Public Key Infrastructure Configuration Guide, Cisco IOS XE Release 2

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

**Cisco IOS XE PKI Overview Understanding and Planning a PKI**  
  Finding Feature Information  
  Information About Cisco IOS XE PKI  
    What Is Cisco IOS XE PKI  
      RSA Keys Overview  
      What Are CAs  
        Hierarchical PKI Multiple CAs  
          When to Use Multiple CAs  
          Certificate Enrollment How It Works  
            Certificate Enrollment Via Secure Device Provisioning  
            Certificate Revocation Why It Occurs  
      Planning for a PKI  
      Where to Go Next  
      Additional References  
      Glossary  
  Deploying RSA Keys Within a PKI  
  Finding Feature Information  
  Prerequisites for Configuring RSA Keys for a PKI  
  Information About RSA Keys Configuration  
    RSA Keys Overview  
    Usage RSA Keys Versus General-Purpose RSA Keys  
    Reasons to Store Multiple RSA Keys on a Router  
    Benefits of Exportable RSA Keys  
    Passphrase Protection While Importing and Exporting RSA Keys  
  How to Set Up and Deploy RSA Keys Within a PKI  
    Generating an RSA Key Pair  
    What to Do Next  
    Generating and Storing Multiple RSA Key Pairs  
    Exporting and Importing RSA Keys
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI High Availability Support 39</td>
</tr>
<tr>
<td>How to Configure Authorization and Revocation of Certificates for Your PKI 39</td>
</tr>
<tr>
<td>Configuring PKI Integration with a AAA Server 39</td>
</tr>
<tr>
<td>Troubleshooting Tips 43</td>
</tr>
<tr>
<td>Configuring a Revocation Mechanism for PKI Certificate Status Checking 44</td>
</tr>
<tr>
<td>The revocation-check Command 44</td>
</tr>
<tr>
<td>Nonces and Peer Communications with OCSP Servers 44</td>
</tr>
<tr>
<td>Configuring Certificate Authorization and Revocation Settings 47</td>
</tr>
<tr>
<td>Configuring Certificate-Based ACLs to Ignore Revocation Checks 47</td>
</tr>
<tr>
<td>Manually Overriding CDPs in a Certificate 47</td>
</tr>
<tr>
<td>Manually Overriding the OCSP Server Setting in a Certificate 48</td>
</tr>
<tr>
<td>Configuring CRL Cache Control 48</td>
</tr>
<tr>
<td>Configuring Certificate Serial Number Session Control 48</td>
</tr>
<tr>
<td>Examples 54</td>
</tr>
<tr>
<td>Troubleshooting Tips 55</td>
</tr>
<tr>
<td>Configuring Certificate Chain Validation 55</td>
</tr>
<tr>
<td>Configuration Examples for Setting Up Authorization and Revocation of Certificates 56</td>
</tr>
<tr>
<td>Examples Configuring and Verifying PKI AAA Authorization 56</td>
</tr>
<tr>
<td>Example Router Configuration 56</td>
</tr>
<tr>
<td>Example Debug of a Successful PKI AAA Authorization 58</td>
</tr>
<tr>
<td>Example Debugs of a Failed PKI AAA Authorization 59</td>
</tr>
<tr>
<td>Examples Configuring a Revocation Mechanism 60</td>
</tr>
<tr>
<td>Example Configuring an OCSP Server 60</td>
</tr>
<tr>
<td>Example Specifying a CRL and Then an OCSP Server 60</td>
</tr>
<tr>
<td>Example Specifying an OCSP Server 60</td>
</tr>
<tr>
<td>Example Disabling Nonces in Communications with the OCSP Server 61</td>
</tr>
<tr>
<td>Example Configuring a Hub Router at a Central Site for Certificate Revocation Checks 61</td>
</tr>
<tr>
<td>Examples Configuring Certificate Authorization and Revocation Settings 64</td>
</tr>
<tr>
<td>Example Configuring CRL Cache Control 64</td>
</tr>
<tr>
<td>Example Configuring Certificate Serial Number Session Control 65</td>
</tr>
<tr>
<td>Examples Configuring Certificate Chain Validation 67</td>
</tr>
<tr>
<td>Example Configuring Certificate Chain Validation from Peer to Root CA 67</td>
</tr>
<tr>
<td>Example Configuring Certificate Chain Validation from Peer to Subordinate CA 67</td>
</tr>
<tr>
<td>Example Configuring Certificate Chain Validation Through a Gap 67</td>
</tr>
<tr>
<td>Additional References 68</td>
</tr>
</tbody>
</table>
Finding Feature Information  141
Information About Source Interface Selection for Outgoing Traffic with Certificate Authority  141
  Certificates That Identify an Entity  142
  Source Interface for Outgoing TCP Connections Associated with a Trustpoint  142
How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority  142
  Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint  142
  Troubleshooting Tips  145
Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority  145
  Source Interface Selection for Outgoing Traffic with Certificate Authority Example  145
Additional References  146
Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority  147
Glossary  147
Cisco IOS XE PKI Overview Understanding and Planning a PKI

Cisco IOS XE public key infrastructure (PKI) provides certificate management to support security protocols such as IP Security (IPSec), secure shell (SSH), and secure socket layer (SSL).

This module identifies and describes concepts that are needed to understand, plan for, and implement a PKI.

- Finding Feature Information, page 1
- Information About Cisco IOS XE PKI, page 1
- Planning for a PKI, page 5
- Where to Go Next, page 5
- Additional References, page 6
- Glossary, page 7

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Cisco IOS XE PKI

- What Is Cisco IOS XE PKI, page 1
- RSA Keys Overview, page 2
- What Are CAs, page 3
- Certificate Enrollment How It Works, page 4
- Certificate Revocation Why It Occurs, page 5

What Is Cisco IOS XE PKI

A PKI is composed of the following entities:
- Peers communicating on a secure network
- At least one certification authority (CA) that grants and maintains certificates
- Digital certificates, which contain information such as the certificate validity period, peer identity information, encryptions keys that are used for secure communications, and the signature of the issuing CA
- An optional registration authority (RA) to offload the CA by processing enrollment requests
- A distribution mechanism (such as Lightweight Directory Access Protocol [LDAP] or HTTP) for certificate revocation lists (CRLs)

PKI provides customers with a scalable, secure mechanism for distributing, managing, and revoking encryption and identity information in a secured data network. Every entity (a person or a device) participating in the secured communications is enrolled in the PKI in a process where the entity generates an Rivest, Shamir, and Adelman (RSA) key pair (one private key and one public key) and has their identity validated by a trusted entity (also known as a CA or trustpoint).

After each entity enrolls in a PKI, every peer (also known as an end host) in a PKI is granted a digital certificate that has been issued by a CA. When peers must negotiate a secured communication session, they exchange digital certificates. Based on the information in the certificate, a peer can validate the identity of another peer and establish an encrypted session with the public keys contained in the certificate.

Although you can plan for and set up your PKI in a number of different ways, the figure below shows the major components that make up a PKI and suggests an order in which each decision within a PKI can be made. The figure is a suggested approach; you can choose to set up your PKI from a different perspective.

Figure 1  Deciding How to Set Up Your PKI

RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.
RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

What Are CAs

A CA, also known as a trustpoint, manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use a CA provided by a third-party CA vendor, or you can use an “internal” CA, which is the Cisco IOS XE Certificate Server.

- Hierarchical PKI Multiple CAs, page 3

Hierarchical PKI Multiple CAs

PKI can be set up in a hierarchical framework to support multiple CAs. At the top of the hierarchy is a root CA, which holds a self-signed certificate. The trust within the entire hierarchy is derived from the RSA key pair of the root CA. The subordinate CAs within the hierarchy can be enrolled with either the root CA or with another subordinate CA. These enrollment options are how multiple tiers of CAs are configured. Within a hierarchical PKI, all enrolled peers, can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA.

The figure below shows the enrollment relationships among CAs within a three-tiered hierarchy.

![Three-Tiered CA Hierarchy Sample Topology](image)

Each CA corresponds to a trustpoint. For example, CA11 and CA12 are subordinate CAs, holding CA certificates that have been issued by CA1; CA111, CA112, and CA113 are also subordinate CAs, but their CA certificates have been issued by CA11.

- When to Use Multiple CAs, page 3

When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed in branch offices while the root CA is at the office headquarters. Also, different granting policies...
can be implemented per CA, so you can set up one CA to automatically grant certificate requests while another CA within the hierarchy requires each certificate request to be manually granted.

Scenarios in which at least a two-tier CA is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multiple tier CA helps to control the size of the CRLs.
- When online enrollment protocols are used, the root CA can be kept offline with the exception of issuing subordinate CA certificates. This scenario provides added security for the root CA.

## Certificate Enrollment How It Works

Certificate enrollment is the process of obtaining a certificate from a CA. Each end host that wants to participate in the PKI must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA. The table below and the following steps describe the certificate enrollment process.

### Figure 3  Certificate Enrollment Process

1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA (or the RA, if applicable).
3. The CA receives the certificate enrollment request, and, depending on your network configuration, one of the following options occurs:
   - Manual intervention is required to approve the request.
   - The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.

   If you configure the end host to automatically request certificates from the CA, you should have an additional authorization mechanism.

4. After the request is approved, the CA signs the request with its private key and returns the completed certificate to the end host.
5. The end host writes the certificate to a storage area such as NVRAM.

- Certificate Enrollment Via Secure Device Provisioning, page 5
Certificate Enrollment Via Secure Device Provisioning

Secure Device Provisioning (SDP) is a web-based certificate enrollment interface that can be used to easily deploy PKI between two end devices, such as a Cisco IOS XE client and a Cisco IOS XE certificate server. SDP (also refer red to as Trusted Transitive Introduction [TTI]) is a communication protocol that provides a bidirectional introduction between two end entities, such as a new network device and a Virtual Private Network (VPN). SDP involves the following three entities:

- **Introducer**—A mutually trusted device that introduces the petitioner to the registrar. The introducer can be a device user, such as a system administrator.
- **Petitioner**—A new device that is joined to the secure domain.
- **Registrar**—A certificate server or other server that authorizes the petitioner.

SDP is implemented over a web browser in three phases—welcome, introduction, and completion. Each phase is shown to the user via a web page.

Certificate Revocation Why It Occurs

After each participant has successfully enrolled in the PKI, the peers are ready to begin negotiations for a secure connection with each other. Thus, the peers present their certificates for validation followed by a revocation check. After the peer verifies that the other peer’s certificate was issued by an authenticated CA, the CRL or Online Certificate Status Protocol (OCSP) server is checked to ensure that the certificate has not been revoked by the issuing CA. The certificate usually contains a certificate distribution point (CDP) in the form of a URL. Cisco IOS software uses the CDP to locate and retrieve the CRL. If the CDP server does not respond, the Cisco IOS software reports an error, which may result in the peer’s certificate being rejected.

Planning for a PKI

Planning for a PKI requires evaluating the requirements and expected use for each of the PKI components shown in Planning for a PKI, page 5. It is recommended that you (or the network administrator) thoroughly plan the PKI before beginning any PKI configuration.

Although there are a number of approaches to consider when planning the PKI, this document begins with peer-to-peer communication and proceeds as shown in Planning for a PKI, page 5. However you or the network administrator choose to plan the PKI, understand that certain decisions influence other decisions within the PKI. For example, the enrollment and deployment strategy could influence the planned CA hierarchy. Thus, it is important to understand how each component functions within the PKI and how certain component options are dependent upon decisions made earlier in the planning process.

Where to Go Next

As suggested in Where to Go Next, page 5, you begin to configure a PKI by setting up and deploying RSA keys. For more information, see the “Deploying RSA Keys Within a PKI” module in the Cisco IOS XE Security Configuration Guide: Secure Connectivity.
Additional References

The following sections provide references related to Cisco IOS XE PKI.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
<tr>
<td>Setting up and deploying RSA keys</td>
<td>“Deploying RSA Keys Within a PKI” module in the Cisco IOS XE Security Configuration Guide: Secure Connectivity</td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2459</td>
<td>Internet X.509 Public Key Infrastructure Certificate and CRL Profile</td>
</tr>
</tbody>
</table>
RFCs | Title
---|---
RFC 2511 | Internet X.509 Certificate Request Message Format
RFC 2528 | Internet X.509 Public Key Infrastructure
RFC 2559 | Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2
RFC 2560 | X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP
RFC 2585 | Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP
RFC 2587 | Internet X.509 Public Key Infrastructure LDAPv2 Schema
RFC 2875 | Diffie-Hellman Proof-of-Possession Algorithms
RFC 3029 | Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols

Technical Assistance

Description | Link
---|---
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.
To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.
Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.

http://www.cisco.com/techsupport

Glossary

**CDP** -- certificate distribution point. Field within a digital certificate containing information that describes how to retrieve the CRL for the certificate. The most common CDPs are HTTP and LDAP URLs. A CDP may also contain other types of URLs or an LDAP directory specification. Each CDP contains one URL or directory specification.

**Certificates** -- Electronic documents that bind a user’s or device’s name to its public key. Certificates are commonly used to validate a digital signature.
CRL -- certificate revocation list. Electronic document that contains a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when the certificate was issued and when it expires. A new CRL is issued when the current CRL expires.

CA -- certification authority. Service responsible for managing certificate requests and issuing certificates to participating IPSec network devices. This service provides centralized key management for the participating devices and is explicitly trusted by the receiver to validate identities and to create digital certificates.

peer certificate -- Certificate presented by a peer, which contains the peer’s public key and is signed by the trustpoint CA.

PKI -- public key infrastructure. System that manages encryption keys and identity information for components of a network that participate in secured communications.

RA -- registration authority. Server that acts as a proxy for the CA so that CA functions can continue when the CA is offline. Although the RA is often part of the CA server, the RA could also be an additional application, requiring an additional device to run it.

RSA keys -- Public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router.
Deploying RSA Keys Within a PKI

This module explains how to set up and deploy Rivest, Shamir, and Adelman (RSA) keys within a public key infrastructure (PKI). An RSA key pair (a public and a private key) is required before you can obtain a certificate for your router; that is, the end host must generate a pair of RSA keys and exchange the public key with the certification authority (CA) to obtain a certificate and enroll in a PKI.

- Finding Feature Information, page 9
- Prerequisites for Configuring RSA Keys for a PKI, page 9
- Information About RSA Keys Configuration, page 9
- How to Set Up and Deploy RSA Keys Within a PKI, page 11
- Configuration Examples for RSA Key Pair Deployment, page 21
- Where to Go Next, page 26
- Additional References, page 26
- Feature Information for Deploying RSA Keys Within a PKI, page 27

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring RSA Keys for a PKI

- Before setting up and deploying RSA keys for a PKI, you should be familiar with the module “Cisco IOS XE PKI Overview: Understanding and Planning a PKI.”
- All commands that begin as “crypto ca” have been changed to begin as “crypto pki.” Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About RSA Keys Configuration

- RSA Keys Overview, page 10
RSA Keys Overview

An RSA key pair consists of a public key and a private key. When setting up your PKI, you must include the public key in the certificate enrollment request. After the certificate has been granted, the public key will be included in the certificate so that peers can use it to encrypt data that is sent to the router. The private key is kept on the router and used both to decrypt the data sent by peers and to digitally sign transactions when negotiating with peers.

RSA key pairs contain a key modulus value. The modulus determines the size of the RSA key. The larger the modulus, the more secure the RSA key. However, keys with large modulus values take longer to generate, and encryption and decryption operations take longer with larger keys.

If you want a modulus value between 512 and 1024, enter an integer value that is a multiple of 64. If you want a value higher than 1024, enter 1536 or 2048. If you enter a value greater than 512, key generation may take a minute or longer.

**Note**

Peer public RSA key modulus values up to 4096 bits are automatically supported. The largest private RSA key modulus is 4096 bits. Therefore, the largest RSA private key a router may generate or import is 4096 bits. However, RFC 2409 restricts the private key size to 2048 bits or less for RSA encryption. The recommended modulus value for a CA is 2048 bits; the recommended modulus value for a client is 1024 bits.

Usage RSA Keys Versus General-Purpose RSA Keys

There are two mutually exclusive types of RSA key pairs—usage keys and general-purpose keys. When you generate RSA key pairs (via the `crypto key generate rsa` command), you will be prompted to select either usage keys or general-purpose keys.

**Usage RSA Keys**

Usage keys consist of two RSA key pairs—one RSA key pair is generated and used for encryption and one RSA key pair is generated and used for signatures. With usage keys, each key is not unnecessarily exposed. (Without usage keys, one key is used for both authentication methods, increasing the exposure of that key.)

**General-Purpose RSA Keys**

General-purpose keys consist of only one RSA key pair that used for both encryption and signatures. General-purpose key pairs are used more frequently than usage key pairs.

Reasons to Store Multiple RSA Keys on a Router

Configuring multiple RSA key pairs allows the Cisco IOS software to maintain a different key pair for each CA with which it is dealing or the software can maintain multiple key pairs and certificates with the same CA. As a result, the Cisco IOS software can match policy requirements for each CA without compromising
the requirements specified by the other CAs, such as key length, key lifetime, and general-purpose versus usage keys.

Named key pairs (which are specified via the `label key-label` option) allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate.

### Benefits of Exportable RSA Keys

⚠️ **Caution**

Exportable RSA keys should be carefully evaluated before use because using exportable RSA keys introduces the risk that these keys might be exposed. Any existing RSA keys are not exportable. New keys are generated as nonexportable by default. It is not possible to convert an existing nonexportable key to an exportable key.

Users can share the private RSA key pair of a router with standby routers, therefore transferring the security credentials between networking devices. The key pair that is shared between two routers will allow one router to immediately and transparently take over the functionality of the other router. If the main router were to fail, the standby router could be dropped into the network to replace the failed router without the need to regenerate keys, reenroll with the CA, or manually redistribute keys.

Exporting and importing an RSA key pair also enables users to place the same RSA key pair on multiple routers so that all management stations using Secure Shell (SSH) can be configured with a single public RSA key.

**Exportable RSA Keys in PEM-Formatted Files**

Using privacy-enhanced mail (PEM)-formatted files to import or export RSA keys can be helpful for customers who are using secure socket layer (SSL) or secure shell (SSH) applications to manually generate RSA key pairs and import the keys back into their PKI applications. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS XE routers instead of generating new keys.

### Passphrase Protection While Importing and Exporting RSA Keys

You have to include a passphrase to encrypt the PKCS12 file or the PEM file that will be exported, and when the PKCS12 or PEM file is imported, the same passphrase has to be entered to decrypt it. Encrypting the PKCS12 or PEM file when it is being exported, deleted, or imported protects the file from unauthorized access and use while it is being transported or stored on an external device.

The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

**How to Convert an Exportable RSA Key Pair to a Nonexportable RSA Key Pair**

Passphrase protection protects the external PKCS12 or PEM file from unauthorized access and use. To prevent an RSA key pair from being exported, it must be labeled “nonexportable.” To convert an exportable RSA key pair into a nonexportable key pair, the key pair must be exported and then reimported without specifying the “exportable” keyword.

### How to Set Up and Deploy RSA Keys Within a PKI

- Generating an RSA Key Pair, page 12
Generating an RSA Key Pair

Perform this task to manually generate an RSA key pair.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** (general-keys | usage-keys) [label *key-label*] [modulus *modulus-size*] [exportable]
4. **exit**
5. **show crypto key mypubkey rsa**

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto key generate rsa (general-keys</td>
<td>usage-keys) [label <em>key-label</em>] [modulus <em>modulus-size</em>] [exportable]</td>
</tr>
<tr>
<td>Example:</td>
<td>• If a <em>key-label</em> argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the router, is used.</td>
</tr>
<tr>
<td>Router(config)# crypto key generate rsa general-keys modulus 360</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
---|---
**Step 5** `show crypto key mypubkey rsa` | (Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.

**Example:**
```
Router# show crypto key mypubkey rsa
```

### What to Do Next
After you have successfully generated an RSA key pair, you can proceed to any of the additional tasks in this module to generate additional RSA key pairs, perform export and import of RSA key pairs, or configure additional security parameters for the RSA key pair (such as encrypting or locking the private key).

### Generating and Storing Multiple RSA Key Pairs
Perform this task to configure the router to generate and store multiple RSA key pairs and associate the key pairs with a trustpoint.

A trustpoint (also known as a CA) manages certificate requests and issues certificates to participating network devices. These services provide centralized key management for the participating devices and are explicitly trusted by the receiver to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You must have already generated an RSA key pair as shown in the task “Generating an RSA Key Pair, page 12.”

### SUMMARY STEPS
1. `crypto pki trustpoint name`
2. `rsakeypair key-label [key-size [encryption-key-size]]`
3. `exit`
4. `exit`
5. `show crypto key mypubkey rsa`

### DETAILED STEPS

#### Command or Action | Purpose
---|---
**Step 1** `crypto pki trustpoint name` | Creates a trustpoint and enters ca-trustpoint configuration mode.

