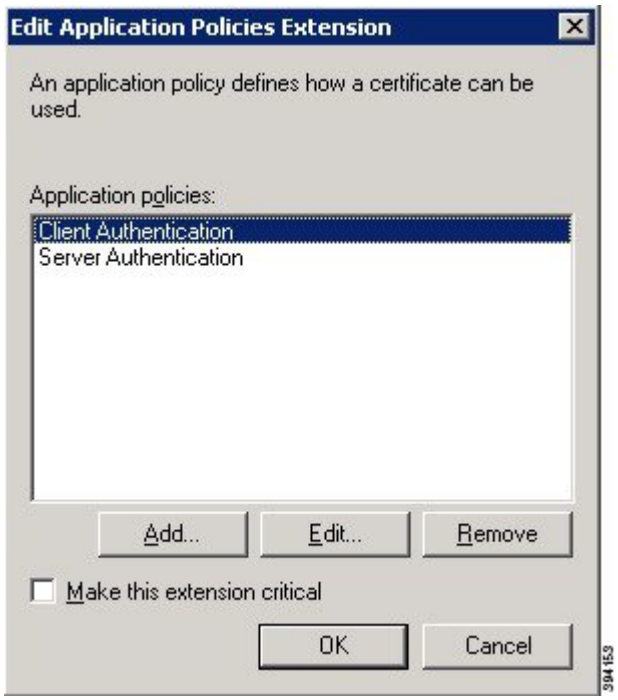


6. On the **Extensions** tab, select **Application Policies** and click **Edit**.

Enable AD CS to Issue Client and Server Certificates



7. Add **Client Authentication** to the set of application policies:
   a. Click **Add**
   b. Select **Client Authentication** and click **OK**
   c. Click **OK**