**Example:**
```
Router(config)# crypto pki trustpoint fancy-ca
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 2** rsakeypair key-label [key-size [encryption-key-size]] | Specifies the key pair that is to be used with the trustpoint.  
· Specify the key-size argument for generating the key and specify the encryption-key-size argument to request separate encryption, signature keys, and certificates. |
| Example: Router(ca-trustpoint)# rsakeypair fancy-keys | |
| **Step 3** exit | Exits ca-trustpoint configuration mode. |
| Example: Router(ca-trustpoint)# exit | |
| **Step 4** exit | Exits global configuration mode. |
| Example: Router(config)# exit | |
| **Step 5** show crypto key mypubkey rsa | (Optional) Displays the RSA public keys of your router.  
This step allows you to verify that the RSA key pair has been successfully generated. |
| Example: Router# show crypto key mypubkey rsa | |

### Exporting and Importing RSA Keys

This section contains the following tasks that can be used for exporting and importing RSA keys. Whether you are using PKCS12 files or PEM files, exportable RSA keys allow you to use existing RSA keys on Cisco IOS routers instead of having to generate new RSA keys if the main router were to fail.

- Exporting and Importing RSA Keys in PKCS12 Files, page 14
- Exporting and Importing RSA Keys in PEM-Formatted Files, page 16

### Exporting and Importing RSA Keys in PKCS12 Files

Exporting and importing RSA key pairs enables users to transfer security credentials between devices. The key pair that is shared between two devices allows one device to immediately and transparently take over the functionality of the other router.

You must generate an RSA key pair and mark it “exportable” as specified in the task “Generating an RSA Key Pair, page 12.”
- When you import a PKCS12 file that was generated by a third-party application, the PKCS12 file must include a CA certificate.
- If you have to reexport an RSA key pair after you have already exported the key pair and imported the key pair to a target router, you must specify the `exportable` keyword when you are importing the RSA key pair.
- The largest RSA key a router may import is 2048-bits.

### SUMMARY STEPS

1. `crypto pki trustpoint name`  
2. `rsakeypair key-label [key-size [encryption-key-size]]`  
3. `exit`  
4. `crypto pki export trustpointname pkcs12 destination-url passphrase`  
5. `crypto pki import trustpointname pkcs12 source-url passphrase`  
6. `exit`  
7. `show crypto key mypubkey rsa`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>crypto pki trustpoint name</code></td>
<td>Creates the trustpoint name that is to be associated with the RSA key pair and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong><code>Router(config)# crypto pki trustpoint my-ca</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>rsakeypair key-label [key-size [encryption-key-size]]</code></td>
<td>Specifies the key pair that is to be used with the trustpoint.</td>
</tr>
<tr>
<td><strong>Example:</strong><code>Router(ca-trustpoint)# rsakeypair my-keys</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>exit</code></td>
<td>Exits ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong><code>Router(ca-trustpoint)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>
## Command or Action

<table>
<thead>
<tr>
<th>Step 4 crypto pki export trustpointname pkcs12 destination-url passphrase</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Exports the RSA keys via the trustpoint name.  
Note You can export the trustpoint using any of the following file system types: flash, FTP, null, NVRAM, remote file copying (RCP), SCP, system, TFTP, Webflash, Xmodem, or Ymodem. |

**Example:**

Router(config)# crypto pki export my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD

<table>
<thead>
<tr>
<th>Step 5 crypto pki import trustpointname pkcs12 source-url passphrase</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imports the RSA keys to the target router.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

Router(config)# crypto pki import my-ca pkcs12 tftp://tftpserver/my-keys PASSWORD

<table>
<thead>
<tr>
<th>Step 6 exit</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exits global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

Router(config)# exit

<table>
<thead>
<tr>
<th>Step 7 show crypto key mypubkey rsa</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) Displays the RSA public keys of your router.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

Router# show crypto key mypubkey rsa

---

## Exporting and Importing RSA Keys in PEM-Formatted Files

Perform this task to export or import RSA key pairs in PEM files.

You must generate an RSA key pair and mark it “exportable” as specified in the task “Generating an RSA Key Pair, page 12.”

- The largest RSA key a router may import is 2048 bits.
SUMMARY STEPS

1. `crypto key generate rsa {usage-keys | general-keys} label key-label [exportable]
2. `crypto key export rsa key-label pem {terminal | url url} {3des|des} passphrase
3. `crypto key import rsa key-label pem {usage-keys | {terminal | url url} [exportable]} passphrase
4. `exit
5. `show crypto key mypubkey rsa

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> crypto key generate rsa {usage-keys</td>
<td>general-keys} label key-label [exportable]</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# crypto key generate rsa general-keys label mykey exportable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto key export rsa key-label pem {terminal</td>
<td>url url} {3des</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# crypto key export rsa mycs pem url nvram: 3des PASSWORD</td>
<td>Tip Be sure to keep the PEM file safe. For example, you may want to store it on another backup router.</td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto key import rsa key-label pem {usage-keys</td>
<td>{terminal</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# crypto key import rsa mycs2 pem url nvram: PASSWORD</td>
<td>Note If you do not want the key to be exportable from your CA, import it back to the CA after it has been exported as a nonexportable key pair. Thus, the key cannot be taken off again.</td>
</tr>
<tr>
<td><strong>Step 4</strong> `exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> `show crypto key mypubkey rsa</td>
<td>(Optional) Displays the RSA public keys of your router.</td>
</tr>
<tr>
<td><strong>Example:</strong> Router# show crypto key mypubkey rsa</td>
<td></td>
</tr>
</tbody>
</table>
Encrypting and Locking Private Keys on a Router

Digital signatures are used to authenticate one device to another device. To use digital signatures, private information (the private key) must be stored on the device that is providing the signature. The stored private information may aid an attacker who steals the hardware device that contains the private key; for example, a thief might be able to use the stolen router to initiate a secure connection to another site by using the RSA private keys stored in the router.

Note

RSA keys are lost during password recovery operations. If you lose your password, the RSA keys will be deleted when you perform the password recovery operation. (This function prevents an attacker from performing password recovery and then using the keys.)

To protect the private RSA key from an attacker, a user can encrypt the private key that is stored in NVRAM via a passphrase. Users can also “lock” the private key, which blocks new connection attempts from a running router and protects the key in the router if the router is stolen by an attempted attacker.

Perform this task to encrypt and lock the private key that is saved to NVRAM.

Before encrypting or locking a private key, you should perform the following tasks:

- Generate an RSA key pair as shown in the task “Generating an RSA Key Pair, page 12.”
- Optionally, you can authenticate and enroll each router with the CA server.

Note

The RSA keys must be unlocked while enrolling the CA. The keys can be locked while authenticating the router with the CA because the private key of the router is not used during authentication.

Note

Interaction with Applications

An encrypted key is not effective after the router boots up until you manually unlock the key (via the `crypto key unlock rsa` command). Depending on which key pairs are encrypted, this functionality may adversely affect applications such as IP security (IPsec), SSH, and SSL; that is, management of the router over a secure channel may not be possible until the necessary key pair is unlocked.

> 

SUMMARY STEPS

1. `crypto key encrypt [write] rsa [name key-name] passphrase passphrase`
2. `exit`
3. `show crypto key mypubkey rsa`
4. `crypto key lock rsa name key-name ] passphrase passphrase`
5. `show crypto key mypubkey rsa`
6. `crypto key unlock rsa [name key-name] passphrase passphrase`
7. `configure terminal`
8. `crypto key decrypt [write] rsa [name key-name] passphrase passphrase`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step 1</th>
<th><strong>crypto key encrypt [write] rsa [name key-name] passphrase passphrase</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encrypts the RSA keys.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>After this command is issued, the router can continue to use the key; the key remains unlocked.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> If the <strong>write</strong> keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the encrypted key will be lost next time the router is reloaded.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th><strong>exit</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exits global configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th><strong>show crypto key mypubkey rsa</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) Shows that the private key is encrypted (protected) and unlocked.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> You can also use this command to verify that applications such as Internet Key Exchange (IKE) and SSH are properly working after the key has been encrypted.</td>
<td></td>
</tr>
</tbody>
</table>

| Step 4 | **crypto key lock rsa name key-name | passphrase passphrase** | **Purpose** |
|--------|-----------------------------------|------------------------|
|        | (Optional) Locks the encrypted private key on a running router. |             |
|        | **Note** After the key is locked, it cannot be used to authenticate the router to a peer device. This behavior disables any IPSec or SSL connections that use the locked key. Any existing IPSec tunnels created on the basis of the locked key will be closed. If all RSA keys are locked, SSH will automatically be disabled. |             |

<table>
<thead>
<tr>
<th>Step 5</th>
<th><strong>show crypto key mypubkey rsa</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) Shows that the private key is protected and locked.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The output will also show failed connection attempts via applications such as IKE, SSH, and SSL.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th><strong>crypto key unlock rsa [name key-name] passphrase passphrase</strong></th>
<th><strong>Purpose</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) Unlocks the private key.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> After this command is issued, you can continue to establish IKE tunnels.</td>
<td></td>
</tr>
</tbody>
</table>
Removing RSA Key Pair Settings

You might want to remove an RSA key pair for one of the following reasons:

- During manual PKI operations and maintenance, old RSA keys can be removed and replaced with new keys.
- An existing CA is replaced and the new CA requires newly generated keys; for example, the required key size might have changed in an organization so that you would have to delete the old 1024-bit keys and generate new 2048-bit keys.

Perform this task to remove all RSA keys or the specified RSA key pair that has been generated by your router.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto key zeroize rsa key-pair-label
4. exit
5. show crypto key mypubkey rsa

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>• Enter your password if prompted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 7 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 8 crypto key decrypt [write] rsa [name key-name] passphrase passphrase</td>
<td>(Optional) Deletes the encrypted key and leaves only the unencrypted key.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto key decrypt write rsa name pki.company.com passphrase password</td>
</tr>
</tbody>
</table>

Note: The write keyword immediately saves the unencrypted key to NVRAM. If the write keyword is not issued, the configuration must be manually written to NVRAM; otherwise, the key will remain encrypted the next time the router is reloaded.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto key zeroize rsa <em>key-pair-label</em></td>
<td>Deletes RSA key pairs from your router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto key zeroize rsa fancy-keys</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> show crypto key mypubkey rsa</td>
<td>(Optional) Displays the RSA public keys of your router. This step allows you to verify that the RSA key pair has been successfully generated.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show crypto key mypubkey rsa</td>
<td></td>
</tr>
</tbody>
</table>

## Configuration Examples for RSA Key Pair Deployment

### Generating and Specifying RSA Keys Example

The following example is a sample trustpoint configuration that shows how to generate and specify the RSA key pair “exampleCAkeys”:

```
crypto key generate rsa general-purpose exampleCAkeys
crypto ca trustpoint exampleCAkeys
enroll url http://exampleCAkeys/certsrv/mscep/mscep.dll
rsakeypair exampleCAkeys 1024 1024
```

### Exporting and Importing RSA Keys Examples

This section contains the following configuration examples:

#### Exporting and Importing RSA Keys in PKCS12 Files Example

In the following example, an RSA key pair “mynewkp” is generated on Router A, and a trustpoint name “mynewtp” is created and associated with the RSA key pair. The trustpoint is exported to a TFTP server, so
that it can be imported on Router B. By importing the trustpoint “mynewtp” to Router B, the user has imported the RSA key pair “mynewkp” to Router B.

Router A

crypto key generate rsa general label mykeys exportable
! The name for the keys will be: mynewkp
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA keys ...[OK]
! crypto pki trustpoint mynewtp rsakeypair mykeys
exit
crypto pki export mytp pkcs12 flash:myexport password mypassword123
Destination filename [myexport]? Writing pkcs12 file to tftp:/mytftpserver/myexport
CRYPTO_PKI:Exported PKCS12 file successfully.
Verifying checksum... OK (0x3307)
! July 8 17:30:09 GMT:%CRYPTO-6-PKCS12EXPORT_SUCCESS:PKCS #12 Successfully Exported.

Router B

crypto pki import mynewtp pkcs12 flash:myexport password mypassword123
Source filename [myexport]? CRYPTO_PKI:Imported PKCS12 file successfully.
! July 8 18:07:50 GMT:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.

Exporting and Importing and RSA Keys in PEM Files Example

The following example shows the generation, exportation, and importation of the RSA key pair "mytp", and verifies its status:

! Generate the key pair
!
Router(config)# crypto key generate rsa general-purpose label mytp exportable

The name for the keys will be: mytp
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys ...[OK]
!
! Archive the key pair to a remote location, and use a good password.
!
Router(config)# crypto pki export mytp pem url nvram:mytp 3des password mypassword123

% Key name:mytp
Usage:General Purpose Key
Exporting public key...
Destination filename [mytp.pub]?
Writing file to nvram:mytp.pub
Exporting private key...
Destination filename [mytp.prv]?
Writing file to nvram:mytp.prv
!
! Import the key as a different name.
!
Router(config)# crypto pki import mytp2 pem url nvram:mytp2 password mypassword123

% Importing public key or certificate PEM file...
Source filename [mytp2.pub]?

Exporting and Importing and RSA Keys in PEM Files Example
Exporting Router RSA Key Pairs and Certificates from PEM Files Example

The following example shows how to generate and export the RSA key pair “aaa” and certificates of the router in PEM files that are associated with the trustpoint “mycs.” This example also shows PEM-formatted files, which include PEM boundaries before and after the base64-encoded data, that are used by other SSL and SSH applications.

```
Router(config)# crypto key generate rsa general-keys label aaa exportable
The name for the keys will be:aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.
% Generating 512 bit RSA keys ...[OK]

Router(config)# crypto pki trustpoint mycs

Router(ca-trustpoint)# enrollment url http://mycs
rsakeypair aaa

Router(config)# crypto pki authenticate mycs
Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.

Router(config)# crypto pki enroll mycs
% Start certificate enrollment..
% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate.
```
For security reasons your password will not be saved in the configuration. Please make a note of it.

Password:
Re-enter password:

The fully-qualified domain name in the certificate will be: Router
The subject name in the certificate will be: host.company.com
Include the router serial number in the subject name? [yes/no]: n
Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
The certificate request sent to Certificate Authority
The certificate request fingerprint will be displayed.

Router(config)# crypto ca certificate fingerprint

% The certificate request fingerprint will be displayed.
% The show crypto ca certificate command will also show the fingerprint.

Router(config)# crypto ca keypair

% Certificate received from Certificate Authority

% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.ca]?
Reading file from tftp://10.1.1.2/username/msca.ca
Loading username/msca.ca from 10.1.1.2 (via FastEthernet0)!
[OK - 1082 bytes]
% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.prv]?
Reading file from tftp://10.1.1.2/username/msca.prv
Loading username/msca.prv from 10.1.1.2 (via FastEthernet0)!
[OK - 573 bytes]
% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.crt]?
Reading file from tftp://10.1.1.2/username/msca.crt
Loading username/msca.crt from 10.1.1.2 (via FastEthernet0)!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#

Importing Router RSA Key Pairs and Certificate from PEM Files Example

The following example shows how to import the RSA key pairs and certificate to the trustpoint “ggg” from PEM files via TFTP:

Router(config)# crypto pki import ggg pem url tftp://10.1.1.2/username/msca password

% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.ca]?
Reading file from tftp://10.1.1.2/username/msca.ca
Loading username/msca.ca from 10.1.1.2 (via FastEthernet0)!
[OK - 1082 bytes]
% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.prv]?
Reading file from tftp://10.1.1.2/username/msca.prv
Loading username/msca.prv from 10.1.1.2 (via FastEthernet0)!
[OK - 573 bytes]
% Importing certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [username/msca.crt]?
Reading file from tftp://10.1.1.2/username/msca.crt
Loading username/msca.crt from 10.1.1.2 (via FastEthernet0)!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#

Encrypting and Locking Private Keys on a Router Examples

This section contains the following configuration examples:
Configuring and Verifying an Encrypted Key Example

The following example shows how to encrypt the RSA key “pki-123.company.com.” Thereafter, the `show crypto key mypubkey rsa` command is issued to verify that the RSA key is encrypted (protected) and unlocked.

```
Router(config)# crypto key encrypt rsa name pki-123.company.com passphrase password
Router(config)# exit
Router# show crypto key mypubkey rsa
% Key pair was generated at:00:15:32 GMT Jun 25 2003
Key name:pki-123.company.com
Usage:General Purpose Key
*** The key is protected and UNLOCKED. ***
Key is not exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E0CC9A 1D23B52C
CD00910C ABD392AE BA6D0E3F FC47A0EF 8AFFEE340 0EC1E62B D40E7DCC
23C4D09E
03018B98 E0C07B42 3CFD1A32 2A3A13C0 1FF919C5 8DE9565F 1F020301 0001
% Key pair was generated at:00:15:33 GMT Jun 25 2003
Key name:pki-123.company.com.server
Usage:Encryption Key
Key is exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D7808D C5FF14AC
854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FF5757
3A41CE04 FDCB4A04 B9C6B84F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4
DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001
Router#
```

Configuring and Verifying a Locked Key Example

The following example shows how to lock the key “pki-123.company.com.” Thereafter, the `show crypto key mypubkey rsa` command is issued to verify that the key is protected (encrypted) and locked.

```
Router# crypto key lock rsa name pki-123.company.com passphrase password
!
Router# show crypto key mypubkey rsa
% Key pair was generated at:20:29:41 GMT Jun 20 2003
Key name:pki-123.company.com
Usage:General Purpose Key
*** The key is protected and LOCKED. ***
Key is exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00D3491E 2A21D383
854D7DA8 58AFBDAC 4E11A7DD E6C40AC6 66473A9F 0C845120 7C0C6EC8 1FF5757
3A41CE04 FDCB4A04 B9C6B84F BC7D624B 470339A3 DE739D3E F7DDB549 91CD4DA4
DF190D26 7033958C 8A61787B D40D28B8 29BCD0ED 4E6275C0 6D020301 0001
Router#
```
Where to Go Next

After you have generated an RSA key pair, you should set up the trustpoint. If you have already set up the trustpoint, you should authenticate and enroll the routers in a PKI. For information on enrollment, see the module “Configuring Certificate Enrollment for a PKI.”

Additional References

The following sections provide references related to configuring RSA keys for a PKI.

- Related Documents, page 26
- Technical Assistance, page 26

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of PKI, including RSA keys, certificate enrollment, and CAs</td>
<td>Cisco IOS XE PKI Overview: Understanding and Planning a PKI</td>
</tr>
<tr>
<td>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>
Feature Information for Deploying RSA Keys Within a PKI

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1  Feature Information for Deploying RSA Keys Within a PKI

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cisco IOS XE 4096-Bit Peer Public Key Support</td>
<td>Cisco IOS XE Release 2.4</td>
<td>This feature introduces Cisco IOS XE 4096-bit peer public key support.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following section provides information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RSA Keys Overview</td>
</tr>
<tr>
<td>Exporting and Importing RSA Keys</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature allows you to transfer security credentials between devices by exporting and importing RSA keys. The key pair that is shared between two devices will allow one device to immediately and transparently take over the functionality of the other router.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Benefits of Exportable RSA Keys</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Exporting and Importing RSA Keys in PKCS12 Files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified by this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>crypto ca export pkcs12, crypto ca import pkcs12, crypto key generate rsa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(IKE)</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Import of RSA Key Pair and Certificates in PEM Format | Cisco IOS XE Release 2.1          | This feature allows customers to use PEM-formatted files to import or export RSA key pairs. PEM-formatted files allow customers to directly use existing RSA key pairs on their Cisco IOS XE routers instead of generating new keys. The following sections provide information about this feature:  
  - Benefits of Exportable RSA Keys  
  - Exporting_and_Importing_RSA_Keys_in_PEM-Formatted_Files  
  The following commands were introduced by this feature: crypto ca export pem, crypto ca import pem, crypto key export pem, crypto key import pem |
| Multiple RSA Key Pair Support                    | Cisco IOS XE Release 2.1          | This feature allows a user to configure a router to have multiple RSA key pairs. Thus, the Cisco IOS XE software can maintain a different key pair for each identity certificate. The following sections provide information about this feature:  
  - Reasons_to_Store_Multiple_RSA_Keys_on_a_Router_1055232  
  - Generating_and_Storing_Multiple_RSA_Key_Pairs  
  The following commands were introduced or modified by this feature: crypto key generate rsa, crypto key zeroize rsa, rsakeypair |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Protected Private Key Storage| Cisco IOS XE Release 2.1         | This feature allows a user to encrypt and lock the RSA private keys that are used on a Cisco IOS XE router, thereby, preventing unauthorized use of the private keys. The following section provides information about this feature:  
  • Encrypting_and_Locking_Private_Keys_on_a_Router  
  The following commands were introduced or modified by this feature: crypto key decrypt rsa, crypto key encrypt rsa, crypto key lock rsa, crypto key unlock rsa, show crypto key mypubkey rsa |
Configuring Authorization and Revocation of Certificates in a PKI

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI).

- Finding Feature Information, page 31
- Prerequisites for Authorization and Revocation of Certificates, page 31
- Information About Authorization and Revocation of Certificates, page 32
- How to Configure Authorization and Revocation of Certificates for Your PKI, page 39
- Configuration Examples for Setting Up Authorization and Revocation of Certificates, page 56
- Additional References, page 68
- Feature Information for Certificate Authorization and Revocation, page 69

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy

Tip

It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the certification authority (CA).
- Enrolled peer devices with the CA.
Identified and configured the protocol (such as IP Security [IPsec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

"crypto ca" to "crypto pki" CLI Change

All commands that begin as “crypto ca” have been changed to begin as “crypto pki.” Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

Information About Authorization and Revocation of Certificates

- PKI Authorization, page 32
- PKI and AAA Server Integration for Certificate Status, page 32
- CRLs or OCSP Server Choosing a Certificate Revocation Mechanism, page 34
- When to Use Certificate-Based ACLs for Authorization or Revocation, page 36
- PKI Certificate Chain Validation, page 38
- PKI High Availability Support, page 39

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.

There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server. (For more information on using certificate-based ACLs for authentication, see the section “When to Use Certificate-Based ACLs for Authorization or Revocation, page 36.”)

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

PKI and AAA Server Integration for Certificate Status

Integrating your PKI with an authentication, authorization, and accounting (AAA) server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of “all” from the AAA server provides authorization. Likewise,
a label of “none” from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but “none” is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be “ipsec,” “ssl,” or “osp.” (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

**Note**
Currently, no application component supports specification of the application label.

- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

- RADIUS or TACACS+ Choosing a AAA Server Protocol, page 33
- Attribute-Value Pairs for PKI and AAA Server Integration, page 33

### RADIUS or TACACS+ Choosing a AAA Server Protocol

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to “cisco,” which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute “cert-application=all” is added under the PKI service for the particular user or user group to authorize the specific username.

### Attribute-Value Pairs for PKI and AAA Server Integration

The table below lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

**Note**
Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The all parameter (within the authorization username command) specifies that the entire subject name of the certificate will be used as the authorization username.

<table>
<thead>
<tr>
<th>AV Pair</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>cisco-avpair=pki:cert-application=all</td>
<td>Valid values are “all” and “none.”</td>
</tr>
</tbody>
</table>
### AV Pair Value

**cisco-avpair=pki:cert-trustpoint=msca**  
The value is a Cisco IOS XE command-line interface (CLI) configuration trustpoint label.  
**Note** The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS XE router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.

**cisco-avpair=pki:cert-serial=16318DB7000100001671**  
The value is a certificate serial number.  
**Note** The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS XE router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.

**cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003**  
The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year.  
**Note** Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).

### CRLs or OCSP Server Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms--certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the PKI and AAA Server Integration for Certificate Status section.

The following sections explain how each revocation mechanism works:

- What Is a CRL, page 34
- What Is OCSP, page 35

### What Is a CRL
A certificate revocation list (CRL) is a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate for which the CA is responsible has been revoked. By default, a new CRL is downloaded after the currently cached CRL expires. An administrator may also configure the duration for which CRLs are cached in router memory or disable CRL caching completely. The CRL caching configuration applies to all CRLs associated with a trustpoint.

When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router does not know that the certificate has been revoked. The certificate passes the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device uses the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified through the `cdp-url` command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the security association (SA) and Internet Key Exchange (IKE) lifetimes.
- The CRL lifetime determines the length of time between CA-issued updates to the CRL. The default CRL lifetime value, which is 168 hours [1 week], can be changed through the `lifetime crl` command.
- The method of the CDP determines how the CRL is retrieved; some possible choices include HTTP, Lightweight Directory Access Protocol (LDAP), SCEP, or TFTP. HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
- The location of the CDP determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Tip

Although the Cisco IOS XE software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

What Is OCSP

OCSP is an online mechanism that is used to determine certificate validity and provides the following flexibility as a revocation mechanism:
OCSP can provide real-time certificate status checking.
OCSP allows the network administrator to specify a central OCSP server, which can service all devices within a network.
OCSP also allows the network administrator the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates.
OCSP server validation is usually based on the root CA certificate or a valid subordinate CA certificate, but may also be configured so that external CA certificates or self-signed certificates may be used. Using external CA certificates or self-signed certificates allows the OCSP servers certificate to be issued and validated from an alternative PKI hierarchy.

A network administrator can configure an OCSP server to collect and update CRLs from different CA servers. The devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question and an optional unique identifier for the OCSP request, or a nonce. The OCSP server holds a copy of the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer including the nonce. If the nonce in the response from the OCSP server does not match the original nonce sent by the peer, the response is considered invalid and certificate verification fails. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

When to Use an OCSP Server

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:
- Real-time certificate revocation status is necessary. CRLs are updated only periodically and the latest CRL may not always be cached by the client device. For example, if a client does not yet have the latest CRL cached and a newly revoked certificate is being checked, that revoked certificate will successfully pass the revocation check.
- There are a large number of revoked certificates or multiple CRLs. Caching a large CRL consumes large portions of Cisco IOS XE memory and may reduce resources available to other processes.
- CRLs expire frequently, causing the CDP to handle a larger load of CRLs.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components
such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value—equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

- Ignore Revocation Checks Using a Certificate-Based ACL, page 37

**Ignore Revocation Checks Using a Certificate-Based ACL**

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

**Ignoring Revocation Lists**

To allow a trustpoint to enforce CRLs except for specific certificates, enter the `match certificate` command with the `skip revocation-check` keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the `match certificate` command with the `skip revocation-check` keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

**Ignoring Expired Certificates**

To configure your router to ignore expired certificates, enter the `match certificate` command with the `allow expired-certificate` keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to “allow” the expired certificate until the peer can obtain a new certificate.
- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

**Note**

If Network Time Protocol (NTP) is available only via the IPSec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be “brought up” because the certificate of the hub is not yet valid.

- “Expired” is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.
Skipping the AAA Check of the Certificate

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the `match certificate` command with the `skip authorization-check` keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the `match certificate` command with the `skip authorization-check` keyword to skip the certificate check so that the tunnel can be established.

The `match certificate` command and the `skip authorization-check` keyword should be configured after PKI integration with an AAA server is configured.

---

**Note**

If the AAA server is available only via an IPSec connection, the AAA server cannot be contacted until after the IPSec connection is established. The IPSec connection cannot be “brought up” because the certificate of the AAA server is not yet valid.

---

PKI Certificate Chain Validation

A certificate chain establishes a sequence of trusted certificates --from a peer certificate to the root CA certificate. Within a PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Each CA corresponds to a trustpoint.

When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate, or trustpoint, is reached. In Cisco IOS XE Release 2.1 and later releases, an administrator may configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates.

Configuring the level to which a certificate chain is processed allows for the reauthentication of trusted certificates, the extension of a trusted certificate chain, and the completion of a certificate chain that contains a gap.

**Reauthentication of Trusted Certificates**

The default behavior is for the router to remove any trusted certificates from the certificate chain sent by the peer before the chain is validated. An administrator may configure certificate chain path processing so that the router does not remove CA certificates that are already trusted before chain validation, so that all certificates in the chain are re-authenticated for the current session.

**Extending the Trusted Certificate Chain**

The default behavior is for the router to use its trusted certificates to extend the certificate chain if there are any missing certificates in the certificate chain sent by the peer. The router will validate only certificates in the chain sent by the peer. An administrator may configure certificate chain path processing so that the certificates in the peer’s certificate chain and the router’s trusted certificates are validated to a specified point.

**Completing Gaps in a Certificate Chain**

An administrator may configure certificate chain processing so that if there is a gap in the configured Cisco IOS XE trustpoint hierarchy, certificates sent by the peer can be used to complete the set of certificates to be validated.
Note If the trustpoint is configured to require parent validation and the peer does not provide the full certificate chain, the gap cannot be completed and the certificate chain is rejected and invalid.

Note It is a configuration error if the trustpoint is configured to require parent validation and there is no parent trustpoint configured. The resulting certificate chain gap cannot be completed and the subordinate CA certificate cannot be validated. The certificate chain is invalid.

PKI High Availability Support

High Availability (HA) support is provided for the client-side PKI only. There is no CA server support. Route processor to route processor redundancy and Stateful Switchover (SSO) are supported.

In Service Software Upgrade (ISSU) and box-to-box HA are not supported.

How to Configure Authorization and Revocation of Certificates for Your PKI

- Configuring PKI Integration with a AAA Server, page 39
- Configuring a Revocation Mechanism for PKI Certificate Status Checking, page 44
- Configuring Certificate Authorization and Revocation Settings, page 47
- Configuring Certificate Chain Validation, page 55

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.
The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

- Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.
- Some AAA servers limit the available character set that may be used for the username (for example, a space [ ] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.
- The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.
- CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.
- Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS XE software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS XE software style (that is, with the least significant RDN first) is used.

```plaintext
or
radius-server host hostname [key string]
```

**SUMMARY STEPS**

1. enable
2. configure terminal
3. aaa new-model
4. aaa authorization network listname [method]
5. crypto pki trustpoint name
6. enrollment url url
7. revocation-check method
8. exit
9. authorization username subjectname subjectname
10. authorization list listname
11. tacacs-server host hostname [key string]
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
| **Example:** Router> enable |
| **Step 2** configure terminal | Enters global configuration mode. |
| **Example:** Router# configure terminal |
| **Step 3** aaa new-model | Enables the AAA access control model. |
| **Example:** Router(config)# aaa new-model |
| **Step 4** aaa authorization network listname [method] | Sets the parameters that restrict user access to a network.  
- *method* --Can be *group radius*, *group tacacs+*, or *group group-name*. |
| **Example:** Router (config)# aaa authorization network maxaaa group tacacs+ |
| **Step 5** crypto pki trustpoint name | Declares the trustpoint and a given name and enters ca-trustpoint configuration mode. |
| **Example:** Router (config)# crypto pki trustpoint msca |
| **Step 6** enrollment url url | Specifies the enrollment parameters of your CA.  
- The *url* argument is the URL of the CA to which your router should send certificate requests. |
| **Example:** Router (ca-trustpoint)# enrollment url http://caserver.myexample.com |
| **Step 7** revocation-check method | (Optional) Checks the revocation status of a certificate. |
| **Example:** Router (ca-trustpoint)# revocation-check crl |
### Command or Action | Purpose
--- | ---
#### Step 8 | exit
- **Example:**
  ```
  Router (ca-trustpoint)# exit
  ```
  Exits ca-trustpoint configuration mode and returns to global configuration mode.

#### Step 9 | authorization username subjectname subjectname
- **Example:**
  ```
  Router (config)# authorization username subjectname subjectname serialnumber
  ```
  Sets parameters for the different certificate fields that are used to build the AAA username.

  - The `subjectname` argument can be any of the following:
    - `all` -- Entire distinguished name (subject name) of the certificate.
    - `commonname` -- Certification common name.
    - `country` -- Certificate country.
    - `email` -- Certificate e-mail.
    - `ipaddress` -- Certificate IP address.
    - `locality` -- Certificate locality.
    - `organization` -- Certificate organization.
    - `organizationalunit` -- Certificate organizational unit.
    - `postalcode` -- Certificate postal code.
    - `serialnumber` -- Certificate serial number.
    - `state` -- Certificate state field.
    - `streetaddress` -- Certificate street address.
    - `title` -- Certificate title.
    - `unstructuredname` -- Certificate unstructured name.

#### Step 10 | authorization list listname
- **Example:**
  ```
  Route (config)# authorization list maxaaa
  ```
  Specifies the AAA authorization list.
### Command or Action

**Step 11**  
**tacacs-server host hostname [key string]**

**Example:**

```bash
or
```

**Example:**

```bash
radius-server host hostname [key string]
```

**Example:**

```bash
Router(config)# tacacs-server host 192.0.2.2  
key a_secret_key
```

**Example:**

```bash
or
```

**Example:**

```bash
Router(config)# radius-server host 192.0.2.1  
key another_secret_key
```

### Purpose

- Specifies a TACACS+ host.
- or
- Specifies a RADIUS host.

---

**Troubleshooting Tips, page 43**

**Troubleshooting Tips**

To display debug messages for the trace of interaction (message type) between the CA and the router, use the `debug crypto pki transactions` command. (See the sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)

**Successful Exchange**

```bash
Router# debug crypto pki transactions  
Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match  
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.  
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check
```

Each line that shows “CRYPTO_PKI_AAA” indicates the state of the AAA authorization checks. Each of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

```bash
Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipsecca_script_aalist,  
PKIAAA-L, <all>)  
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")  
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")  
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "15DE")  
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed  
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```
Failed Exchange

Router# debug crypto pki transactions
Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "233D")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30:00
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

Perform this task to set up a CRL as the certificate revocation mechanism--CRLs or OCSP--that is used to check the status of certificates in a PKI.

- The revocation-check Command, page 44
- Nonces and Peer Communications with OCSP Servers, page 44

The revocation-check Command

Use the revocation-check command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer’s certificate--unless you include the none keyword in your configuration. If the none keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Nonces and Peer Communications with OCSP Servers

When using OCSP, nonces, unique identifiers for OCSP requests, are sent by default during peer communications with your OCSP server. The use of nonces offers a more secure and reliable communication channel between the peer and OCSP server.

If your OCSP server does not support nonces, you may disable the sending of nonces. For more information, check your OCSP server documentation.

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.
Note

- OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.
- If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Step 3 crypto pki trustpoint name</td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
</tbody>
</table>

**SUMMARY STEPS**

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. ocs p url url
5. revocation-check method1 [method2 method3]
6. ocs p disable-nonce
7. exit
8. exit
9. show crypto pki certificates
10. show crypto pki trustpoints [status | label [status]]
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4** ocsp url url | (Optional) Specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL will override the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate.  
Example:  
Router(ca-trustpoint)# ocsp url http://ocsp-server |
| **Step 5** revocation-check method1 [method2 method3] | Checks the revocation status of a certificate.  
- **crl** --Certificate checking is performed by a CRL. This is the default option.  
- **none** --Certificate checking is ignored.  
- **ocsp** --Certificate checking is performed by an OCSP server.  
Example:  
Router(ca-trustpoint)# revocation-check ocsp none  
If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down. |
| **Step 6** ocsp disable-nonce | (Optional) Specifies that a nonce, or an OCSP request unique identifier, will not be sent during peer communications with the OCSP server.  
Example:  
Router(ca-trustpoint)# ocsp disable-nonce |
| **Step 7** exit | Returns to global configuration mode.  
Example:  
Router(ca-trustpoint)# exit |
| **Step 8** exit | Returns to privileged EXEC mode.  
Example:  
Router(config)# exit |
| **Step 9** show crypto pki certificates | (Optional) Displays information about your certificates.  
Example:  
Router# show crypto pki certificates |
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>show crypto pki trustpoints [status</td>
<td>label</td>
</tr>
</tbody>
</table>

**Example:**

Router# show crypto pki trustpoints

### Configuring Certificate Authorization and Revocation Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, to manually override the default CDP location, to manually override the OCSP server setting, to configure CRL caching, or to set session acceptance or rejection based on a certificate serial number, as appropriate.

- Configuring Certificate-Based ACLs to Ignore Revocation Checks, page 47
- Manually Overriding CDPs in a Certificate, page 47
- Manually Overriding the OCSP Server Setting in a Certificate, page 48
- Configuring CRL Cache Control, page 48
- Configuring Certificate Serial Number Session Control, page 48
- Examples, page 54
- Troubleshooting Tips, page 55

### Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the **match certificate** command and **skip revocation-check** keyword.
- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.
- Define a certificate map to match the characteristics identified in the prior step.
- You can add the **match certificate** command and **skip revocation-check** keyword and the **match certificate command** and **allow expired-certificate** keyword to the trustpoint that was created or identified in the first step.

**Note**

Certificate maps are checked even if the peer’s public key is cached. For example, when the public key is cached by the peer, and a certificate map is added to the trustpoint to ban a certificate, the certificate map is effective. This prevents a client with the banned certificate, which was once connected in the past, from reconnecting.

### Manually Overriding CDPs in a Certificate
Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate’s CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

**Manually Overriding the OCSP Server Setting in a Certificate**

Administrators can override the OCSP server setting specified in the Authority Information Access (AIA) field of the client certificate or set by the issuing the `ocsp url` command. One or more OCSP servers may be manually specified, either per client certificate or per group of client certificates by the `match certificate override ocsp` command. The `match certificate override ocsp` command overrides the client certificate AIA field or the `ocsp url` command setting if a client certificate is successfully matched to a certificate map during the revocation check.

**Note**

Only one OCSP server can be specified per client certificate.

**Configuring CRL Cache Control**

By default, a new CRL will be downloaded after the currently cached CRL expires. Administrators can either configure the maximum amount of time in minutes a CRL remains in the cache by issuing the `crl cache delete-after` command or disable CRL caching by issuing the `crl cache none` command. Only the `crl-cache delete-after` command or the `crl-cache none` command may be specified. If both commands are entered for a trustpoint, the last command executed will take effect and a message will be displayed.

Neither the `crl-cache none` command nor the `crl-cache delete-after` command affects the currently cached CRL. If you configure the `crl-cache none` command, all CRLs downloaded after this command is issued will not be cached. If you configure the `crl-cache delete-after` command, the configured lifetime will only affect CRLs downloaded after this command is issued.

This functionality is useful when a CA issues CRLs with no expiration date or with expiration dates days or weeks ahead.

**Configuring Certificate Serial Number Session Control**

A certificate serial number can be specified to allow a certificate validation request to be accepted or rejected by the trustpoint for a session. A session may be rejected, depending on certificate serial number session control, even if a certificate is still valid. Certificate serial number session control may be configured by using either a certificate map with the `serial-number` field or an AAA attribute, with the `cert-serial-not` command.

Using certificate maps for session control allows an administrator to specify a single certificate serial number. Using the AAA attribute allows an administrator to specify one or more certificate serial numbers for session control.

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
- The certificate map must be configured before the CDP override feature can be enabled or the `serial-number` command is issued.
- The PKI and AAA server integration must be successfully completed to use AAA attributes as described in “PKI and AAA Server Integration for Certificate Status, page 32.”
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki certificate map label sequence-number
4. field-name match-criteria match-value
5. exit
6. crypto pki trustpoint name
7. Do one of the following:
   - crl-cache none
   - crl-cache delete-after time
8. match certificate certificate-map-label [allow expired-certificate | skip revocation-check | skip authorization-check]
9. match certificate certificate-map-label override cdp {url | directory} string
10. match certificate certificate-map-label override ocsp [trustpoint trustpoint-label] sequence-number url ocsp-url
11. exit
12. aaa new-model
13. aaa attribute list list-name
14. attribute type name value
15. exit
16. exit
17. show crypto pki certificates

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td>Step 2 configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki certificate map <strong>label</strong></td>
<td>Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.</td>
</tr>
<tr>
<td><strong>sequence-number</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto pki certificate map Group 10</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <strong>field-name</strong> match-criteria match-value</td>
<td>Specifies one or more certificate fields together with their matching criteria and the value to match.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-certificate-map)#</td>
<td></td>
</tr>
<tr>
<td><strong>subject-name</strong> co MyExample</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <strong>exit</strong></td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-certificate-map)# exit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>crypto pki trustpoint name</strong></td>
</tr>
<tr>
<td></td>
<td>Declares the trustpoint, given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto pki trustpoint Access2</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>* crl-cache none</td>
</tr>
<tr>
<td></td>
<td>* crl-cache delete-after time</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ca-trustpoint)# crl-cache none</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ca-trustpoint)# crl-cache delete-after 20</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>**match certificate certificate-map-label [allow expired-certificate</td>
</tr>
<tr>
<td></td>
<td>(Optional) Associates the certificate-based ACL (that was defined via the crypto pki certificate map command) to a trustpoint.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ca-trustpoint)# match certificate Group skip revocation-check</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 9</strong> match certificate certificate-map-label override cdp [url</td>
<td>directory] string</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# match certificate Group1 override cdp url <a href="http://server.cisco.com">http://server.cisco.com</a></td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> match certificate certificate-map-label override ocsp [trustpoint trustpoint-label] sequence-number url ocsp-url</td>
<td>(Optional) Specifies an OCSP server, either per client certificate or per group of client certificates, and may be issued more than once to specify additional OCSP servers and client certificate settings including alternative PKI hierarchies.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# match certificate mycertmapname override ocsp trustpoint mytp 15 url <a href="http://192.0.2.2">http://192.0.2.2</a></td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> exit</td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# exit</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 12</strong> aaa new-model</td>
<td>(Optional) Enables the AAA access control model.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# aaa new-model</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> aaa attribute list list-name</td>
<td>(Optional) Defines an AAA attribute list locally on a router and enters config-attr-list configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# aaa attribute list crl</td>
<td></td>
</tr>
<tr>
<td><strong>Step 14</strong> attribute type name value</td>
<td>(Optional) Defines an AAA attribute type that is to be added to an AAA attribute list locally on a router.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-attr-list)# attribute type cert-serial-not 6C4A</td>
<td>To configure certificate serial number session control, an administrator may specify a specific certificate in the value field to be accepted or rejected based on its serial number where name is set to cert-serial-not. If the serial number of the certificate matches the serial number specified by the attribute type setting, the certificate will be rejected. For a full list of available AAA attribute types, execute the show aaa attributes command.</td>
</tr>
<tr>
<td><strong>Step 15</strong> exit</td>
<td>Returns to global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config-attr-list)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 16</strong> exit</td>
<td>Returns to privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 17</strong> show crypto pki certificates</td>
<td>(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# show crypto pki certificates</td>
<td></td>
</tr>
</tbody>
</table>
Examples

The following is a sample certificate. The OCSP-related extensions are shown using exclamation points.

Certificate:

Data:

Version: v3
Serial Number: 0x14
Signature Algorithm: MD5withRSA - 1.2.840.113549.1.1.4
Issuer: CN=CA server, OU=PKI, O=Cisco Systems
Validity:
   Not Before: Thursday, August 8, 2002 4:38:05 PM PST
   Not After: Tuesday, August 7, 2003 4:38:05 PM PST
Subject: CN=OCSP server, OU=PKI, O=Cisco Systems
Subject Public Key Info:
   Algorithm: RSA - 1.2.840.113549.1.1.1
   Public Key:
      Exponent: 65537
      Public Key Modulus: (1024 bits):
         <snip>

Extensions:

Identifier: Subject Key Identifier - 2.5.29.14
   Critical: no
   Key Identifier:
      <snip>

Identifier: Authority Key Identifier - 2.5.29.35
   Critical: no
   Key Identifier:
      <snip>

! Identifier: OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
   Critical: no

Identifier: Extended Key Usage: - 2.5.29.37
   Critical: no
   Extended Key Usage:
      OCSPSigning
!

Identifier: CRL Distribution Points - 2.5.29.31
   Critical: no
   Number of Points: 1
   Point 0
      Distribution Point:
         [URIName: ldap://CA-server/CN=CA server, OU=PKI, O=Cisco Systems]
         Signature:
            Algorithm: MD5withRSA - 1.2.840.113549.1.1.4
            Signature:
               <snip>

The following example shows an excerpt of the running configuration output when adding a **match certificate override ocsp** command to the beginning of an existing sequence:

```
mesh certificate map3 override ocsp 5 url http://192.0.2.3/
show running-configuration
```

The following example shows an excerpt of the running configuration output when an existing **match certificate override ocsp** command is replaced and a trustpoint is specified to use an alternative PKI hierarchy:

```
mesh certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
```
match certificate map3 override ocsp trustpoint tp3 5 url http://192.0.2.3/
match certificate map1 override ocsp trustpoint tp1 10 url http://192.0.2.1/
match certificate map4 override ocsp trustpoint tp4 10 url http://192.0.2.4/newvalue
match certificate map2 override ocsp trustpoint tp2 15 url http://192.0.2.2/

Troubleshooting Tips

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches either the certificate or certificates that should be allowed or the AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

Configuring Certificate Chain Validation

Perform this task to configure the processing level for the certificate chain path of your peer certificates.

- The device must be enrolled in your PKI hierarchy.
- The appropriate key pair must be associated with the certificate.

Note

- A trustpoint associated with the root CA cannot be configured to be validated to the next level.

The `chain-validation` command is configured with the `continue` keyword for the trustpoint associated with the root CA, an error message will be displayed and the chain validation will revert to the default `chain-validation` command setting.