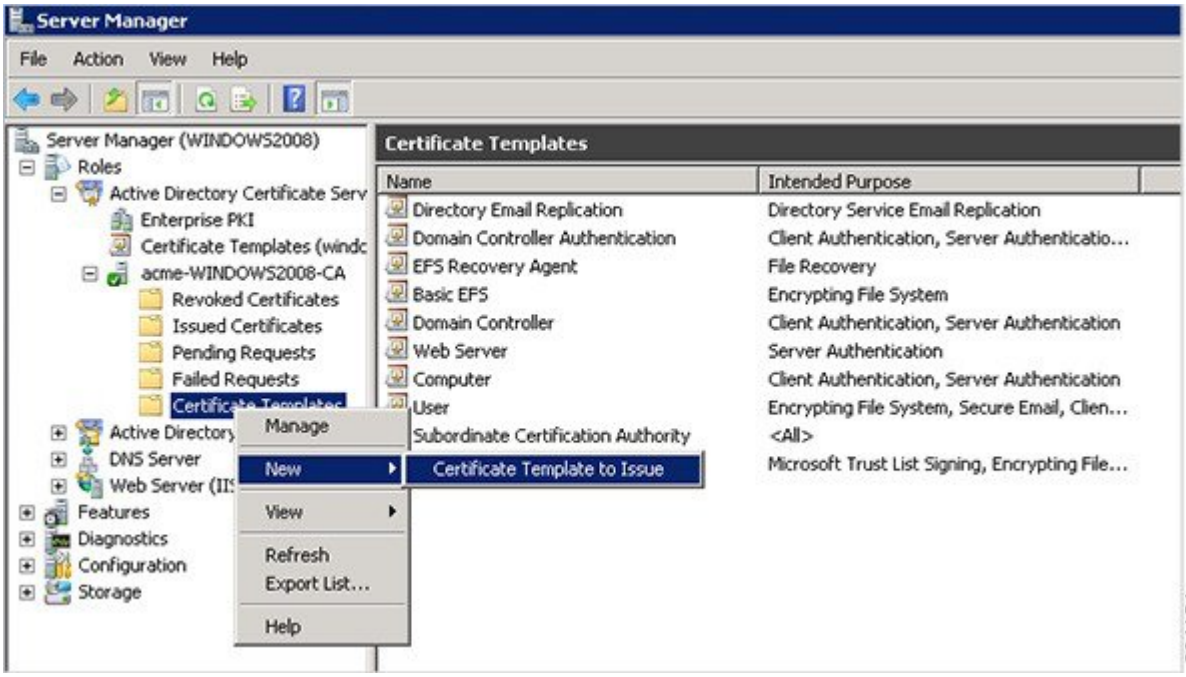


8. Click **OK** to complete the addition of the new template.
9. Add the new template to the Certificate Authority:

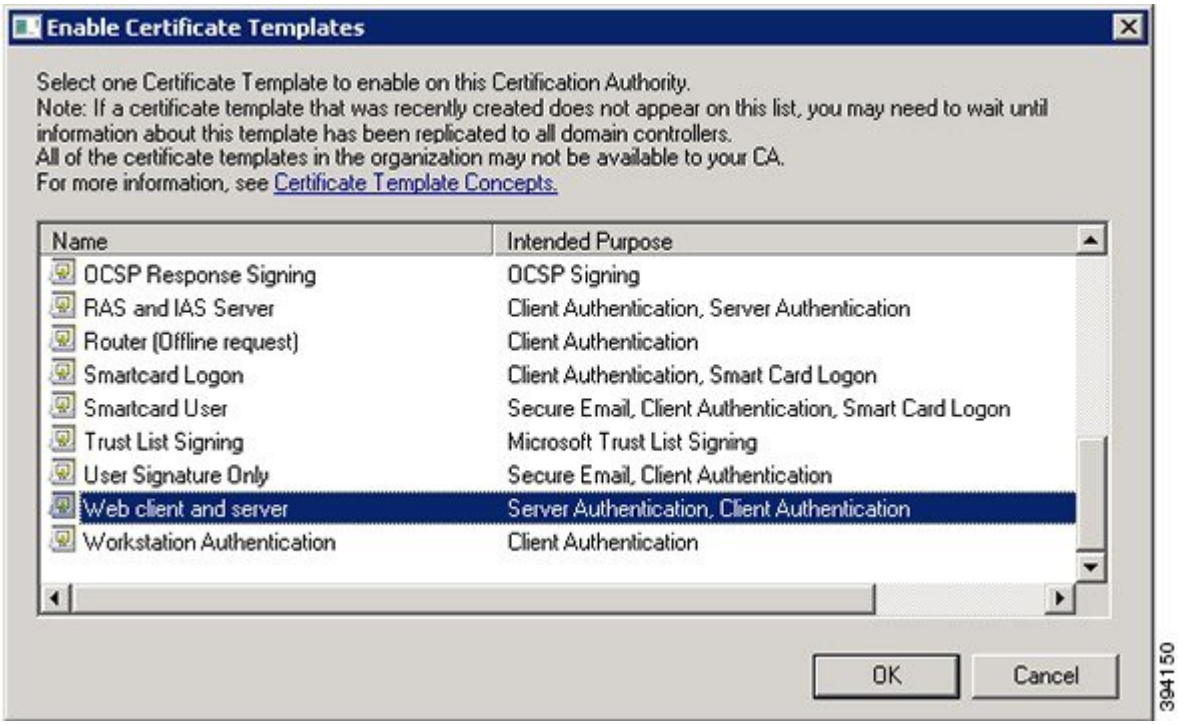
Enable AD CS to Issue Client and Server Certificates

a. Go to **Roles** > **Active Directory Certificate Services** > **<your certificate authority>**.

b. Right-click **Certificate Templates** and select **New** > **Certificate Template to Issue**

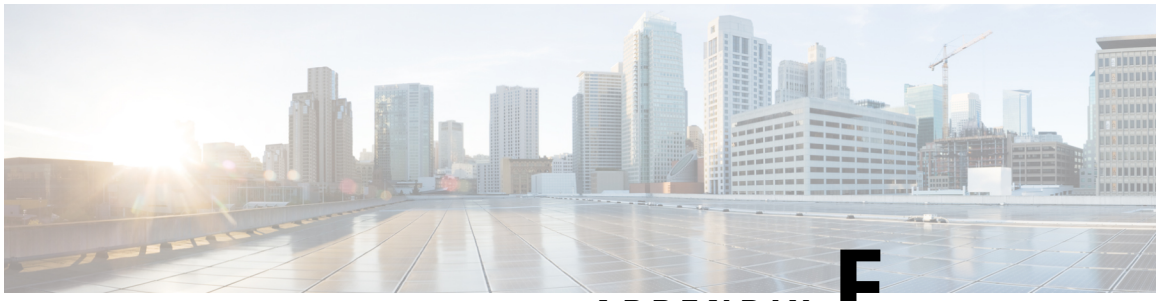


c. Select your new **Web client and server** template and click **OK**.