> 

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. chain-validation [{stop | continue} [parent-trustpoint]]
5. exit

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td>- Enter your password if prompted.</td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
---|---
**Step 2** configure terminal | Enters global configuration mode.
**Example:**
Router# configure terminal

**Step 3** crypto pki trustpoint *name* | Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
**Example:**
Router(config)# crypto pki trustpoint ca-sub1

**Step 4** chain-validation [{stop | continue} [parent-trustpoint]] | Configures the level to which a certificate chain is processed on all certificates including subordinate CA certificates.
- Use the *stop* keyword to specify that the certificate is already trusted. This is the default setting.
- Use the *continue* keyword to specify that the the subordinate CA certificate associated with the trustpoint must be validated.
- The *parent-trustpoint* argument specifies the name of the parent trustpoint the certificate must be validated against.
**Example:**
Router(ca-trustpoint)# chain-validation continue ca-sub1

**Step 5** exit | Returns to global configuration mode
**Example:**
Router(ca-trustpoint)# exit

---

**Configuration Examples for Setting Up Authorization and Revocation of Certificates**

**Examples Configuring and Verifying PKI AAA Authorization**

**Example Router Configuration**

The following `show running-config` command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config
Building configuration...
!
version 12.3
!```
hostname router7200
!

aaa new-model
!

aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa accounting exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name example.com
!
crypto pki trustpoint EM-CERT-SERV
enrollment url http://192.0.2.33:80
serial-number
rsakeypair STOREVPN 1024
auto-enroll
authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
certificate 04
30820214 3082017D A039D201 02020104 300D0609 2A864886 F70D0101 04050030
17311530 13060350 0403130C 4542DD43 4552542D 53455256 301E170D 30340301
31393232 30323535 5A170D30 35303131 38323233 32353535 3033312E 300E0603
55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230320D
312E6772 6962C6E3 6F6D3081 9F300D06 02A86486 86F70D01 01010500 03818D00
30B18907 38323233 5A170D30 02823237 31890D00 5AF39912 284D4D14 9C2E3A03 5A720C34 92F8D6A8
7D3257A6 BCC8596F A38A9B10 47435626 D5A8F2A2 123195BB 81AA5E0
5CA162FF 8F3C3A4F 8B3E9F27 B031642 6B18AEB1 40F2E3B4 F966BEFE 382C7283
3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2B29 704BD68F
FD904043 0F555702 03010001 A3573505 30250603 551D1F04 1E3301C30 1AA018A0
16861468 7474703A 2FF3633 2E323437 3E313337 3E313330 0B8E6350 1D0F0404
02005A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AF4D67
341A6E12 1D300D06 092A8648 86F70D01 01040500 30818001 79E97018 FB955108
12F42A56 2A6384BC ACBE22FE FD16187F DA56737 0CE241AC 039EC75D 3C743F59
08EDEEFF 08EB13A7 37D9E0FA 09D62CD0 08E798CD 2C1DCEBC 3B2505A1 3897330C
15A60D5A 8A13F06D 51043D37 0E6345DF A65F43D7 4E836903 696798A4 04FD61D
EC11F160C 1ABC8D03 49FB11B1 DA0B8D6C 463E0190 F34C59E4
quit
certificate ca 01
03082020 30820170 A039D201 02020104 300D0609 2A864886 F70D0101 04050030
17311530 13060350 0403130C 4542DD43 4552542D 53455256 301E170D 30333132
31393232 30323535 5A170D30 35303131 38323233 32353535 3033312E 300E0603
55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230320D
312E6772 6962C6E3 6F6D3081 9F300D06 02A86486 86F70D01 01010500 03818D00
30B18907 38323233 5A170D30 02823237 31890D00 5AF39912 284D4D14 9C2E3A03 5A720C34 92F8D6A8
7D3257A6 BCC8596F A38A9B10 47435626 D5A8F2A2 123195BB 81AA5E0
5CA162FF 8F3C3A4F 8B3E9F27 B031642 6B18AEB1 40F2E3B4 F966BEFE 382C7283
3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2B29 704BD68F
FD904043 0F555702 03010001 A3573505 30250603 551D1F04 1E3301C30 1AA018A0
16861468 7474703A 2FF3633 2E323437 3E313337 3E313330 0B8E6350 1D0F0404
02005A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AF4D67
341A6E12 1D300D06 092A8648 86F70D01 01040500 30818001 79E97018 FB955108
12F42A56 2A6384BC ACBE22FE FD16187F DA56737 0CE241AC 039EC75D 3C743F59
08EDEEFF 08EB13A7 37D9E0FA 09D62CD0 08E798CD 2C1DCEBC 3B2505A1 3897330C
15A60D5A 8A13F06D 51043D37 0E6345DF A65F43D7 4E836903 696798A4 04FD61D
EC11F160C 1ABC8D03 49FB11B1 DA0B8D6C 463E0100 F34C59E4
quit
crypto isakmp policy 10encr 3des
group 2
!
crypto ipsec transform-set ISC_TS_1 esp-3des esp-sha-hmac

crypto ipsec profile ISC_IPSEC_PROFILE_2
Example Debug of a Successful PKI AAA Authorization

The following `show debugging` command output shows a successful authorization using the PKI Integration with AAA Server feature:

```
Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on
Router# May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
```
Example Debugs of a Failed PKI AAA Authorization

The following show debugging command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

Example Debugs of a Failed PKI AAA Authorization

The following show debugging command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

Example Debugs of a Failed PKI AAA Authorization

The following show debugging command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.
Example Configuring an OCSP Server

The following example shows how to configure the router to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

Example Specifying a CRL and Then an OCSP Server

The following example shows how to configure the router to download the CRL from the CDP. If the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Example Specifying an OCSP Server

The following example shows how to configure your router to use the OCSP server at the HTTP URL “http://myocspserver:81.” If the server is down, the revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```
Example Disabling Nonces in Communications with the OCSP Server

The following example shows communications when a nonce, or a unique identifier for the OCSP request, is disabled for communications with the OCSP server:

Router(config)# crypto pki trustpoint mytp
Router(config)# ocsp url http://myocspserver:81
Router(config)# revocation-check ocsp none
Router(config)# ocsp disable-nonce

Example Configuring a Hub Router at a Central Site for Certificate Revocation Checks

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration—only the PKI-related configuration is shown.

Home Office Hub Configuration

crypto pki trustpoint VPN-GW
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Central VPN Gateway
revocation-check crl

Central Site Hub Router

Router# show crypto ca certificate
Certificate
Status: Available
Certificate Serial Number: 2F62BE140000000000CA0
Certificate Usage: General Purpose
Issuer: cn=Central Certificate Authority
o=Home Office Inc
Subject: Name: Central VPN Gateway
  cn=Central VPN Gateway
  o=Home Office Inc
CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl
Validity Date:
  start date: 00:43:26 GMT Sep 26 2003
  end date: 00:53:26 GMT Sep 26 2004
  renew date: 00:00:00 GMT Jan 1 1970
Associated Trustpoints: VPN-GW
CA Certificate
Status: Available
Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
Certificate Usage: Signature
Issuer: cn=Central Certificate Authority
  o=Home Office Inc
Subject:
  cn=Central Certificate Authority
  o=Home Office Inc
CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl
Validity Date:
  start date: 22:19:29 GMT Oct 31 2002
  end   date: 22:27:27 GMT Oct 31 2017
Associated Trustpoints: VPN-GW

Trustpoint on the Branch Office Router

crypto pki trustpoint home-office
  enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
  serial-number none
  fqdn none
  ip-address none
  subject-name o=Home Office Inc,cn=Branch 1
  revocation-check crl

A certificate map is entered on the branch office router.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#

The output from the show certificate command on the central site hub router shows that the certificate was
issued by the following:

cn=Central Certificate Authority
  o=Home Office Inc

These two lines are combined into one line using a comma (,) to separate them, and the original lines are
added as the first criteria for a match.

Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office Inc
!The above line wrapped but should be shown on one line with the line above it.

The same combination is done for the subject name from the certificate on the central site router (note that
the line that begins with “Name:” is not part of the subject name and must be ignored when creating the
certificate map criteria). This is the subject name to be used in the certificate map.

cn=Central VPN Gateway
  o=Home Office Inc

Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
Now the certificate map is added to the trustpoint that was configured earlier.

Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (config)# exit

The configuration is checked (most of configuration is not shown).

Router# write term
!Many lines left out
  .
  .
  .
crypto pki trustpoint home-office
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Branch 1
revocation-check crl
match certificate central-site skip revocation-check
!
crypto pki certificate map central-site 10
issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
subject-name eq cn = central vpn gateway, o = home office inc

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

crypto pki trustpoint home-office
  auth list allow_list
  auth user subj commonname

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

match certificate central-site skip authorization-check

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the match certificate command and central-site skip authorization-check (argument and keyword), the branch office cannot establish the tunnel until it has checked the CRL or the AAA server. (The tunnel will not be established unless the match certificate command and central-site skip authorization-check argument and keyword are used.)

The match certificate command and allow expired-certificate keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

crypto pki trustpoint VPN-GW
  enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
  serial-number none
  fqdn none
  ip-address none
  subject-name o=Home Office Inc,cn=Central VPN Gateway
  revocation-check crl

Trustpoint on the Branch 1 Site Router

Router# show crypto ca certificate
Certificate
  Status: Available
  Certificate Serial Number: 2F62BE1400000000000CA0
  Certificate Usage: General Purpose
  Issuer:
    cn=Central Certificate Authority
    o=Home Office Inc
  Subject:
    Name: Branch 1 Site
    cn=Branch 1 Site
    o=Home Office Inc
  CRL Distribution Points:
    http://ca.home-office.com/CertEnroll/home-office.crl
  Validity Date:
    start date: 00:43:26 GMT Sep 26 2003
end date: 00:53:26 GMT Oct 3 2003
renew date: 00:00:00 GMT Jan 1 1970
Associated Trustpoints: home-office
CA Certificate
Status: Available
Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
Certificate Usage: Signature
Issuer:
cn=Central Certificate Authority
o=Home Office Inc
Subject:
cn=Central Certificate Authority
o=Home Office Inc
CRL Distribution Points:
http://ca.home-office.com/CertEnroll/home-office.crl
Validity Date:
start date: 22:19:29 GMT Oct 31 2002
end date: 22:27:27 GMT Oct 31 2017
Associated Trustpoints: home-office

A certificate map is entered on the central site router.

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office Inc
The above line wrapped but should be part of the line above it.
Router (ca-certificate-map)# subject-name eq cn=Branch 1 Site,o=home office inc
The certificate map is added to the trustpoint.

Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config)# exit
The configuration should be checked (most of the configuration is not shown).

Router# write term
!many lines left out
crypto pki trustpoint VPN-GW
   enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
   serial-number none
   fqdn none
   ip-address none
   subject-name o=Home Office Inc,cn=Central VPN Gateway
   revocation-check crl
   match certificate branch1 allow expired-certificate
!crypto pki certificate map central-site 10
   issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
   subject-name eq cn = central vpn gateway, o = home office inc
!many lines left out
The match certificate command and branch1 allow expired-certificate (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

---

Examples Configuring Certificate Authorization and Revocation Settings

Example Configuring CRL Cache Control

The following example shows how to disable CRL caching for all CRLs associated with the CA1 trustpoint:

crypto pki trustpoint CA1
enrollment url http://CA1:80
ip-address FastEthernet0/0
crl query ldap://ldap_CA1
revocation-check crl
crl-cache none

The current CRL is still cached immediately after executing the example configuration shown above:

Router# show crypto pki crls

CRL Issuer Name:
  cn=name Cert Manager,ou=pki,o=example.com,c=US
  LastUpdate: 18:57:42 GMT Nov 26 2005
  NextUpdate: 22:57:42 GMT Nov 26 2005
  Retrieved from CRL Distribution Point:
    ldap://ldap.example.com/CN=name Cert Manager,O=example.com

When the current CRL expires, a new CRL is then downloaded to the router at the next update. The crl-cache none command takes effect and all CRLs for the trustpoint are no longer cached; caching is disabled. You can verify that no CRL is cached by executing the show crypto pki crls command. No output will be shown because there are no CRLs cached.

The following example shows how to configure the maximum lifetime of 2 minutes for all CRLs associated with the CA1 trustpoint:

crypto pki trustpoint CA1
enrollment url http://CA1:80
ip-address FastEthernet0/0
crl query ldap://ldap_CA1
revocation-check crl
crl-cache delete-after 2

The current CRL is still cached immediately after executing the example configuration above for setting the maximum lifetime of a CRL:

Router# show crypto pki crls

CRL Issuer Name:
  cn=name Cert Manager,ou=pki,o=example.com,c=US
  LastUpdate: 18:57:42 GMT Nov 26 2005
  NextUpdate: 22:57:42 GMT Nov 26 2005
  Retrieved from CRL Distribution Point:
    ldap://ldap.example.com/CN=name Cert Manager,O=example.com

When the current CRL expires, a new CRL is downloaded to the router at the next update and the crl-cache delete-after command takes effect. This newly cached CRL and all subsequent CRLs will be deleted after a maximum lifetime of 2 minutes.

You can verify that the CRL will be cached for 2 minutes by executing the show crypto pki crls command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.

Router# show crypto pki crls

CRL Issuer Name:
  cn=name Cert Manager,ou=pki,o=example.com,c=US
  LastUpdate: 18:57:42 GMT Nov 26 2005
  NextUpdate: 22:57:42 GMT Nov 26 2005
  Retrieved from CRL Distribution Point:
    ldap://ldap.example.com/CN=name Cert Manager,O=example.com

Example Configuring Certificate Serial Number Session Control
The following example shows the configuration of certificate serial number session control using a certificate map for the CA1 trustpoint:

crypto pki trustpoint CA1
  enrollment url http://CA1
  chain-validation stop
  crl query ldap://ldap_server
  revocation-check crl
  match certificate crl

crypto pki certificate map crl 10
  serial-number co 279d

Note
If the match-criteria value is set to eq (equal) instead of co (contains), the serial number must match the certificate map serial number exactly, including any spaces.

The following example shows the configuration of certificate serial number session control using AAA attributes. In this case, all valid certificates will be accepted if the certificate does not have the serial number “4ACA.”

crypto pki trustpoint CA1
  enrollment url http://CA1
  ip-address FastEthernet0/0
  crl query ldap://ldap_CA1
  revocation-check crl
  aaa new-model
  !
  aaa attribute list crl
  attribute-type aaa-cert-serial-not 4ACA

The server log shows that the certificate with the serial number “4ACA” was rejected. The certificate rejection is shown using exclamation points.

```
Dec 3 04:24:39.051: CRYPTO_PKI: Trust-Point CA1 picked up
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.051: CRYPTO_PKI: unlocked trustpoint CA1, refcount is 0
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.135: CRYPTO_PKI: Found a issuer match
Dec 3 04:24:39.135: CRYPTO_PKI: Using CA1 to validate certificate
Dec 3 04:24:39.135: CRYPTO_PKI: checking AAA authorization (CRL, PKIAAA-L1, <all>)
Dec 3 04:24:39.135: CRYPTO_PKI: pre-authorization chain validation status (0x4)
Dec 3 04:24:39.135: AAA/AUTH(0x21): Pick method list 'CRL'

Dec 3 04:24:39.175: CRYPTO_PKI: AAA: reply attribute ("cert-application" = "all")
Dec 3 04:24:39.175: CRYPTO_PKI: AAA: reply attribute ("cert-trustpoint" = "CA1")
Dec 3 04:24:39.175: CRYPTO_PKI: AAA: reply attribute ("cert-serial-not" = "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI: cert-serial doesn't match ("4ACA" != "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI: AAA: post-authorization chain validation status (0x7)
Dec 3 04:24:39.175: CRYPTO_PKI: chain cert was anchored to trustpoint CA1, and chain validation result was: CRYPTO_PKI_CERT_NOTAUTHORIZED
```
Examples Configuring Certificate Chain Validation

Example Configuring Certificate Chain Validation from Peer to Root CA

In the following configuration example, all of the certificates will be validated—the peer, SubCA11, SubCA1, and RootCA certificates.

```plaintext
crypto pki trustpoint RootCA
  enrollment terminal
  chain-validation stop
  revocation-check none
  rsa-keypair RootCA

crypto pki trustpoint SubCA1
  enrollment terminal
  chain-validation continue RootCA
  revocation-check none
  rsa-keypair SubCA1

crypto pki trustpoint SubCA11
  enrollment terminal
  chain-validation continue SubCA1
  revocation-check none
  rsa-keypair SubCA11
```

Example Configuring Certificate Chain Validation from Peer to Subordinate CA

In the following configuration example, the following certificates will be validated—the peer and SubCA1 certificates.

```plaintext
crypto pki trustpoint RootCA
  enrollment terminal
  chain-validation stop
  revocation-check none
  rsa-keypair RootCA

crypto pki trustpoint SubCA1
  enrollment terminal
  chain-validation continue RootCA
  revocation-check none
  rsa-keypair SubCA1

crypto pki trustpoint SubCA11
  enrollment terminal
  chain-validation continue SubCA1
  revocation-check none
  rsa-keypair SubCA11
```

Example Configuring Certificate Chain Validation Through a Gap

In the following configuration example, SubCA1 is not in the configured Cisco IOS XE hierarchy but is expected to have been supplied in the certificate chain presented by the peer.

If the peer supplies the SubCA1 certificate in the presented certificate chain, the following certificates will be validated—the peer, SubCA11, and SubCA1 certificates.
If the peer does not supply the SubCA1 certificate in the presented certificate chain, the chain validation will fail.

crypto pki trustpoint RootCA
  enrollment terminal
  chain-validation stop
  revocation-check none
  rsakeypair RootCA

crypto pki trustpoint SubCA1
  enrollment terminal
  chain-validation continue RootCA
  revocation-check none
  rsakeypair SubCA1

Additional References

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples</td>
<td><em>Cisco IOS Security Command Reference</em></td>
</tr>
<tr>
<td>Overview of PKI, including RSA keys, certificate enrollment, and CAs</td>
<td>“Cisco IOS XE PKI Overview: Understanding and Planning a PKI” module in the <em>Cisco IOS XE Security Configuration Guide: Secure Connectivity</em></td>
</tr>
<tr>
<td>RSA key generation and deployment</td>
<td>“Deploying RSA Keys Within a PKI” module in the <em>Cisco IOS XE Security Configuration Guide: Secure Connectivity</em></td>
</tr>
<tr>
<td>Certificate enrollment: supported methods, enrollment profiles, configuration tasks</td>
<td>“Configuring Certificate Enrollment for a PKI” module in the <em>Cisco IOS XE Security Configuration Guide: Secure Connectivity</em></td>
</tr>
</tbody>
</table>

Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>
RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>

**Feature Information for Certificate Authorization and Revocation**

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.
Table 3  Feature Information for PKI Certificate Authorization and Revocation

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| Cache Control Enhancements for Certification Revocation Lists | Cisco IOS XE Release 2.4 | This feature provides users the ability to disable CRL caching or to specify the maximum lifetime for which a CRL will be cached in router memory. It also provides functionality to configure certificate serial number session control. The following sections provide information about this feature:  
  • What Is a CRL  
  • Configuring Certificate Authorization and Revocation Settings  
  • Examples: Configuring Certificate Authorization and Revocation Settings  
  The following commands were introduced or modified by this feature: crl-cache delete-after, crl-cache none, crypto pki certificate map |
| Certificate-Complete Chain Validation | Cisco IOS XE Release 2.4 | This feature provides users the ability to configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates. The following sections provide information about this feature:  
  • PKI Certificate Chain Validation  
  • Configuring Certificate Chain Validation  
  • Examples: Configuring Certificate Chain Validation  
  The following command was introduced by this feature: chain-validation |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| OCSP - Server Certification from Alternate Hierarchy   |                   | This feature provides users with the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates, and provides the capability for OCSP server validation based on external CA certificates or self-signed certificates. The following sections provide information about this feature:  
  - What Is OCSP  
  - Configuring Certificate Authorization and Revocation Settings  
  The following command was introduced by this feature: `match certificate override ocs` |
| Optional OCSP Nonce                                     | Cisco IOS XE Release 2.1 | This feature provides users with the ability to configure the sending of a nonce, or unique identifier for an OCSP request, during OCSP communications. The following sections provide information about this feature:  
  - What Is OCSP  
  - Configuring a Revocation Mechanism for PKI Certificate Status Checking  
  - Example: Disabling Nonces in Communications with the OCSP Server |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| Certificate Security Attribute-Based Access Control | Cisco IOS XE Release 2.1 | Under the IPsec protocol, CA interoperability permits Cisco IOS devices and a CA to communicate so that the Cisco IOS device can obtain and use digital certificates from the CA. Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action. This feature adds fields to the certificate that allow specifying an ACL, creating a certificate-based ACL.  
The following sections provide information about this feature:  
  • When to Use Certificate-Based ACLs for Authorization or Revocation  
  • Configuring Certificate Authorization and Revocation Settings
 |

The following commands were introduced or modified by this feature: `crypto pki certificate map`, `crypto pki trustpoint match certificate`
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| Online Certificate Status Protocol (OCSP)        | Cisco IOS XE Release 2.1 | This feature allows users to enable OCSP instead of CRLs to check certificate status. Unlike CRLs, which provide only periodic certificate status, OCSP can provide timely information regarding the status of a certificate. The following sections provide information about this feature:  
  - CRLs or OCSP Server Choosing a Certificate Revocation Mechanism  
  - Configuring a Revocation Mechanism for PKI Certificate Status Checking  
  The following commands were introduced by this feature: `ocsp url`, `revocation-check` |
| PKI AAA Authorization Using the Entire Subject Name | Cisco IOS XE Release 2.1 | This feature provides users with the ability to query the AAA server using the entire subject name from the certificate as a unique AAA username.  
  The following sections provide information about this feature:  
  - Attribute-Value Pairs for PKI and AAA Server Integration  
  - Configuring PKI Integration with a AAA Server  
  The following command was modified by this feature: `authorization username` |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
</table>
| PKI Integration with AAA Server      | Cisco IOS XE Release 2.1 | This feature provides additional scalability for authorization by generating a AAA username from the certificate presented by the peer. A AAA server is queried to determine whether the certificate is authorized for use by the internal component. The authorization is indicated by a component-specified label that must be present in the AV pair for the user. The following sections provide information about this feature:  
  - PKI and AAA Server Integration for Certificate Status  
  - Configuring PKI Integration with a AAA Server  
  The following commands were introduced by this feature:  
    **authorization list, authorization username** }
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Software Releases</th>
<th>Feature Configuration Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKI: Query Multiple Servers During Certificate</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature introduces the ability for Cisco IOS XE software to make multiple attempts to retrieve the CRL, allowing operations to continue when a particular server is not available. In addition, the ability to override the CDPs in a certificate with a manually configured CDP has been introduced. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate’s CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP. The following sections provide information about this feature:</td>
</tr>
<tr>
<td>Revocation Check</td>
<td></td>
<td>• Querying All CDPs During Revocation Check</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Manually Overriding CDPs in a Certificate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following command was introduced by this feature: <code>match certificate override cdp</code></td>
</tr>
<tr>
<td>Feature Name</td>
<td>Software Releases</td>
<td>Feature Configuration Information</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------</td>
</tr>
</tbody>
</table>
| Using Certificate ACLs to Ignore Revocation Check and Expired Certificates | Cisco IOS XE Release 2.1 | This feature allows a certificate that meets specified criteria to be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. Certificate ACLs are used to specify the criteria that the certificate must meet to be accepted or to avoid revocation checking. In addition, if AAA communication is protected by a certificate, this feature provides for the AAA checking of the certificate to be ignored. The following sections provide information about this feature:  
  - Ignore Revocation Checks Using a Certificate-Based ACL  
  - Configuring Certificate-Based ACLs to Ignore Revocation Checks  
  The following command was modified by this feature: `match certificate` |
| Query Mode Definition Per Trustpoint              | Cisco IOS XE Release 2.1 | This feature was integrated into Cisco IOS XE Release 2.1. |
| PKI High Availability                             | Cisco IOS XE Release 3.2S | Public Key Infrastructure (PKI) High Availability refers to the technology and system-design approach that supports the operation of a certificate-based public key cryptographic system that is available at all times. This feature was integrated into Cisco IOS XE Release 3.2S. The following sections provide information about this feature:  
  - PKI High Availability Support |
Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Configuring Certificate Enrollment for a PKI

Certificate enrollment, which is the process of obtaining a certificate from a certification authority (CA), occurs between the end host that requests the certificate and the CA. Each peer that participates in the public key infrastructure (PKI) must enroll with a CA. This module describes the different methods available for certificate enrollment and how to set up each method for a participating PKI peer.

- Finding Feature Information, page 79
- Prerequisites for PKI Certificate Enrollment, page 79
- Restrictions for PKI Certificate Enrollment, page 79
- Information About Certificate Enrollment for a PKI, page 80
- How to Configure Certificate Enrollment for a PKI, page 83
- Configuration Examples for PKI Certificate Enrollment Requests, page 103
- Additional References, page 109
- Feature Information for PKI Certificate Enrollment, page 111

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for PKI Certificate Enrollment

Before you configure peers for certificate enrollment, you must:

- Authenticate the CA.
- Have a generated Rivest, Shamir, and Adelman (RSA) key pair to enroll and a PKI in which to enroll.

Restrictions for PKI Certificate Enrollment
Cisco IOS certificate servers cannot be configured using Cisco IOS XE software. The Cisco IOS certificate servers must be set up using Cisco IOS software (T- or mainline-based) images.

**Information About Certificate Enrollment for a PKI**

- What Are CAs, page 80
- Authentication of the CA, page 80
- Supported Certificate Enrollment Methods, page 81
- Registration Authorities (RA), page 82
- Automatic Certificate Enrollment, page 82
- Certificate Enrollment Profiles, page 83

**What Are CAs**

A CA manages certificate requests and issues certificates to participating network devices. These services (managing certificate requests and issuing certificates) provide centralized key management for the participating devices to validate identities and to create digital certificates. Before any PKI operations can begin, the CA generates its own public key pair and creates a self-signed CA certificate; thereafter, the CA can sign certificate requests and begin peer enrollment for the PKI.

You can use the Cisco IOS XE certificate server or a CA provided by a third-party CA vendor.

---

**Note**

Cisco IOS XE certificate servers cannot be configured using Cisco IOS XE software. The Cisco IOS certificate servers must be set up using Cisco IOS software (T- or mainline-based images).

---

**Authentication of the CA**

The certificate of the CA must be authenticated before the device will be issued its own certificate and before certificate enrollment can occur. Authentication of the CA typically occurs only when you initially configure PKI support at your router. To authenticate the CA, issue the `crypto pki authenticate` command, which authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA.

**Authentication via the fingerprint Command**

You can issue the `fingerprint` command to preenter a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.

If a fingerprint is not preentered for a trustpoint, and if the authentication request is interactive, you must verify the fingerprint that is displayed during authentication of the CA certificate. If the authentication request is noninteractive, the certificate will be rejected without a preentered fingerprint.

---

**Note**

If the authentication request is made using the command-line interface (CLI), the request is an interactive request. If the authentication request is made using HTTP or another management tool, the request is a noninteractive request.
Supported Certificate Enrollment Methods

Cisco IOS XE software supports the following methods to obtain a certificate from a CA:

- Simple Certificate Enrollment Protocol (SCEP)--A Cisco developed enrollment protocol that uses HTTP to communicate with the CA or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

- PKCS12--The router imports certificates in PKCS12 format from an external server.

- IOS File System (IFS)--The router uses any file system that is supported by Cisco IOS XE software (such as TFTP, FTP, flash, and NVRAM) to send a certificate request and to receive the issued certificate. Users may enable IFS certificate enrollment when their CA does not support SCEP.

- Manual cut-and-paste--The router displays the certificate request on the console terminal, allowing the user to enter the issued certificate on the console terminal. A user may manually cut-and-paste certificate requests and certificates when there is no network connection between the router and CA.

- Enrollment profiles--The router sends HTTP-based enrollment requests directly to the CA server instead of to the RA-mode CS. Enrollment profiles can be used if a CA server does not support SCEP.

- Self-signed certificate enrollment for a trustpoint--The secure HTTP (HTTPS) server generates a self-signed certificate that is to be used during the secure socket layer (SSL) handshake, establishing a secure connection between the HTTPS server and the client. The self-signed certificate is then saved in the router’s startup configuration (NVRAM). The saved, self-signed certificate can then be used for future SSL handshakes, eliminating the user intervention that was necessary to accept the certificate every time the router reloaded.

Note

To take advantage of automated certificate and key rollover functionality, you must be running a CA that supports rollover and SCEP must be used as your client enrollment method. If you are running a Cisco IOS XE CA, you must be running Cisco IOS XE Release 2.1 or a later release for rollover support.

Note

To take advantage of autoenrollment and auto reenrollment, do not use either TFTP or manual cut-and-paste enrollment as your enrollment method. Both TFTP and manual cut-and-paste enrollment methods are manual enrollment processes, requiring user input.

Cisco IOS Suite-B Support for Certificate Enrollment for a PKI

Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.

Suite-B adds the following support for the certificate enrollment for a PKI:

- Elliptic Curve Digital Signature Algorithm (ECDSA) (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.
- PKI support for validation of for X.509 certificates using ECDSA signatures.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS.
See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.

Registration Authorities (RA)

A Cisco IOS XE certificate server can be configured to run in RA mode. An RA offloads authentication and authorization responsibilities from a CA. When the RA receives a SCEP or manual enrollment request, the administrator can either reject or grant it on the basis of local policy. If the request is granted, it will be forwarded to the issuing CA, and the CA can be configured to automatically generate the certificate and return it to the RA. The client can later retrieve the granted certificate from the RA.

Automatic Certificate Enrollment

Certificate autoenrollment allows the CA client to automatically request a certificate from its CA server. This automatic router request eliminates the need for operator intervention when the enrollment request is sent to the CA server. Automatic enrollment is performed on startup for any trustpoint CA that is configured and that does not have a valid client certificate. When the certificate expires, a new certificate is automatically requested.

---

**Note**

When automatic enrollment is configured, clients automatically request client certificates. The CA server performs its own authorization checks; if these checks include a policy to automatically issue certificates, all clients will automatically receive certificates, which is not very secure. Thus, automatic certificate enrollment should be combined with additional authentication and authorization mechanisms (such as Secure Device Provisioning (SDP), leveraging existing certificates, and one-time passwords).

---

Automated Client Certificate and Key Rollover

By default, the automatic certificate enrollment function requests a new client certificate and keys from the CS before the client’s current certificate expires. Certificate and key rollover allows the certificate renewal rollover request to be made before the certificate expires by retaining the current key and certificate until the new, or rollover, certificate is available. After a specified amount of time, the rollover certificate and keys will become the active certificate and keys. The expired certificate and keys are immediately deleted upon rollover and removed from the certificate chain and CRL.

The setup for automatic rollover is twofold: CA clients must be automatically enrolled and the client’s CAs must be automatically enrolled and have the `auto-rollover` command enabled.

An optional renewal percentage parameter can be used with the `auto-enroll` command to allow a new certificate to be requested when a specified percentage of the lifetime of the certificate has passed. For example, if the renewal percentage is configured as 90 and the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. In order for automatic rollover to occur, the renewal percentage must be less than 100. The specified percent value must not be less than 10. If a client certificate is issued for less than the configured validity period due to the impending expiration of the CA certificate, the rollover certificate will be issued for the balance of that period. A minimum of 10 percent of the configured validity period, with an absolute minimum of 3 minutes, is required to allow rollover enough time to function.
Tip

If CA autoenrollment is not enabled, you may manually initiate rollover on an existing client with the `crypto pki enroll` command if the expiration time of the current client certificate is equal to or greater than the expiration time of the corresponding CA certificate. The client will initiate the rollover process, which only occurs if the server is configured for automated rollover and has an available rollover server certificate.

Note

A key pair is also sent if configured by the `auto-enroll re-generate` command and keyword. It is recommended that a new key pair be issued for security reasons.

Certificate Enrollment Profiles

Enrollment profiles allow users to specify certificate authentication, enrollment, and reenrollment parameters when prompted. The values for these parameters are referenced by two templates that make up the profile. One template contains parameters for the HTTP request that is sent to the CA server to obtain the certificate of the CA (also known as certificate authentication); the other template contains parameters for the HTTP request that is sent to the CA for certificate enrollment.

Configuring two templates enables users to specify different URLs or methods for certificate authentication and enrollment; for example, authentication (getting the certificate of the CA) can be performed via TFTP (using the `authentication url` command) while enrollment can be performed manually (using the `enrollment terminal` command).

Users may specify the PKCS7 format for certificate renewal requests.

Note

A single enrollment profile can have up to three separate sections for each task—certificate authentication, enrollment, and reenrollment.

How to Configure Certificate Enrollment for a PKI

This section contains the following enrollment option procedures. If you configure enrollment or autoenrollment (the first task), you cannot configure manual certificate enrollment. Also, if you configure TFTP or manual cut-and-paste certificate enrollment, you cannot configure autoenrollment, auto reenrollment, an enrollment profile, nor can you utilize the automated CA certificate rollover capability.

- Configuring Certificate Enrollment or Autoenrollment, page 83
- Configuring Manual Certificate Enrollment, page 89
- Configuring a Persistent Self-Signed Certificate for Enrollment via SSL, page 94
- Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, page 99

Configuring Certificate Enrollment or Autoenrollment

Perform this task to configure certificate enrollment for clients participating in your PKI.

Before configuring automatic certificate enrollment requests, you should ensure that all necessary enrollment information is configured.
Prerequisites for Enabling Automated Client Certificate and Key Rollover

CA client support for certificate rollover is automatically enabled when using autoenrollment. For automatic CA certificate rollover to run successfully, the following prerequisites are applicable:

- Your network devices must support shadow PKI.
- Your clients must be running Cisco IOS XE Release 2.1 or a later release.
- The client’s CS must support automatic rollover. See the section “Automatic CA Certificate and Key Rollover” in the chapter Configuring and Managing a Cisco IOS XE Certificate Server for PKI Deployment for more information on CA server automatic rollover configuration.

Prerequisites for Specifying Autoenrollment Initial Key Generation Location

To specify the location of the autoenrollment initial key generation, you must be running Cisco IOS XE Release 2.1 or a later release.

RSA Key Pair Restriction for Autoenrollment

Trustpoints configured to generate a new key pair using the `regenerate` command or the `regenerate` keyword of the `auto-enroll` command must not share key pairs with other trustpoints. To give each trustpoint its own key pair, use the `rsakeypair` command in ca-trustpoint configuration mode. Sharing key pairs among regenerating trustpoints is not supported and will cause loss of service on some of the trustpoints because of key and certificate mismatches.

Restrictions for Automated Client Certificate and Key Rollover

In order for clients to run automatic CA certificate rollover successfully, the following restrictions are applicable:

- SCEP must be used to support rollover. Any device that enrolls with the PKI using an alternative to SCEP as the certificate management protocol or mechanism (such as enrollment profiles, manual enrollment, or TFTP enrollment) will not be able to take advantage of the rollover functionality provided by SCEP.
- If the configuration cannot be saved to the startup configuration after a shadow certificate is generated, rollover will not occur.
**SUMMARY STEPS**

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment [mode] [retry period minutes] [retry count number] url url [pem]
5. eckeypair label
6. subject-name [x.500-name]
7. ip address {ip address | interface | none
8. serial-number [none]
9. auto-enroll [percent] [regenerate
10. usage method1 [method2 [method3]]
11. password string
12. rsakeypair key-label key-size encryption-key-size ]
13. fingerprint ca-fingerprint
14. on devicename :
15. exit
16. crypto pki authenticate name
17. exit
18. copy system:running-config nvram:startup-config
19. show crypto pki certificates

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>· Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki trustpoint name</td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto pki trustpoint mytp</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

**Step 4**

**enrollment [mode] [retry period minutes] [retry count number] url url [pem]**

**Purpose**

Specifies the URL of the CA on which your router should send certificate requests.

- **mode** -- Specifies RA mode if your CA system provides an RA.
- **retry period minutes** -- Specifies the wait period between certificate request retries. The default is 1 minute between retries.
- **retry count number** -- Specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. (Specify from 1 to 100 retries.)
- **url** -- URL of the file system where your router should send certificate requests. For enrollment method options, see the `enrollment` command in the *Cisco IOS Security Command Reference*.
- **pem** -- Adds privacy-enhanced mail (PEM) boundaries to the certificate request.

**Note**

An enrollment method other than TFTP or manual cut-and-paste must be configured to support autoenrollment.

**Example:**

```
Router(ca-trustpoint)# enrollment url http://cat.example.com
```

### Command or Action

**Step 5**

**eckeypair label**

**Purpose**

(Optional) Configures the trustpoint to use an Elliptic Curve (EC) key on which certificate requests are generated using ECDSA signatures. The `label` argument specifies the EC key label that is configured using the `crypto key generate rsa` or `crypto key generate ec keysize` command in global configuration mode. See the Configuring Internet Key Exchange for IPsec VPNs feature module for more information.

**Note**

If an ECDSA signed certificate is imported without a trustpoint configuration, then the label defaults to the FQDN value.

**Example:**

```
Router(ca-trustpoint)# eckeypair Router_1_Key
```

### Command or Action

**Step 6**

**subject-name [x.500-name]**

**Purpose**

(Optional) Specifies the requested subject name that will be used in the certificate request.

- **x.500-name** -- If it is not specified, the fully qualified domain name (FQDN), which is the default subject name, will be used.

**Example:**

```
Router(ca-trustpoint)# subject-name cat
```

### Command or Action

**Step 7**

**ip address [ip address | interface | none**

**Purpose**

(Optional) Includes the IP address of the specified interface in the certificate request.

Issue the `none` keyword if no IP address should be included.

**Note**

If this command is enabled, you will not be prompted for an IP address during enrollment for this trustpoint.

**Example:**

```
Router(ca-trustpoint)# ip address 192.168.1.66
```

### Command or Action

**Step 8**

**serial-number [none]**

**Purpose**

(Optional) Specifies the router serial number in the certificate request, unless the `none` keyword is issued.

**Example:**

```
Router(ca-trustpoint)# serial-number
```
### Configuring Certificate Enrollment for a PKI

**How to Configure Certificate Enrollment for a PKI**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 9</strong> auto-enroll [percent] [regenerate]</td>
<td>(Optional) Enables autoenrollment, allowing the client to automatically request a rollover certificate from the CA. If autoenrollment is not enabled, the client must be manually reenrolled in your PKI upon certificate expiration.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# auto-enroll regenerate</td>
<td></td>
</tr>
<tr>
<td>· By default, only the Domain Name System (DNS) name of the router is included in the certificate.</td>
<td></td>
</tr>
<tr>
<td>· Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached.</td>
<td></td>
</tr>
<tr>
<td>· Use the regenerate keyword to generate a new key for the certificate even if a named key already exists.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> If the key pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: &quot;! RSA key pair associated with trustpoint is exportable.&quot;</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> It is recommended that a new key pair be generated for security reasons.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 10</strong> usage method1 [method2 [method3]]</td>
<td>(Optional) Specifies the intended use for the certificate. Available options are ike, ssl-client, and ssl-server; the default is ike.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# usage ssl-client</td>
<td></td>
</tr>
<tr>
<td><strong>Step 11</strong> password string</td>
<td>(Optional) Specifies the revocation password for the certificate. If this command is enabled, you will not be prompted for a password during enrollment for this trustpoint.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# password string1</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> When SCEP is used, this password can be used to authorize the certificate request--often via a one-time password or similar mechanism.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 12</strong> rsakeypair key-label key-size encryption-key-size ]</td>
<td>(Optional) Specifies which key pair to associate with the certificate.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# rsakeypair cat</td>
<td></td>
</tr>
<tr>
<td>· A key pair with key-label will be generated during enrollment if it does not already exist or if the auto-enroll regenerate command was issued.</td>
<td></td>
</tr>
<tr>
<td>· Specify the key-size argument for generating the key, and specify the encryption-key-size argument to request separate encryption, signature keys, and certificates.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> If this command is not enabled, the FQDN key pair is used.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> fingerprint ca-fingerprint</td>
<td>(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> If the fingerprint is not provided and authentication of the CA certificate is interactive, the fingerprint will be displayed for verification.</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action | Purpose
---|---
**Step 14** on device-name: | (Optional) Specifies that RSA keys will be created on the specified device upon autoenrollment initial key generation. Devices that may be specified include NVRAM and local disks. USB tokens may be used as a storage device.

**Example:**
```
Router(ca-trustpoint)# on usbtoken0:
```

**Step 15** exit | Exits ca-trustpoint configuration mode and returns to global configuration mode.

**Example:**
```
Router(ca-trustpoint)# exit
```

**Step 16** crypto pki authenticate name | Retrieves the CA certificate and authenticates it.
- Check the certificate fingerprint if prompted.

**Example:**
```
Router(config)# crypto pki authenticate mytp
```

**Note** This command is optional if the CA certificate is already loaded into the configuration.

**Step 17** exit | Exits global configuration mode.

**Example:**
```
Router(config)# exit
```

**Step 18** copy system:running-config nvram:startup-config | (Optional) Copies the running configuration to the NVRAM startup configuration.

**Note** Autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

**Example:**
```
Router# copy system:running-config nvram:startup-config
```

**Step 19** show crypto pki certificates | (Optional) Displays information about your certificates, including any rollover certificates.

**Example:**
```
Router# show crypto pki certificates
```

### Examples
The following example shows the configuration for the “mytp-A” certificate server and its associated trustpoint, where RSA keys generated by the initial autoenrollment for the trustpoint will be stored on a USB token, “usbtoken0”:
```
crypto pki server mytp-A
```
Configuring Manual Certificate Enrollment

Manual certificate enrollment can be set up via TFTP or the manual cut-and-paste method. Both options can be used if your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform one of the following tasks to set up manual certificate enrollment:

- **PEM-Formatted Files for Certificate Enrollment Request**, page 89
- **Configuring Cut-and-Paste Certificate Enrollment**, page 89
- **Configuring TFTP Certificate Enrollment**, page 92

PEM-Formatted Files for Certificate Enrollment Request

Using PEM-formatted files for certificate requests can be helpful for customers who are using terminal or profile-based enrollment to request certificates from their CA server. Customers using PEM-formatted files can directly use existing certificates on their routers.

Restrictions for Manual Certificate Enrollment

**Switching Enrollment URLs When Using SCEP**

We do not recommend switching URLs if SCEP is used; that is, if the enrollment URL is “http://myca,” do not change the enrollment URL after getting the CA certificate and before enrolling the certificate. A user can switch between TFTP and manual cut-and-paste.

**Key Regeneration Restriction**

Do not regenerate the keys manually using the `crypto key generate` command; key regeneration will occur when the `crypto pki enroll` command is issued if the `regenerate` keyword is specified.

Configuring Cut-and-Paste Certificate Enrollment

Perform this task to configure manual certificate enrollment via the cut-and-paste method for peers participating in your PKI.
### SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment terminal pem
5. fingerprint ca-fingerprint
6. exit
7. crypto pki authenticate name
8. crypto pki enroll name
9. crypto pki import name certificate
10. exit
11. show crypto pki certificates

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router# configure terminal</td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki trustpoint name</td>
<td>Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(config)# crypto pki trustpoint mytp</td>
</tr>
<tr>
<td><strong>Step 4</strong> enrollment terminal pem</td>
<td>Specifies manual cut-and-paste certificate enrollment method. The certificate request will be displayed on the console terminal so that you may manually copied (or cut).</td>
</tr>
<tr>
<td>Example:</td>
<td>Router(ca-trustpoint)# enrollment terminal</td>
</tr>
<tr>
<td></td>
<td>• pem --Configures the trustpoint to generate PEM-formatted certificate requests to the console terminal.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 5</th>
<th>fingerprint ca-fingerprint</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(ca-trustpoint)# fingerprint 12EFS3FA 355CD23E 12EFS3FA 355CD23E</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If the fingerprint is not provided, it will be displayed for verification.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# exit</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Exits ca-trustpoint configuration mode and returns to global configuration mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>crypto pki authenticate name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki authenticate mytp</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Retrieves the CA certificate and authenticates it.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>crypto pki enroll name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki enroll mytp</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Generates certificate request and displays the request for copying and pasting into the certificate server.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are also given the choice about displaying the certificate request to the console terminal. The base-64 encoded certificate with or without PEM headers as requested is displayed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>crypto pki import name certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki import mytp certificate</td>
</tr>
<tr>
<td><strong>Purpose:</strong></td>
<td>Imports a certificate manually at the console terminal (pasting). The base-64 encoded certificate is accepted from the console terminal and inserted into the internal certificate database.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>You must enter this command twice if usage keys, a signature key and an encryption key, are used. The first time the command is entered, one of the certificates is pasted into the router. The second time the command is entered, the other certificate is pasted into the router. It does not matter which certificate is pasted first.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If this applies to the certificate authority you are using, import the general purpose certificate. The router will not use one of the two key pairs generated.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
Router(config)# exit
```

<table>
<thead>
<tr>
<th><strong>Step 11</strong> show crypto pki certificates</th>
<th>(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.</th>
</tr>
</thead>
</table>

**Example:**