The new **Web client and server** template can now be used when submitting a certificate request to the Microsoft Certification Authority.

Enable AD CS to Issue Client and Server Certificates



APPENDIX F

Authorize a Request and Generate a Certificate Using Microsoft Certification Authority

This section describes how to authorize a certificate request and generate a PEM certificate file using Microsoft Certification Authority.

Note The CA component of Microsoft Active Directory Certificate Services (AD CS) must be able to issue a certificate that can be used for authentication of the Expressway as client or server.

AD CS in Windows Server 2008 Standard R2 (and later) can issue these types of certificates, if you create a certificate template for them. **Earlier versions of Windows Server Standard Edition are not suitable**.

1. Copy the certificate request file (for example, **certcsr.der** if generated via OpenSSL) to a location, such as the desktop, on the server where the Microsoft Certification Authority application is installed.
2. Submit the certificate request from a command prompt:
   - To generate a certificate with Server Authentication and Client Authentication, which is required if you want to configure a neighbor or traversal zone with mutual authentication (TLS verify mode), type:

     ```
     certreq -submit -attrib "CertificateTemplate:Webclientandserver"

     C:\Users\<user>\Desktop\certcsr.der
     ```

     See Enable AD CS to Issue Client and Server Certificates for details about how to set up the `Webclientandserver` certificate template.
   - To generate a certificate with Server Authentication only, type:

     ```
     certreq -submit -attrib "CertificateTemplate:WebServer"
     C:\Users\<user>\Desktop\certcsr.der
     ```

   This triggers the Certification Authority window to open:

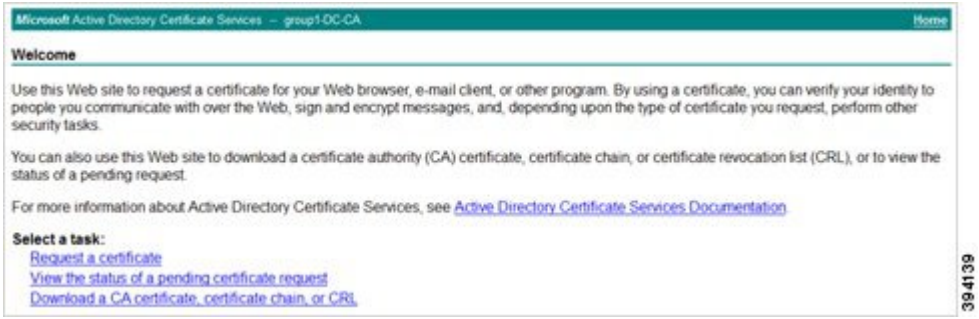
Authorize a Request and Generate a Certificate Using Microsoft Certification Authority

Note The command must be run as the administrator user.

3. Select the **Certification Authority** to use (typically only one is offered) and click **OK**.
4. When requested, save the certificate (browse to the required folder if the default **Libraries** > **Documents** folder is not to be used) calling it **server.cer** for example.
5. Rename **server.cer** to **server.pem** for use with the Expressway.