```
Router# show crypto pki certificates
```

### Configuring TFTP Certificate Enrollment

Perform this task to configure manual certificate enrollment using a TFTP server.

- You must know the correct URL to use if you are configuring certificate enrollment via TFTP.
- The router must be able to write a file to the TFTP server for the `crypto pki enroll` command.
- If using a file specification with the `enrollment` command, the file must contain the CA certificate either in binary format or be base-64 encoded.
- You must know if your CA ignores key usage information in a certificate request and issues only a general purpose usage certificate.

**Caution**

Some TFTP servers require that the file must exist on the server before it can be written. Most TFTP servers require that the file be “write-able” by the world. This requirement may pose a risk because any router or other device may write or overwrite the certificate request; thus, the replacement certificate request will not be used by the CA administrator, who must first check the enrollment request fingerprint before granting the certificate request.

### SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment [mode] [retry period minutes] [retry count number] url url [pem]
5. fingerprint ca-fingerprint
6. exit
7. crypto pki authenticate name
8. crypto pki enroll name
9. crypto pki import name certificate
10. exit
11. show crypto pki certificates
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
- Enter your password if prompted. |
<p>| <strong>Example:</strong> Router&gt; enable |
| <strong>Step 2</strong> configure terminal | Enters global configuration mode. |
| <strong>Example:</strong> Router# configure terminal |
| <strong>Step 3</strong> crypto pki trustpoint name | Declares the trustpoint and a given name and enters ca-trustpoint configuration mode. |
| <strong>Example:</strong> Router(config)# crypto pki trustpoint mytp |
| <strong>Step 4</strong> enrollment [mode] [retry period minutes] [retry count number] url url [pem] | Specifies TFTP as the enrollment method to send the enrollment request and to retrieve the CA certificate and router certificate and any optional parameters. |
| <strong>Note</strong> For TFTP enrollment, the url must be configured as a TFTP url, tftp://example_tftp_url. |
| <strong>Example:</strong> Router(ca-trustpoint)# enrollment url tftp://certserver/file_specification |
| <strong>Step 5</strong> fingerprint ca-fingerprint | (Optional) Specifies the fingerprint of the CA certificate received via an out-of-band method from the CA administrator. |
| <strong>Note</strong> If the fingerprint is not provided, it will be displayed for verification. |
| <strong>Example:</strong> Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E |
| <strong>Step 6</strong> exit | Exits ca-trustpoint configuration mode and returns to global configuration mode. |
| <strong>Example:</strong> Router(config)# exit |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong> crypto pki authenticate name</td>
<td>Retrieves the CA certificate and authenticates it from the specified TFTP server.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki authenticate mytp</td>
</tr>
<tr>
<td><strong>Step 8</strong> crypto pki enroll name</td>
<td>Generates certificate request and writes the request out to the TFTP server. You are prompted for enrollment information, such as whether to include the router FQDN and IP address in the certificate request. You are queried about whether or not to display the certificate request to the console terminal. The filename to be written is appended with the extension “.req”. For usage keys, a signature key and an encryption key, two requests are generated and sent. The usage key request filenames are appended with the extensions “-sign.req” and “-encr.req” respectively.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki enroll mytp</td>
</tr>
<tr>
<td><strong>Step 9</strong> crypto pki import name certificate</td>
<td>Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate. The router will attempt to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from “.req” to “.crt”. For usage key certificates, the extensions “-sign.crt” and “-encr.crt” are used. The router will parse the received files, verify the certificates, and insert the certificates into the internal certificate database on the router. <strong>Note</strong> Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The router will not use one of the two keypairs generated.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# crypto pki import mytp certificate</td>
</tr>
<tr>
<td><strong>Step 10</strong> exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router(config)# exit</td>
</tr>
<tr>
<td><strong>Step 11</strong> show crypto pki certificates</td>
<td>(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Router# show crypto pki certificates</td>
</tr>
</tbody>
</table>

**Configuring a Persistent Self-Signed Certificate for Enrollment via SSL**

This section contains the following tasks:
These tasks are optional because if you enable the HTTPS server, it generates a self-signed certificate automatically using default values.

Persistent Self-Signed Certificates Overview

The SSL protocol can be used to establish a secure connection between an HTTPS server and a client (web browser). During the SSL handshake, the client expects the SSL server’s certificate to be verifiable using a certificate the client already possesses.

If Cisco IOS XE software does not have a certificate that the HTTPS server can use, the server generates a self-signed certificate by calling a PKI application programming interface (API). When the client receives this self-signed certificate and is unable to verify it, intervention is needed. The client asks you if the certificate should be accepted and saved for future use. If you accept the certificate, the SSL handshake continues.

Future SSL handshakes between the same client and the server use the same certificate. However, if the router is reloaded, the self-signed certificate is lost. The HTTPS server must then create a new self-signed certificate. This new self-signed certificate does not match the previous certificate so you are once again asked to accept it.

Requesting acceptance of the router’s certificate each time that the router reloads may present an opportunity for an attacker to substitute an unauthorized certificate when you are being asked to accept the certificate. Persistent self-signed certificates overcome all these limitations by saving a certificate in the router’s startup configuration.

Restrictions

You can configure only one trustpoint for a persistent self-signed certificate.

Note

Do not change the IP domain name or the hostname of the router after creating the self-signed certificate. Changing either name triggers the regeneration of the self-signed certificate and overrides the configured trustpoint. WebVPN ties the SSL trustpoint name to the WebVPN gateway configuration. If a new self-signed certificate is triggered, then the new trustpoint name does not match the WebVPN configuration, causing the WebVPN connections to fail.

Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters

Perform the following task to configure a trustpoint and specify self-signed certificate parameters.
SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki trustpoint name
4. enrollment selfsigned
5. subject-name [x.500-name]
6. rsakeypair key-label [key-size [encryption-key-size]]
7. crypto pki enroll name
8. end
9. show crypto pki certificates [trustpoint-name [ verbose]]
10.show crypto pki trustpoints [status | label [status]]

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Enter your password if prompted.</td>
</tr>
<tr>
<td>Router&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki trustpoint name</td>
<td>Declares the CA that your router should use and enters ca-trustpoint configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(config)# crypto pki trustpoint local</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> The crypto pki trustpoint command replaced the crypto pki trustpoint command.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> enrollment selfsigned</td>
<td>Specifies self-signed enrollment.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# enrollment selfsigned</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> subject-name [x.500-name]</td>
<td>(Optional) Specifies the requested subject name to be used in the certificate request.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Router(ca-trustpoint)# subject-name</td>
<td></td>
</tr>
<tr>
<td>• If the x-500-name argument is not specified, the FQDN, which is the default subject name, is used.</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step 6</th>
<th>rsakeypair key-label [key-size [encryption-key-size]]</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) Specifies which key pair to associate with the certificate.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The key-label argument will be generated during enrollment if it does not already exist or if the auto-enroll regenerate command was issued.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Specify the key-size argument for generating the key, and specify the encryption-key-size argument to request separate encryption, signature keys, and certificates.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note If this command is not enabled, the FQDN key pair is used.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(ca-trustpoint)# rsakeypair examplekeys 1024 1024</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 7</th>
<th>crypto pki enroll name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tells the router to generate the persistent self-signed certificate.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(ca-trustpoint)# crypto pki enroll local</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 8</th>
<th>end</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) Exits ca-trustpoint configuration mode and global configuration mode.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(ca-trustpoint)# end</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router(config)# end</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 9</th>
<th>show crypto pki certificates [trustpoint-name [verbose]]</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Displays information about your certificate, the certification authority certificate, and any registration authority certificates.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Router# show crypto pki certificates local verbose</td>
<td></td>
</tr>
</tbody>
</table>

| Step 10 | show crypto pki trustpoints [status | label | [status]] | Purpose |
|---------|-------------------------------------|---------|
|         | Displays the trustpoints that are configured in the router. |         |
|         | Example:                            |         |
|         | Router# show crypto pki trustpoints status |         |

### Enabling the HTTPS Server

Perform the following task to enable the HTTPS server.
To specify parameters, you must create a trustpoint and configure it. To use default values, delete any existing self-signed trustpoints. Deleting all self-signed trustpoints causes the HTTPS server to generate a persistent self-signed certificate using default values as soon as the server is enabled.

**SUMMARY STEPS**

1. `enable`
2. `configure terminal`
3. `ip http secure-server`
4. `end`
5. `copy system:running-config nvram: startup-config`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| Example:          |         |
| Router> enable    |         |
| **Step 2** configure terminal | Enters global configuration mode. |
| Example:          |         |
| Router# configure terminal |         |
| **Step 3** ip http secure-server | Enables the secure HTTP web server.  
  **Note** A key pair (modulus 1024) and a certificate are generated. |
| Example:          |         |
| Router(config)# ip http secure-server |         |
| **Step 4** end | Exits global configuration mode. |
| Example:          |         |
| Router(config)# end |         |
| **Step 5** copy system:running-config nvram: startup-config | Saves the self-signed certificate and the HTTPS server in enabled mode. |
| Example:          |         |
| Router# copy system:running-config nvram: startup-config |         |
Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment

Perform this task to configure an enrollment profile for certificate enrollment or reenrollment of a router with a Cisco IOS XE CA that is already enrolled with a third-party vendor CA.

Enable a router that is enrolled with a third-party vendor CA to use its existing certificate to enroll with the Cisco IOS XE certificate server so the enrollment request is automatically granted. To enable this functionality, you must issue the `enrollment credential` command. Also, you cannot configure manual certificate enrollment.

Before configuring a certificate enrollment profile for the client router that is already enrolled with a third-party vendor CA so that the router can reenroll with a Cisco IOS XE certificate server, you should have already performed the following tasks at the client router:

- Defined a trustpoint that points to the third-party vendor CA.
- Authenticated and enrolled the client router with the third-party vendor CA.

### Note

- To use certificate profiles, your network must have an HTTP interface to the CA.
- If an enrollment profile is specified, an enrollment URL may not be specified in the trustpoint configuration. Although both commands are supported, only one command can be used at a time in a trustpoint.
- Because there is no standard for the HTTP commands used by various CAs, the user is required to enter the command that is appropriate to the CA that is being used.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto pki trustpoint name`
4. `enrollment profile label`
5. `exit`
6. `crypto pki profile enrollment label`
7. Do one of the following:
   - `authentication url url`
8. `authentication command`
9. Do one of the following:
   - `enrollment url url`
10. `enrollment credential label`
11. `enrollment command`
12. `parameter number {value value | prompt string}`
13. `exit`
14. `show crypto pki certificates`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  

- Enter your password if prompted.  

  **Example:**  
  Router> enable |
| **Step 2** configure terminal | Enters global configuration mode.  

  **Example:**  
  Router# configure terminal |
| **Step 3** crypto pki trustpoint *name* | Declares the trustpoint and a given name and enter ca-trustpoint configuration mode.  

  **Example:**  
  Router(config)# crypto pki trustpoint Entrust |
| **Step 4** enrollment profile *label* | Specifies that an enrollment profile is to be used for certificate authentication and enrollment.  

  **Example:**  
  Router(ca-trustpoint)# enrollment profile E |
| **Step 5** exit | Exits ca-trustpoint configuration mode.  

  **Example:**  
  Router(ca-trustpoint)# exit |
| **Step 6** crypto pki profile enrollment *label* | Defines an enrollment profile and enters ca-profile-enroll configuration mode.  

  **Example:**  
  Router(config)# crypto pki profile enrollment E  

  - *label* --Name for the enrollment profile; the enrollment profile name must match the name specified in the enrollment profile command.  

---

Enabling the HTTPS Server

Public Key Infrastructure Configuration Guide, Cisco IOS XE Release 2
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 7</strong></td>
<td>Do one of the following:</td>
</tr>
<tr>
<td></td>
<td>·  <strong>authentication url</strong> <em>url</em></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Specifies the URL of the CA server to which to send certificate authentication requests.</td>
</tr>
<tr>
<td></td>
<td>·  <em>url</em> --URL of the CA server to which your router should send authentication requests. If using HTTP, the URL should read “http://CA_name,” where CA_name is the host DNS name or IP address of the CA. If using TFTP, the URL should read “tftp://certserver/file_specification.” (If the URL does not include a file specification, the FQDN of the router will be used.)</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Specifies manual cut-and-paste certificate authentication.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>authentication terminal</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td><strong>authentication command</strong></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>(Optional) Specifies the HTTP command that is sent to the CA for authentication. This command should be used after the <strong>authentication url</strong> command has been entered.</td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Step 9</strong> Do one of the following:</td>
<td>Specifies the URL of the CA server to which to send certificate enrollment requests via HTTP or TFTP. Specifies manual cut-and-paste certificate enrollment.</td>
</tr>
<tr>
<td>• enrollment <strong>url</strong></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-profile-enroll)# enrollment url <a href="http://entrust:81/cda-cgi/clientcgi.exe">http://entrust:81/cda-cgi/clientcgi.exe</a></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Step 10 enrollment <strong>credential</strong> <em>label</em></td>
<td>(Optional) Specifies the third-party vendor CA trustpoint that is to be enrolled with the Cisco IOS XE CA.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-profile-enroll)# enrollment credential Entrust</td>
<td>Note This command cannot be issued if manual certificate enrollment is being used.</td>
</tr>
<tr>
<td>Step 11 enrollment <strong>command</strong></td>
<td>(Optional) Specifies the HTTP command that is sent to the CA for enrollment.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(ca-profile-enroll)# enrollment command</td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **Step 12**  
parameter number {value value | prompt string} | (Optional) Specifies parameters for an enrollment profile. This command can be used multiple times to specify multiple values. |
| Example:  
Router(ca-profile-enroll)# parameter 1 value aaaa-bbbb-cccc | |
| **Step 13**  
exit | Enter this command two times—one time to exit ca-profile-enroll configuration mode and the second time to exit global configuration mode. |
| Example:  
Router(ca-profile-enroll)# exit | |
| Example:  
Router(config)# exit | |
| **Step 14**  
show crypto pki certificates | (Optional) Displays information about your certificates, the certificates of the CA, and RA certificates. |
| Example:  
Router# show crypto pki certificates | |

- What to Do Next, page 103

**What to Do Next**

If you configured the router to reenroll with a Cisco IOS XE CA, you should configure the Cisco IOS XE certificate server to accept enrollment requests only from clients already enrolled with the specified third-party vendor CA trustpoint to take advantage of this functionality.

**Configuration Examples for PKI Certificate Enrollment Requests**

**Configuring Autoenrollment Example**

The following example shows how to configure the router to automatically enroll with a CA on startup, enabling automatic rollover, and how to specify all necessary enrollment information in the configuration:

crypto pki trustpoint trustpt1  
enrollment url http://trustpt1.company.com//  
subject-name OU=Spiral Dept., O=exampleco.com  
ip-address Fastethernet-0  
serial-number none  
usage ike
In this example, keys are neither regenerated nor rolled over.

### Configuring Certificate Autoenrollment with Key Regeneration Example

The following example shows how to configure the router to automatically enroll with the CA named “trustme1” on startup and enable automatic rollover. The `regenerate` keyword is issued, so a new key will be generated for the certificate and reissued when the automatic rollover process is initiated. The renewal percentage is configured as 90 so if the certificate has a lifetime of one year, a new certificate is requested 36.5 days before the old certificate expires. The changes made to the running configuration are saved to the NVRAM startup configuration because autoenrollment will not update NVRAM if the running configuration has been modified but not written to NVRAM.

```
crypto pki trustpoint trustme1
  enrollment url http://trustme1.company.com/
  subject-name OU=Spiral Dept., O=tiedye.com
  ip-address ethernet0
  serial-number none
  auto-enroll 90 regenerate
  password revokeme
  rsa-keypair trustme1 2048
  exit

crypto pki authenticate trustme1

copy system:running-config nvram:startup-config
```

### Configuring Cut-and-Paste Certificate Enrollment Example

The following example shows how to configure certificate enrollment using the manual cut-and-paste enrollment method:

```
Router(config)#
crypto pki trustpoint TP
Router(config)#
```
configuration terminal
crypt pki authenticate TP
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

Certificate has the following attributes:
Fingerprint: D6C12961 CD78808A 4E02193C 0790082A
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
% Certificate successfully imported

Redisplay certificate request? [yes/no]: y
Display Certificate Request to terminal? [yes/no]: y
Encryption key certificate request -
Certificate Request follows:
MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYDVQQGEwJVUzEiMByGB0w
THKry9CjEjBgYIBgUAMAwEBzRBIGGCQgMBAAGjgcEwgb4wCwYDVR0PBAQDAgKGBAQMBAAHvGA1sqGSI3
! Redisplay enrollment request? [yes/no]: n
Encryption key certificate request -
Certificate Request follows:
MIIDaBCCAvgAwIBAgI0NQ0jqj29TAQoGgYAQYJKoZIhvcNAQkFMgAwEAggCzIhBAgh5+RC
Redisplay enrollment request? [yes/no]: n
Router(config)#

crypto pki enroll TP
% Start certificate enrollment..
% The subject name in the certificate will be:
Router.company.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Display Certificate Request to terminal? [yes/no]: y
Encryption key certificate request -
Certificate Request follows:
MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYDVQQGEwJVUzEiMByGB0w
THKry9CjEjBgYIBgUAMAwEBzRBIGGCQgMBAAGjgcEwgb4wCwYDVR0PBAQDAgKGBAQMBAAHvGA1sqGSI3
! Redisplay enrollment request? [yes/no]: n
Encryption key certificate request -
Certificate Request follows:
MIIDajCCAxSgAwIBAgIKFN7C6QAAAAAMRzANBgkqhkiG9w0BAQUFADA5MQswCQYDVQQGEwJVUzEiMByGB0w
THKry9CjEjBgYIBgUAMAwEBzRBIGGCQgMBAAGjgcEwgb4wCwYDVR0PBAQDAgKGBAQMBAAHvGA1sqGSI3
! Redisplay enrollment request? [yes/no]: n
Router(config)#
Enter the base 64 encoded certificate. End with a blank line or the word "quit" on a line by itself

Certificate
Status: Available
Certificate Serial Number: 14DECE05000000000C48
Certificate Usage: Encryption
Issuer:
CN = TP-CA-root
O = Company
c = US
Subject:
Name: Router.company.com
OID.1.2.840.113549.1.9.2 = Router.company.com
CRL Distribution Point:
http://tpca-root/CertEnroll/crl-root.crl
Validity Date:
start date: 18:16:45 PDT Jun 7 2002
end date: 18:26:45 PDT Jun 7 2003
renew date: 16:00:00 PST Dec 31 1969
Associated Trustpoints: TP
Certificate
Status: Available
Certificate Serial Number: 14DECE05000000000C49
Certificate Usage: Encryption
Issuer:
CN = TP-CA-root
O = Company
c = US
Subject:
Name: Router.company.com
OID.1.2.840.113549.1.9.2 = Router.company.com
CRL Distribution Point:
http://tpca-root/CertEnroll/crl-root.crl
Validity Date:
start date: 18:16:45 PDT Jun 7 2002
end date: 18:26:45 PDT Jun 7 2003
renew date: 16:00:00 PST Dec 31 1969
Associated Trustpoints: TP
Certificate
Status: Available
Certificate Serial Number: 14DECE05000000000C48
Certificate Usage: Encryption
Issuer:
CN = TP-CA-root
O = Company
c = US
Subject:
Name: Router.company.com
OID.1.2.840.113549.1.9.2 = Router.company.com
CRL Distribution Point:
http://tpca-root/CertEnroll/crl-root.crl
Validity Date:
start date: 18:16:45 PDT Jun 7 2002
end date: 18:26:45 PDT Jun 7 2003
renew date: 16:00:00 PST Dec 31 1969
Associated Trustpoints: TP
Certificate
Status: Available
Certificate Serial Number: 14DECE05000000000C49
Certificate Usage: Encryption
Issuer:
CN = TP-CA-root
O = Company
c = US
Subject:
Name: Router.company.com
OID.1.2.840.113549.1.9.2 = Router.company.com
CRL Distribution Point:
http://tpca-root/CertEnroll/crl-root.crl
Validity Date:
start date: 18:16:45 PDT Jun 7 2002
end date: 18:26:45 PDT Jun 7 2003
renew date: 16:00:00 PST Dec 31 1969
Associated Trustpoints: TP
Certificate
Configuring Manual Certificate Enrollment with Key Regeneration Example

The following example shows how to regenerate new keys with a manual certificate enrollment from the CA named “trustme2”:

crypto pki trustpoint trustme2
  enrollment url http://trustme2.company.com/
  subject-name OU=Spiral Dept., O=exampleco.com
  ip-address Fastethernet0
  serial-number none
  regenerate
  password revokeme
  rsakeypair trustme2 2048s
  exit
  crypto pki authenticate trustme2
  crypto pki enroll trustme2

Creating and Verifying a Persistent Self-Signed Certificate Example

The following example shows how to declare and enroll a trustpoint named “local” and generate a self-signed certificate with an IP address:

crypto pki trustpoint local
  enrollment selfsigned
  end
  configure terminal
  Enter configuration commands, one per line. End with CNTL/Z.
  crypto pki enroll local
  Nov 29 20:51:13.067: %SSH-5-ENABLED: SSH 1.99 has been enabled
  Nov 29 20:51:13.267: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
  % Include the router serial number in the subject name? [yes/no]: yes
  Enter Interface name or IP Address[]: Fast
  ethernet 0
  Generate Self Signed Router Certificate? [yes/no]: yes
  Router Self Signed Certificate successfully created
A router can have only one self-signed certificate. If you attempt to enroll a trustpoint configured for a self-signed certificate and one already exists, you receive a notification and are asked if you want to replace it. If so, a new self-signed certificate is generated to replace the existing one.

Enabling the HTTPS Server: Example

The following example shows how to enable the HTTPS server and generate a default trustpoint because one was not previously configured:

```
configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
% Generating 1024 bit RSA keys ...[OK]
*Dec 21 19:14:15.421:%PKI-4-NOAUTOSAVE:Configuration was modified. Issue "write memory"
to save new certificate
Router(config)#
```

**Note** You need to save the configuration to NVRAM if you want to keep the self-signed certificate and have the HTTPS server enabled following router reloads.

The following message also appears:

```
*Dec 21 19:14:10.441:%SSH-5-ENABLED:SSH 1.99 has been enabled
```

**Note** Creation of the key pair used with the self-signed certificate causes the Secure Shell (SSH) server to start. This behavior cannot be suppressed. You may want to modify your access control lists (ACLs) to permit or deny SSH access to the router.

Verifying the Self-Signed Certificate Configuration: Example

The following example displays information about the self-signed certificate that you just created:

```
Router# show crypto pki certificates
Router Self-Signed Certificate
Status: Available
Certificate Serial Number: 01
Certificate Usage: General Purpose
Issuer:
  cn=IOS-Self-Signed-Certificate-3326000105
Subject:
  Name: IOS-Self-Signed-Certificate-3326000105
cn=IOS-Self-Signed-Certificate-3326000105
Validity Date:
  start date: 19:14:14 GMT Dec 21 2004
  end date: 00:00:00 GMT Jan 1 2020
Associated Trustpoints: TP-self-signed-3326000105
```

**Note** The number 3326000105 above is the router’s serial number and varies depending on the router’s actual serial number.
The following example displays information about the key pair corresponding to the self-signed certificate:

Router# show crypto key mypubkey rsa
% Key pair was generated at: 19:14:10 GMT Dec 21 2004
Key name: TP-self-signed-3326000105
Usage: General Purpose Key
Key is not exportable.
Key Data:

The second key pair with the name TP-self-signed-3326000105.server is the SSH key pair and is generated when any key pair is created on the router and SSH starts up.

The following example displays information about the trustpoint named "local":

Router# show crypto pki trustpoints
Trustpoint local:
Subject Name:
serialNumber=C63EBBE9+ipaddress=10.3.0.18+hostname=test.company.com
Serial Number: 01
Persistent self-signed certificate trust point

Configuring Direct HTTP Enrollment Example

The following example show how to configure an enrollment profile for direct HTTP enrollment with a CA server:

crypto pki trustpoint Entrust
enrollment profile E
serial
crypto pki profile enrollment E
authentication url http://entrust:81
authentication command GET /certs/cacert.der
encryption url http://entrust:81/cda-cgi/clientcgi.exe
enrollment command POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
parameter 1 value aaaa-bbbb-cccc
parameter 2 value 5001

Additional References

The following sections provide references related to certificate enrollment for a PKI.
### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of PKI, including RSA keys, certificate enrollment, and CAs</td>
<td>“Cisco IOS XE PKI Overview: Understanding and Planning a PKI” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>RSA key generation and deployment</td>
<td>“Deploying RSA Keys Within a PKI” module in the Cisco IOS Security Configuration Guide: Secure Connectivity</td>
</tr>
<tr>
<td>Security commands</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIB</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>--</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for PKI Certificate Enrollment

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate Autoenrollment | Cisco IOS XE Release 2.1 | This feature introduces certificate autoenrollment, which allows the router to automatically request a certificate from the CA that is using the parameters in the configuration. The following sections provide information about this feature:  
  - Automatic Certificate Enrollment, page 82  
  - Configuring Certificate Enrollment or Autoenrollment, page 83  
  The following commands were introduced by this feature: auto-enroll, rsakeypair, show crypto pki timers |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
</table>
| Certificate Enrollment Enhancements  | Cisco IOS XE Release 2.1 | This feature introduces five new `crypto pki trustpoint` subcommands that provide new options for certificate requests and allow users to specify fields in the configuration instead of having to go through prompts. The following section provides information about this feature:  
  - Configuring Certificate Enrollment or Autoenrollment, page 83  

  The following commands were introduced by this feature: `ip-address (ca-trustpoint), password (ca-trustpoint), serial-number, subject-name, usage` |
| Direct HTTP Enrollment with CA Servers | Cisco IOS XE Release 2.1 | This feature allows users to configure an enrollment profile if their CA server does not support SCEP and they do not want to use an RA-mode CS. The enrollment profile allows users to send HTTP requests directly to the CA server instead of to an RA-mode CS. The following sections provide information about this feature:  
  - Certificate Enrollment Profiles, page 83  
  - Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, page 99  

  The following commands were introduced by this feature: `authentication command, authentication terminal, authentication url, crypto pki profile enrollment, enrollment command, enrollment profile, enrollment terminal, enrollment url, parameter` |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import of RSA Key Pair and</td>
<td>Cisco IOS XE</td>
<td>This feature allows customers to issue certificate requests and receive issued certificates in PEM-formatted files.</td>
</tr>
<tr>
<td>Certificates in PEM Format</td>
<td>Release 2.1</td>
<td>The following section provides information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring Manual Certificate Enrollment, page 89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were modified by this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- enrollment, enrollment terminal</td>
</tr>
<tr>
<td>Key Rollover for Certificate</td>
<td>Cisco IOS XE</td>
<td>This feature allows the certificate renewal request to be made before the certificate expires and retains the old key and certificate until the new certificate is available.</td>
</tr>
<tr>
<td>Renewal</td>
<td>Release 2.1</td>
<td>The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Automatic Certificate Enrollment, page 82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring Certificate Enrollment or Autoenrollment, page 83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring Manual Certificate Enrollment, page 89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified by this feature: auto-enroll, regenerate</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Manual Certificate Enrollment (TFTP Cut-and-Paste)</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature allows users to generate a certificate request and accept CA certificates as well as the router’s certificates via a TFTP server or manual cut-and-paste operations. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Supported Certificate Enrollment Methods, page 81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring Manual Certificate Enrollment, page 89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified by this feature: crypto pki import, enrollment, enrollment terminal</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Persistent Self-Signed Certificates</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature allows the HTTPS server to generate and save a self-signed certificate in the router startup configuration. Thus, future SSL handshakes between the client and the HTTPS server can use the same self-signed certificate without user intervention. In Cisco IOS XE Release 2.1, this feature was implemented on the Cisco ASR series routers. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Supported Certificate Enrollment Methods, page 81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring a Persistent Self-Signed Certificate for Enrollment via SSL, page 94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The following commands were introduced or modified by this feature: enrollment selfsigned, show crypto pki certificates, show crypto pki trustpoints</td>
</tr>
<tr>
<td>PKI Status 1</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This enhancement added the status keyword to the show crypto pki trustpoints command, which allows you to view the current status of the trustpoint. Prior to this enhancement, you had to issue the show crypto pki certificates and the show crypto pki timers commands for the current status. The following section provides information about this enhancement:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- How to Configure Certificate Enrollment for a PKI, page 83</td>
</tr>
<tr>
<td>Feature Name</td>
<td>Releases</td>
<td>Feature Information</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Reenroll Using Existing Certificates | Cisco IOS XE Release 2.1     | This feature allows users to reenroll a router with a Cisco IOS CA via existing certificates from a third-party vendor CA. The following section provides information about this enhancement:  
  - Configuring a Certificate Enrollment Profile for Enrollment or Reenrollment, page 99  
  The following commands were introduced by this feature: enrollment credential, grant auto trustpoint |
<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suite-B support in IOS SW</td>
<td>Cisco IOS XE Release 3.7S</td>
<td>Suite-B adds the following support for certificate enrollment for a PKI:</td>
</tr>
<tr>
<td>crypto</td>
<td></td>
<td>- Elliptic Curve Digital Signature Algorithm (ECDSA) (256 bit and 384 bit curves) is used for the signature operation within X.509 certificates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PKI support for validation of X.509 certificates using ECDSA signatures.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS.</td>
</tr>
<tr>
<td>Trustpoint CLI</td>
<td>Cisco IOS XE Release 2.1</td>
<td>Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPSec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support. The following sections provide information about this feature:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cisco IOS Suite-B Support for Certificate Enrollment for a PKI, page 81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring Certificate Enrollment or Autoenrollment, page 83</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This feature introduces the crypto pki trustpoint command, which adds support for trustpoint CAs.</td>
</tr>
</tbody>
</table>
Configuring Certificate Enrollment for a PKI

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Storing PKI Credentials

Public key infrastructure (PKI) credentials, such as Rivest, Shamir, and Adleman (RSA) keys and certificates can be stored in a specific location on the device, such as NVRAM and flash memory or on a USB eToken 64 KB smart card. USB tokens provide secure configuration distribution, RSA operations such as on-token key generation, signing, and authentication, and the storage of Virtual Private Network (VPN) credentials for deployment.

- Finding Feature Information, page 119
- Prerequisites for Storing PKI Credentials, page 119
- Restrictions for Storing PKI Credentials, page 120
- Information About Storing PKI Credentials, page 120
- How to Configure PKI Storage, page 122
- Configuration Examples for PKI Storage, page 136
- Additional References, page 138
- Feature Information for Storing PKI Credentials, page 139

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release.

To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support.

To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Storing PKI Credentials

Before you can specify the local certificate storage location, your system should meet the following requirements:

- A platform that supports storing PKI credentials as separate files.
- A configuration that contains at least one certificate.
- An accessible local file system.
- A Cisco supported USB token (Safenet/Aladdin eToken PRO 32 KB or 64 KB)
- To use the Cisco supported USB token, you must have Cisco IOS XE Release 3.6 or later installed on your router.
Restrictions for Storing PKI Credentials

When storing certificates to a local storage location, the following restrictions are applicable:

- Only local file systems may be used. An error message will be displayed if a remote file system is selected, and the command will not take effect.
- A subdirectory may be specified if supported by the local file system. NVRAM does not support subdirectories.

Information About Storing PKI Credentials

- Storing Certificates to a Local Storage Location, page 120
- PKI Credentials and USB Tokens, page 120

Storing Certificates to a Local Storage Location

Certificates are stored to NVRAM by default; however, some routers do not have the required amount of NVRAM to successfully store certificates.

All Cisco platforms support NVRAM and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, USB flash, or USB token.

During run time, you can specify what active local storage device you would like to use to store certificates.

PKI Credentials and USB Tokens

To use a secure USB token on your router, you should understand the following concepts:

- How a USB Token Works, page 120
- How RSA Keys are Used with a USB Token, page 121
- Benefits of USB Tokens, page 121

How a USB Token Works

A smart card is a small plastic card, containing a microprocessor and memory that allows you to store and process data. A USB token is a smart card with a USB interface. The token can securely store any type of file within its available storage space (32 KB). Configuration files that are stored on the USB token can be encrypted and accessed only via a user PIN. The device does not load the configuration file unless the proper PIN has been configured for secure deployment of device configuration files.

After you plug the USB token into the device, you must log into the USB token; thereafter, you can change default settings, such as the user PIN (default: 1234567890) and the allowed number of failed login attempts (default: 15 attempts) before future logins are refused. For more information on accessing and configuring the USB token, see the section “Logging Into and Setting Up the USB Token.”

After you have successfully logged into the USB token, you can copy files from the device on to the USB token via the copy command. USB token RSA keys and associated IPsec tunnels remain available until the device is reloaded. To specify the length of time before the keys are removed and the IPsec tunnels are torn down, issue the crypto pki token removal timeout command. The default timeout is zero, which causes
the RSA keys to be removed automatically after the eToken is removed from the device. The default appears in the running configuration as:

```plaintext
crypto pki token default removal timeout 0
```

The table below highlights the capabilities of the USB token.

### Table 5  Functionality Highlights for USB Tokens

<table>
<thead>
<tr>
<th>Function</th>
<th>USB Token</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accessibility</strong></td>
<td>Used to securely store and transfer digital certificates, preshared keys, and device configurations from the USB token to the device.</td>
</tr>
<tr>
<td><strong>Storage Size</strong></td>
<td>32 KB or 64 KB</td>
</tr>
</tbody>
</table>
| **File Types**         | • Typically used to store digital certificates, preshared keys, and device configurations for IPsec VPNs.  
                          | • USB tokens cannot store Cisco IOS images.                               |
| **Security**           | • Files can be encrypted and accessed only with a user PIN.              
                          | • Files can also be stored in a nonsecure format.                        |
| **Boot Configurations**| • The device can use the configuration stored in the USB token during boot time.  
                             | • The device can use the secondary configuration stored in the USB token during boot time. (A secondary configuration allows users to load their IPsec configuration.) |

### How RSA Keys are Used with a USB Token

- RSA keys are loaded after the USB token is successfully logged into the router.
- By default, newly generated RSA keys are stored on the most recently inserted USB token. Regenerated keys should be stored in the same location where the original RSA key was generated.

### Benefits of USB Tokens

USB token support on a Cisco router provides the following application benefits:

**Removable Credentials: Provide or Store VPN Credentials on an External Device for Deployment**

A USB token can use smart card technology to store a digital certificate and configuration for IPSec VPN deployment. This ability enhances the capability of the router to generate RSA public keys to authenticate at least one IPSec tunnel. (Because a router can initiate multiple IPSec tunnels, the USB token can contain several certificates, as appropriate.) Storing VPN credentials on an external device reduces the threat of compromising secure data.
PIN Configuration for Secure File Deployment

A USB token can store a configuration file that can be used for enabling encryption on the router via a user-configured PIN. (That is, no digital certificates, preshared keys, or VPNs are used.)

Touchless or Low Touch Configuration

The USB token can provide remote software configuration and provisioning with little or no human interaction. Configuration is set up as an automated process. That is, the USB token can store a bootstrap configuration that the router can use to boot from after the USB token has been inserted into the router. The bootstrap configuration connects the router to a TFTP server, which contains a configuration that completely configures the router.

RSA Operations

A USB token may be used as a cryptographic device in addition to a storage device. Using a USB token as a cryptographic device allows RSA operations such as key generation, signing, and authentication to be performed on the token.

General-purpose, special-usage, encryption, or signature RSA key pairs with a modulus of 2048 bits or less may be generated from credentials located on your token storage device. Private keys are not distributed and remain on the token by default, however you may configure the private key storage location.

Keys that reside on a USB token are saved to persistent token storage when they are generated. Key deletion will remove the keys stored on the token from persistent storage immediately. (Keys that do not reside on a token are saved to or deleted from non-token storage locations when the write memory or a similar command is issued.)

Remote Device Configuration and Provisioning in a Secure Device Provisioning (SDP) Environment

SDP may be used to configure a USB token. The configured USB token may be transported to provision a device at a remote location. That is, a USB token may be used to transfer cryptographic information from one network device to another remote network device providing a solution for a staged USB token deployment.

For information about using USB tokens with SDP, see document titles in the “Additional References” section.

How to Configure PKI Storage

- Specifying a Local Storage Location for Certificates, page 123
- Setting Up and Using USB Tokens on Cisco Devices, page 124
- Troubleshooting USB Tokens, page 134
Specifying a Local Storage Location for Certificates

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto pki certificate storage location-name
4. exit
5. copy source-url destination-url
6. show crypto pki certificates storage

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki certificate storage location-name</td>
<td>Specifies the local storage location for certificates.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# crypto pki certificate storage flash:/certs</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> copy source-url destination-url</td>
<td>(Optional) Saves the running configuration to the startup configuration.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# copy system:running-config nvram:startup-config</td>
<td></td>
</tr>
</tbody>
</table>

**Note** Settings will only take effect when the running configuration is saved to the startup configuration.
Command or Action | Purpose
--- | ---
Step 6 show crypto pki certificates storage | (Optional) Displays the current setting for the PKI certificate storage location.

Example:
Device# show crypto pki certificates storage

Example
The following is sample output from the `show crypto pki certificates storage` command, which shows that the certificates are stored in the certs subdirectory of disk0:

Device# show crypto pki certificates storage
Certificates will be stored in disk0:/certs/

Setting Up and Using USB Tokens on Cisco Devices

- Storing the Configuration on a USB Token, page 124
- Logging Into and Setting Up the USB Token, page 125
- Configuring the USB Token, page 127
- Setting Administrative Functions on the USB Token, page 131

Storing the Configuration on a USB Token

**SUMMARY STEPS**

1. enable
2. configure terminal
3. boot config usbtoken[0-9]:filename

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
</tbody>
</table>

Step 2 configure terminal | Enters global configuration mode.

Example:
Device# configure terminal
Step 3  **`boot config usbtoken[0-9]:filename`**  
Specifies that the startup configuration file is stored in a secure USB token.

**Example:**
```
Device(config)# boot config usbtoken0:file
```

## Logging Into and Setting Up the USB Token

- Configuring the Device for Automatic Login, page 125
- Configuring the Device for Manual Login, page 126
- What to Do Next, page 127

### Configuring the Device for Automatic Login

The automatic login allows the device to completely come back up without any user or operator intervention. The PIN is stored in the private NVRAM, so it is not visible in the startup or running configuration.

**Note**
A hand-generated startup configuration can contain the automatic login command for deployment purposes, but the `copy system:running-config nvram: startup-config` command must be issued to put the hand-generated configuration in the private configuration.

Perform this task to configure the automatic login on the device

**Note**
Either the manual or automatic login is required.

### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto pki token token-name user-pin [pin]`
4. `exit`
5. `show usbtoken0-9:filename`
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>- Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> configure terminal</td>
<td>Enters global configuration mode, which allows you to configure automatic USB token login.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki token token-name user-pin [pin]</td>
<td>Configures the device to log in to the token automatically, using the specified PIN at device startup or when the USB token is inserted into a USB slot.</td>
</tr>
<tr>
<td></td>
<td>The PIN is encrypted and stored in NVRAM.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>You will be asked to enter your passphrase.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# crypto pki token usbtoken0 user-pin 1234</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device(config)# exit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> show usbtoken0-9:filename</td>
<td>(Optional) Verifies whether the USB token has been logged on to the device.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# show usbtoken0:usbfile</td>
<td></td>
</tr>
</tbody>
</table>

**Configuring the Device for Manual Login**

Unlike automatic login, manual login requires that the user know the actual USB token PIN.

**Note**

Either the manual or automatic login is required.

Manual login can be used when storing a PIN on the device is not desirable. Manual login may also be suitable for some initial deployment or hardware replacement scenarios for which the device is obtained from the local supplier or drop-shipped to the remote site. Manual login can be executed with or without privileges, and it creates files and RSA keys on the USB token available to the Cisco IOS software. If a secondary configuration file is configured, it is executed only with the privileges of the user who is performing the login. Thus, if you want to use manual login and set up the secondary configuration on the USB token to perform anything useful, you need to enable privileges.

Manual login can also be used in recovery scenarios for which the device configuration has been lost. If the scenario contains a remote site that normally connects to the core network with a VPN, the loss of the...
configuration and RSA keys requires out-of-band services that the USB token can provide. The USB token can contain a boot configuration, a secondary configuration, or both, and RSA keys to authenticate the connection.

**SUMMARY STEPS**

1. `enable`
2. `crypto pki token token-name [admin] login [pin]`
3. `show usbtoken 0-9:filename`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td>Example: Device&gt; enable</td>
<td>• Enter your password if prompted.</td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto pki token token-name [admin] login [pin]</td>
<td>Manually logs into the USB token. If the admin keyword is not specified initially you can re-enter the crypto pki token command again with this keyword option.</td>
</tr>
<tr>
<td>Example: Device# crypto pki token usbtoken0 admin login 5678</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> show usbtoken 0-9:filename</td>
<td>(Optional) Verifies whether the USB token has been logged on to the device.</td>
</tr>
<tr>
<td>Example: Device# show usbtoken0:usbfile</td>
<td></td>
</tr>
</tbody>
</table>

**What to Do Next**

After you have logged into the USB token, it is available for use.

- To further configure the USB token, see the “Configuring the USB Token” section.
- To perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the “Setting Administrative Functions on the USB Token” section.

**Configuring the USB Token**

After you have set up automatic login, you may perform this task to further configure the USB token.

- PINs and Passphrases, page 128
- Unlocking and Locking the USB Token, page 128
- Secondary Configuration and Unconfiguration Files, page 128
- What to Do Next, page 130
PINs and Passphrases

For additional PIN security with automatic login, you may encrypt your PIN stored in NVRAM and set up a passphrase for your USB token. Establishing a passphrase allows you to keep your PIN secure; another user needs only to know the passphrase, not the PIN.

When the USB token is inserted into the device, the passphrase is needed to decrypt the PIN. Once the PIN is decrypted, the device can then use the PIN to log in to the USB token.

Note

The user needs a privilege level of 1 to log in.

Unlocking and Locking the USB Token

The USB token itself can be locked (encrypted) or unlocked (decrypted).

Unlocking the USB token allows it to be used. Once unlocked, Cisco IOS software treats the token as if it were automatically logged in. Any keys on the USB token are loaded, and if a secondary configuration file is on the token, it is executed with full user privileges (privilege level 15) independent of the privilege level of the logged-in user.

Locking the token, unlike logging out of the token, deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if configured.

Secondary Configuration and Unconfiguration Files

Configuration files that exist on a USB token are called secondary configuration files. If you create and configure a secondary configuration file, it is executed after the token is logged in. The existence of a secondary configuration file is determined by the presence of a secondary configuration file option in the Cisco IOS configuration stored in NVRAM. When the token is removed or logged out and the removal timer expires, a separate secondary unconfiguration file is processed to remove all secondary configuration elements from the running configuration. Secondary configuration and secondary unconfiguration files are executed at privilege level 15 and are not dependent on the level of the user logged in.

SUMMARY STEPS

1. enable
2. crypto pki token token-name unlock [pin]
3. configure terminal
4. crypto pki token token-name encrypted-user-pin [write]
5. crypto pki token token-name secondary unconfig file
6. exit
7. crypto pki token token-name lock [pin]
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | `enable` | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| **Example:** | Device> `enable` | |
| **Step 2** | `crypto pki token token-name unlock [pin]` | (Optional) Allows the token to be used if the USB token has been locked.  
  Once unlocked, Cisco IOS software treats the token as if it has been automatically logged in. Any keys on the token are loaded and if a secondary configuration file exists, it is executed. |
| **Example:** | Device# `crypto pki token mytoken unlock mypin` | |
| **Step 3** | `configure terminal` | Enters global configuration mode. |
| **Example:** | Device# `configure terminal` | |
| **Step 4** | `crypto pki token token-name encrypted-user-pin [write]` | (Optional) Encrypts the stored PIN in NVRAM. |
| **Example:** | Device(config)# `crypto pki token mytoken encrypted-user-pin write` | |
| **Step 5** | `crypto pki token token-name secondary unconfig file` | (Optional) Specifies the secondary configuration file and its location. |
| **Example:** | Device(config)# `crypto pki token mytoken secondary unconfig configs/myunconfigfile.cfg` | |
| **Step 6** | `exit` | Enters privileged EXEC mode. |
| **Example:** | Device(config)# `exit` | |
| **Step 7** | `crypto pki token token-name lock [pin]` | (Optional) Deletes any RSA keys loaded from the token and runs the secondary unconfiguration file, if it exists. |
| **Example:** | Device# `crypto pki token mytoken lock mypin` | |
Examples

The following example shows both the configuration and encryption of a user PIN and then the device reloading and the user PIN being unlocked:

! Configuring the user PIN

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# crypto pki token usbtoken0: userpin

Enter password: mypassword

! Encrypt the user PIN

Device(config)# crypto pki token usbtoken0: encrypted-user-pin

Enter passphrase: mypassphrase

Device(config)# exit

Device#

Sep 20 21:51:38.076: %SYS-5-CONFIG_I: Configured from console by console

Device# show running config

crypto pki token usbtoken0 user-pin *encrypted*

! Reloading the router.

Device> enable

Password:

! Decrypting the user pin.

Device# crypto pki token usbtoken0: unlock

Token eToken is usbtoken0

Enter passphrase: mypassphrase

Token login to usbtoken0(eToken) successful

Device#

Sep 20 22:31:13.128: %CRYPTO-6-TOKENLOGIN: Cryptographic Token eToken Login Successful

The following example shows a how a secondary unconfiguration file might be used to remove secondary configuration elements from the running configuration. For example, a secondary configuration file might be used to set up a PKI trustpoint. A corresponding unconfiguration file, named mysecondaryunconfigfile.cfg, might contain this command line:

no crypto pki trustpoint token-tp

If the token were removed and the following commands executed, the trustpoint and associated certificates would be removed from the device’s running configuration:

Device# configure terminal
Device(config)# no crypto pki trustpoint mytoken secondary unconfig mysecondaryunconfigfile.cfg

What to Do Next

After you have logged into and configured the USB token, it is available for use. If you want to perform USB token administrative tasks, such as changing the user PIN, copying files from the router to the USB token set key storage location, and changing USB tokens, see the “Setting Administrative Functions on the USB Token” section.
Setting Administrative Functions on the USB Token

Perform this task to change default settings, such as the user PIN, the maximum number of failed attempts on the USB token, or the credential storage location.

SUMMARY STEPS

1. enable
2. crypto pki token token-name admin | change-pin [pin]
3. crypto pki token token-name device-name: label token-label
4. configure terminal
5. crypto key storage device-name:
6. crypto key generate rsa [general-keys | usage-keys | signature | encryption] [label key-label] [exportable] [modulus modulus-size] [storage device-name:] [redundancy] [on device-name]:
7. crypto key move rsa keylabel [non-exportable | [on | storage]] location
8. crypto pki token {token-name | default} removal timeout [seconds]
9. crypto pki token {token-name | default} max-retries [number]
10. exit
11. copy usbflash[0-9]:filename destination-url
12. show usbtok0[0-9]:filename
13. crypto pki token token-name logout

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> enable</td>
<td>Enables privileged EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device&gt; enable</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> crypto pki token token-name admin</td>
<td>(Optional) Changes the user PIN number on the USB token.</td>
</tr>
<tr>
<td>change-pin [pin]</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> Device# crypto pki token usbtok0 admin change-pin</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> After the PIN has been changed, you must reset the login failure count to zero (via the crypto pki token max-retries command). The maximum number of allowable login failures is set (by default) to 15.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> crypto pki token token-name device-name: label token-label</td>
<td>(Optional) Sets or changes the name of the USB token.</td>
</tr>
<tr>
<td><strong>Example:</strong> Device# crypto pki token mytoken usb0: label newlabel</td>
<td></td>
</tr>
<tr>
<td><strong>Tip</strong> This command is useful when configuring multiple USB tokens for automatic login, secondary configuration files, or other token specific settings.</td>
<td></td>
</tr>
</tbody>
</table>
### Step 4
**configure terminal**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Device# configure terminal</td>
<td></td>
</tr>
</tbody>
</table>

### Step 5
**crypto key storage device-name**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>(Optional) Sets the default RSA key storage location for newly created keys.</td>
</tr>
<tr>
<td>Device(config)# crypto key storage usbtoken0:</td>
<td></td>
</tr>
</tbody>
</table>

**Note** Regardless of configuration settings, existing keys are stored on the device from where they were originally loaded.

### Step 6
**crypto key generate rsa [general-keys | usage-keys | signature | encryption] [key-label] [exportable] [modulus modulus-size] [storage device-name] [redundancy] [on device-name]:**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td>(Optional) Generates the RSA key pair for the certificate server.</td>
</tr>
<tr>
<td>Device(config)# crypto key generate rsa label tokenkey1 storage usbtoken0:</td>
<td></td>
</tr>
</tbody>
</table>

- The `storage` keyword specifies the key storage location.
- When specifying a label name by specifying the `key-label` argument, you must use the same name for the label that you plan to use for the certificate server (through the `crypto pki server cs-label` command). If a `key-label` argument is not specified, the default value, which is the fully qualified domain name (FQDN) of the device, is used.
- If the exportable RSA key pair is manually generated after the CA certificate has been generated, and before issuing the `no shutdown` command, then use the `crypto ca export pkcs12` command to export a PKCS12 file that contains the certificate server certificate and the private key.
  - By default, the modulus size of a CA key is 1024 bits. The recommended modulus for a CA key is 2048 bits. The range for a modulus size of a CA key is from 350 to 4096 bits.
  - The `on` keyword specifies that the RSA key pair is created on the specified device, including a Universal Serial Bus (USB) token, local disk, or NVRAM. The name of the device is followed by a colon (:) |

**Note** Keys created on a USB token must be 2048 bits or less.
### Command or Action

| Step 7 | crypto key move rsa keylabel [non-exportable | [on | storage]] location |
|--------|--------------------------------------------------|
| Purpose| (Optional) Moves existing Cisco IOS credentials from the current storage location to the specified storage location.  
By default, the RSA key pair remains stored on the current device.  
Generating the key on the device and moving it to the token takes less than a minute. Generating a key on the token, using the on keyword could take five to ten minutes, and is dependent on hardware key generation routines available on the USB token.  
When an existing RSA key pair is generated in Cisco IOS, stored on a USB token, and used for an enrollment, it may be necessary to move those existing RSA key pairs to an alternate location for permanent storage.  
This command is useful when using SDP with USB tokens to deploy credentials. |

**Example:**
```
Device(config)# crypto key move rsa keypairname non-exportable on token
```

| Step 8 | crypto pki token {token-name | default} removal timeout [seconds] |
|--------|-------------------------------------------------------------------|
| Purpose| (Optional) Sets the time interval, in seconds, that the device waits before removing the RSA keys that are stored in the USB token after the USB token has been removed from the device.  
**Note** If this command is not issued, all RSA keys and IPsec tunnels associated with the USB token are torn down immediately after the USB token is removed from the device. |

**Example:**
```
Device(config)# crypto pki token usbtoken0 removal timeout 60
```

| Step 9 | crypto pki token {token-name | default} max-retries [number] |
|--------|------------------------------------------------------------------|
| Purpose| (Optional) Sets the maximum number of consecutive failed login attempts allowed before access to the USB token is denied.  
• By default, the value is set at 15. |

**Example:**
```
Device(config)# crypto pki token usbtoken0 max-retries 20
```

<table>
<thead>
<tr>
<th>Step 10</th>
<th>exit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose</td>
<td>Exits global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**
```
Device(config)# exit
```

<table>
<thead>
<tr>
<th>Step 11</th>
<th>copy usbflash[0-9]:filename destination-url</th>
</tr>
</thead>
</table>
| Purpose| Copies files from USB token to the device.  
• destination-url—See the copy command page documentation for a list of supported options. |

**Example:**
```
Device# copy usbflash0:file1 nvram:
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 12</strong> show usbtoken[0-9]:filename</td>
<td><em>(Optional)</em> Displays information about the USB token. You can use this command to verify whether the USB token has been logged in to the device.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# show usbtoken:usbfile</td>
<td></td>
</tr>
<tr>
<td><strong>Step 13</strong> crypto pki token token-name logout</td>
<td>Logs the device out of the USB token.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>Device# crypto pki token usbstoken0 logout</td>
<td></td>
</tr>
<tr>
<td><strong>Troubleshooting USB Tokens</strong></td>
<td></td>
</tr>
</tbody>
</table>