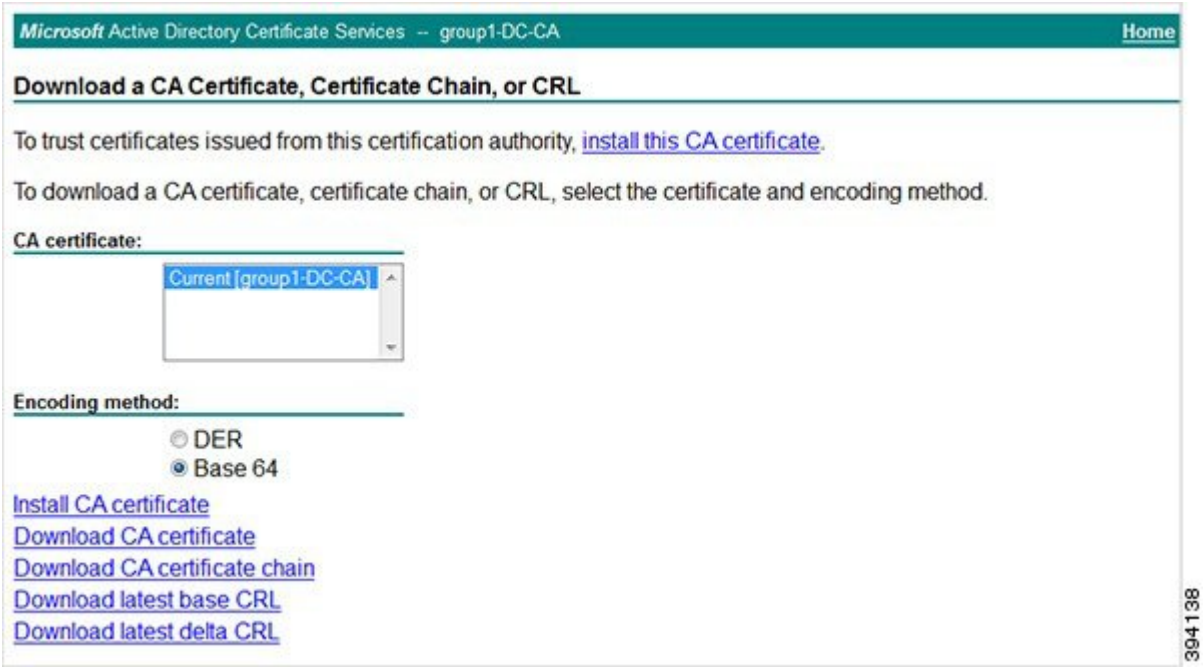
Get the Microsoft CA certificate

1. In your web browser, go to <IP or URL of the Microsoft Certificate Server>/certsrv and log in.

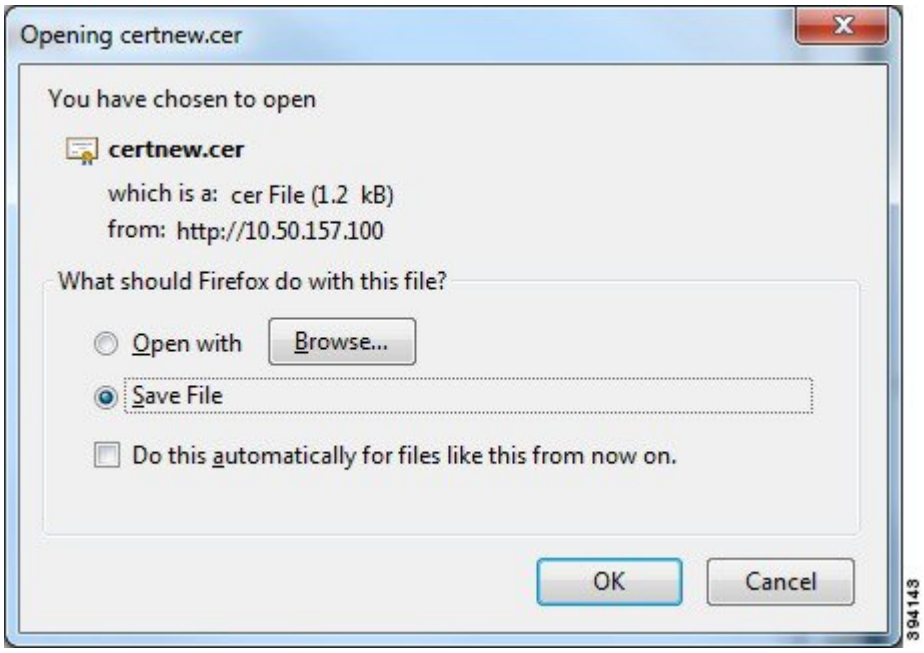


2. Select **Download a CA certificate, certificate chain or CRL**

Authorize a Request and Generate a Certificate Using Microsoft Certification Authority



3. Select the option **Base 64** under **Encoding method**.
4. Click **Download CA certificate** link.



5. Choose **Save File** and click **OK**.
6. Rename **certnew.cer** to **certnew.pem**.

Files **server.pem** and **certnew.pem** are now available.

Authorize a Request and Generate a Certificate Using Microsoft Certification Authority

Go to the Loading Certificates and Keys Onto Expressway section in this document to know how to upload **server.pem** and **certnew.pem** to Expressway.