This section contains descriptions of the following Cisco IOS commands that can be used to help troubleshoot possible problems that may arise while using a USB token:

- Troubleshooting the USB Port Connection, page 134
- Determining USB Token Support Connectivity, page 135
- Displaying USB Token Information, page 135

**Troubleshooting the USB Port Connection**

Use the **show file systems** command to determine whether the router recognizes that there is a USB module plugged into a USB port. The USB module should appear on the list of file systems. If the module does not appear on the list, it can indicate any of the following problems:

- A connection problem with the USB module.
- The Cisco IOS image running on the router does not support a USB module.
- A hardware problem with the USB module itself.

Sample output from the **show file systems** command showing a USB token appears below. The USB module listing appears in the last line of the examples.

```
Device# show file systems
File Systems:                                       Type    Flags    Prefixes
Size(b)   Free(b)  Name          rw    archive:         
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
129880064 69414912 disk          rw    flash:#       
491512    486395   nvram        rw    nvram:          
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
-         -         -             -    -                
```
Determining USB Token Support Connectivity

Use the `show usb-devices summary` command to determine if a USB token device is supported by Cisco (see the text in bold).

```
Device# show usb-devices summary

USB Device: OHCI Host Controller
Bus: 03 Port: 00 Cnt: 00 Speed: 12
Vendor: 1d6b ProdID: 0001 Rev: 2.06
Serial Number: 0001:01:11.1

USB Device: OHCI Host Controller
Bus: 02 Port: 00 Cnt: 00 Speed: 12
Vendor: 1d6b ProdID: 0001 Rev: 2.06
Serial Number: 0001:01:11.0

USB Device: Token 4.28.1.1 2.7.195
Bus: 02 Port: 00 Cnt: 01 Speed: 12
Vendor: 0529 ProdID: 0600 Rev: 1.00
Serial Number:

USB Device: EHCI Host Controller
Bus: 01 Port: 00 Cnt: 00 Speed: 480
Vendor: 1d6b ProdID: 0002 Rev: 2.06
Serial Number: 0001:01:11.2

USB Device: eUSB
Bus: 01 Port: 03 Cnt: 01 Speed: 480
Vendor: 0e39 ProdID: 2b00 Rev: b9.00
Serial Number: 1E884812183636210510
```

Displaying USB Token Information

Use the `dir` command with the `filesystem` keyword option `usbtoken0-9` to display all files, directories, and their permission strings on the USB token.

The following sample output displays directory information for the USB token:

```
Device# dir usbtoken1:
Directory of usbtoken1:/
  2  d---   64 Dec 22 2032 05:23:40 +00:00  1000
  5  d---   4096 Dec 22 2032 05:23:40 +00:00  1001
  8  d---   0 Dec 22 2032 05:23:40 +00:00  1002
10  d---   512 Dec 22 2032 05:23:42 +00:00  1003
12  d---   0 Dec 22 2032 05:23:42 +00:00  5000
13  d---   0 Dec 22 2032 05:23:42 +00:00  6000
14  d---   0 Dec 22 2032 05:23:42 +00:00  7000
15  ----   940 Jun 27 1992 12:50:42 +00:00 mystartup-config
16  ----  1423 Jun 27 1992 12:51:14 +00:00 myrunning-config
32768 bytes total (858 bytes free)
```

The following sample output displays directory information for all devices to which the device is aware:

```
Device# dir all-filesystems
Directory of archive:/
No files in directory
No space information available
Directory of system:/
  2  drwx   0 <no date> its
 115 dr-x   0 <no date> lib
144 dr-x   0 <no date> memory
 1  -rw-  1906 <no date> running-config
114 dr-x   0 <no date> vfiles
```
Configuration Examples for PKI Storage

Example: Storing Certificates to a Specific Local Storage Location

The following configuration example shows how to store certificates to the certs subdirectory. The certs subdirectory does not exist and is automatically created.

Router# dir nvram:
114 -rw- 4687 <no date> startup-config
115 ---- 5545 <no date> private-config
116 -rw- 4687 <no date> underlying-config
  1 ----  34 <no date> persistent-data
  3 -rw- 707 <no date> ioscaroot#7401CA.cer
  9 -rw- 863 <no date> msca-root#826E.cer
 10 -rw- 759 <no date> msca-root#1BA8CA.cer
 11 -rw- 863 <no date> msca-root#75B8.cer
 24 -rw- 1149 <no date> storagename#6500CA.cer
 26 -rw- 863 <no date> msca-root#83EE.cer
129016 bytes total (92108 bytes free)

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# crypto pki certificate storage disk0:/certs
Requested directory does not exist -- created
Certificates will be stored in disk0:/certs/
Router(config)# end
Router# write
*May 27 02:09:00:%SYS-5-CONFIG_I:Configured from console by consolemem
Building configuration...[OK]
Router# directory disk0:/certs
Directory of disk0:/certs/
 14 -rw- 707 May 27 2005 02:09:02 +00:00 ioscaroot#7401CA.cer
 15 -rw- 863 May 27 2005 02:09:02 +00:00 msca-root#826E.cer
 16 -rw- 759 May 27 2005 02:09:02 +00:00 msca-root#1BA8CA.cer
 17 -rw- 863 May 27 2005 02:09:02 +00:00 msca-root#75B8.cer
 18 -rw- 1149 May 27 2005 02:09:02 +00:00 storagename#6500CA.cer
 19 -rw- 863 May 27 2005 02:09:02 +00:00 msca-root#83EE.cer
47894528 bytes total (2093456 bytes free)
! The certificate files are now on disk0/certs:
Example: Logging Into a USB Token and Saving RSA Keys to the USB Token

The following configuration example shows how to log in to the USB token, generate RSA keys, and store the RSA keys on the USB token:

! Configure the router to automatically log into the eToken
configure terminal
configure terminal
crypto pki token default user-pin 0 1234567890
! Generate RSA keys and enroll certificates with the CA.
crypto pki trustpoint IOSCA
enrollment url http://10.23.2.2
exit
crypto ca authenticate IOSCA
Certificate has the following attributes:
Fingerprint MD5:23272BD4 37E3D9A4 236F7E1A F534444E
Fingerprint SHA1:D1B4D9F8 D603249A 793B3CAF 8342E1FE 3934EB7A
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
crypto pki enroll
crypto pki enroll IOSCA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
% password to the CA Administrator in order to revoke your certificate.
% For security reasons your password will not be saved in the configuration.
% Please make a note of it.
Password:
Re-enter password:
% The subject name in the certificate will include:c2851-27.cisco.com
% Include the router serial number in the subject name? [yes/no]:no
% Include an IP address in the subject name? [no]:no
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% Certificate request sent to Certificate Authority
% The "show crypto ca certificate IOSCA verbose" command will show the fingerprint.
Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint MD5:E6DDAB1B
0E30EFE6 54529D8A DA787DBA
Jan 13 06:47:19.413:CRYPTO_PKI: Certificate Request Fingerprint SHA1:3B0F33B
7F5C02A10 3935042B C4B6C50D 6139251
Jan 13 06:47:21.021:%PKI-6-CERTRET:Certificate received from Certificate Authority
% Issue the write memory command, which will automatically save the RSA keys to the
% eToken ! instead of private NVRAM.
Router# write memory
Building configuration...
[OK]
* Jan 13 06:47:29.481:%CRYPTO-6-TOKENSTOREKEY:Key c2851-27.cisco.com stored on
Cryptographic Token eToken Successfully

The following sample output from the show crypto key mypubkey rsa command displays stored credentials after they are successfully loaded from the USB token. Credentials that are stored on the USB token are in the protected area. When storing the credentials on the USB token, the files are stored in a directory called /keystore. However, the key files are hidden from the command-line interface (CLI).

Router# show crypto key mypubkey rsa
% Key pair was generated at:06:37:26 UTC Jan 13 2005
Key name:c2851-27.cisco.com
Usage:General Purpose Key
Key is not exportable.
Key Data:
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43A70DD
732E0F4E 3CA0CDAB 387AF0F5 EBB22F24 2A3C1FAE 5D51FE83 F3DEA954 7FBD3603
7C977854 88E99B9B 7FCC93021 7F46ABP8 A8BA26D6 172B3109 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
% Key name: c2851-27.cisco.com.server
Usage:Encryption Key
Key is not exportable.
Key Data:
307C300D 06092A86 4886F70D 01010105 00034B00 30480241 00E3C644 43A70DD
732E0F4E 3CA0CDAB 387AF0F5 EBB22F24 2A3C1FAE 5D51FE83 F3DEA954 7FBD3603
7C977854 88E99B9B 7FCC93021 7F46ABP8 A8BA26D6 172B3109 B5020301 0001
% Key pair was generated at:06:37:27 UTC Jan 13 2005
Usage:Encryption Key
Key is not exportable.
Key Data:
## Additional References

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connecting the USB modules to the router</td>
<td>Cisco Access Router USB Flash Module and USB eToken Hardware Installation Guide</td>
</tr>
<tr>
<td>eToken and USB flash data sheet</td>
<td>USB eToken and USB Flash Features Support</td>
</tr>
<tr>
<td>RSA keys</td>
<td>Deploying RSA Keys Within a PKI</td>
</tr>
<tr>
<td>File management (loading, copying, and rebooting files)</td>
<td>Cisco Configuration Fundamentals Configuration Guide on Cisco.com</td>
</tr>
<tr>
<td>USB Token RSA Operations: Using USB tokens for RSA operations upon initial autoenrollment</td>
<td>See the “Configuring Certificate Enrollment or Autoenrollment” section of the “Configuring Certificate Enrollment for a PKI” feature document.</td>
</tr>
<tr>
<td>SDP setup, configuration and use with USB tokens</td>
<td>See the feature information section for the feature names on using SDP and USB tokens to deploy PKI credentials in the “Setting Up Secure Device Provisioning (SDP) for Enrollment in a PKI” feature document.</td>
</tr>
</tbody>
</table>

### Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</td>
<td><a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a></td>
</tr>
</tbody>
</table>
**Feature Information for Storing PKI Credentials**

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate—Storage Location</td>
<td>Cisco IOS XE Release 2.1</td>
<td>This feature allows you to specify the storage location of local certificates for platforms that support storing certificates as separate files. All Cisco platforms support NVRAM, which is the default location, and flash local storage. Depending on your platform, you may have other supported local storage options including bootflash, slot, disk, or USB flash. The following commands were introduced by this feature: <code>crypto pki certificate storage</code>, <code>show crypto pki certificates storage</code></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB eToken 64KB Smart Card Support</td>
<td>Cisco IOS XE Release 3.6</td>
<td>This feature allows USB eToken 64 KB smart card support for RSA key and certificate storage location. USB tokens provide secure configuration distribution, RSA operations such as on-token key generation, signing, and authentication, and the storage of Virtual Private Network (VPN) credentials for deployment. The following commands were introduced by this feature: binary file, crypto key move rsa, crypto key storage, crypto pki token change-pin, crypto pki token encrypted-user-pin, crypto pki token label, crypto pki token lock, crypto pki token login, crypto pki token logout, crypto pki token max-retries, crypto pki token removal timeout, crypto pki token secondary config, crypto pki token unlock, crypto pki token user-pin, show usb-devices summary, show usb driver, show usbtoken, template file.</td>
</tr>
</tbody>
</table>

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.
Source Interface Selection for Outgoing Traffic with Certificate Authority

The Source Interface Selection for Outgoing Traffic with Certificate Authority feature allows you to specify that the address of an interface be used as the source address for all outgoing TCP connections associated with that trustpoint when a designated trustpoint has been configured.

- Finding Feature Information, page 141
- Information About Source Interface Selection for Outgoing Traffic with Certificate Authority, page 141
- How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority, page 142
- Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority, page 145
- Additional References, page 146
- Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority, page 147
- Glossary, page 147

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Source Interface Selection for Outgoing Traffic with Certificate Authority

- Certificates That Identify an Entity, page 142
- Source Interface for Outgoing TCP Connections Associated with a Trustpoint, page 142
Certificates That Identify an Entity

Certificates can be used to identify an entity. A trusted server, known as the certification authority (CA), issues the certificate to the entity after determining the identity of the entity. A router that is running Cisco IOS XE software obtains its certificate by making a network connection to the CA. Using the Simple Certificate Enrollment Protocol (SCEP), the router transmits its certificate request to the CA and receives the granted certificate. The router obtains the certificate of the CA in the same manner using SCEP. When validating a certificate from a remote device, the router may again contact the CA or a Lightweight Directory Access Protocol (LDAP) or HTTP server to determine whether the certificate of the remote device has been revoked. (This process is known as checking the certificate revocation list [CRL].)

In some configurations, the router may make the outgoing TCP connection using an interface that does not have a valid or routable IP address. The user must specify that the address of a different interface be used as the source IP address for the outgoing connection. Cable modems are a specific example of this requirement because the outgoing cable interface (the RF interface) usually does not have a routable address. However, the user interface (usually FastEthernet) does have a valid IP address.

Source Interface for Outgoing TCP Connections Associated with a Trustpoint

The `crypto pki trustpoint` command is used to specify a trustpoint. The `source interface` command is used along with the `crypto pki trustpoint` command to specify the address of the interface that is to be used as the source address for all outgoing TCP connections associated with that trustpoint.

*Note*

If the interface address is not specified using the `source interface` command, the address of the outgoing interface is used.

How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority

- Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint, page 142

Configuring the Interface for All Outgoing TCP Connections Associated with a Trustpoint

Perform this task to configure the interface that you want to use as the source address for all outgoing TCP connections associated with a trustpoint.
### SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `crypto pki trustpoint name`
4. `enrollment url url`
5. `source interface interface-address`
6. `interface type slot/port`
7. `description string`
8. `ip address ip-address mask`
9. `interface type slot/port`
10. `description string`
11. `ip address ip-address mask`
12. `crypto map map-name`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** enable | Enables privileged EXEC mode.  
  - Enter your password if prompted. |
| **Example:**  
  ```  
  Router> enable  
  ``` | |
| **Step 2** `configure terminal` | Enters global configuration mode. |
| **Example:**  
  ```  
  Router# configure terminal  
  ``` | |
| **Step 3** `crypto pki trustpoint name` | Declares the Certificate Authority (CA) that your router should use and enters ca-trustpoint configuration mode. |
| **Example:**  
  ```  
  Router (config)# crypto pki trustpoint ms-ca  
  ``` | |
| **Step 4** `enrollment url url` | Specifies the enrollment parameters of your CA. |
| **Example:**  
  ```  
  Router (ca-trustpoint)# enrollment url http://yourname:80/certsrv/mscep/mscep.dll  
  ``` | |
### How to Configure Source Interface Selection for Outgoing Traffic with Certificate Authority

#### Command or Action | Purpose
--- | ---
**Step 5** source interface interface-address | Interface to be used as the source address for all outgoing TCP connections associated with that trustpoint.

**Example:**

```
Router (ca-trustpoint)# interface fastethernet1/0
```

**Step 6** interface type slot / port | Configures an interface type and enters interface configuration mode.

**Example:**

```
Router (ca-trustpoint)# interface fastethernet1/0
```

**Step 7** description string | Adds a description to an interface configuration.

**Example:**

```
Router (config-if)# description inside interface
```

**Step 8** ip address ip-address mask | Sets a primary or secondary IP address for an interface.

**Example:**

```
Router (config-if)# ip address 10.1.1.1 255.255.255.0
```

**Step 9** interface type slot / port | Configures an interface type.

**Example:**

```
Router (config-if)# interface fastethernet1/0
```

**Step 10** description string | Adds a description to an interface configuration.

**Example:**

```
Router (config-if)# description outside interface 10.1.1.205 255.255.255.0
```

**Step 11** ip address ip-address mask | Sets a primary or secondary IP address for an interface.

**Example:**

```
Router (config-if)# ip address 10.2.2.205 255.255.255.0
```
Command or Action | Purpose
---|---
Step 12 crypto map map-name | Applies a previously defined crypto map set to an interface.

Example:

Router (config-if)# crypto map mymap

- Troubleshooting Tips, page 145

**Troubleshooting Tips**

Ensure that the interface specified in the command has a valid address. Attempt to ping the router using the address of the specified interface from another device (possibly the HTTP or LDAP server that is serving the CRL). You can do the same thing by using a traceroute to the router from the external device.

You can also test connectivity between the router and the CA or LDAP server by using Cisco IOS XE command-line interface (CLI). Enter the `ping ip` command and respond to the prompts. If you answer “yes” to the “Extended commands [n]:” prompt, you will be able to specify the source address or interface.

In addition, you can use Cisco IOS XE CLI to input a traceroute command. If you enter the `traceroute ip` command (in EXEC mode), you will be prompted for the destination and source address. You should specify the CA or LDAP server as the destination and the address of the interface that you specified in the “source interface” as the source address.

**Configuration Examples for Source Interface Selection for Outgoing Traffic with Certificate Authority**

**Source Interface Selection for Outgoing Traffic with Certificate Authority Example**

In the following example, the router is located in a branch office. The router uses IP Security (IPSec) to communicate with the main office. FastEthernet 1 is the “outside” interface that connects to the Internet Service Provider (ISP). FastEthernet 0 is the interface connected to the LAN of the branch office. To access the CA server located in the main office, the router must send its IP datagrams out interface FastEthernet 1 (address 10.2.2.205) using the IPSec tunnel. Address 10.2.2.205 is assigned by the ISP. Address 10.2.2.205 is not a part of the branch office or main office.

The CA cannot access any address outside the company because of a firewall. The CA sees a message coming from 10.2.2.205 and cannot respond (that is, the CA does not know that the router is located in a branch office at address 10.1.1.1, which it is able to reach).

Adding the `source interface` command tells the router to use address 10.1.1.1 as the source address of the IP datagram that it sends to the CA. The CA is able to respond to 10.1.1.1.

This scenario is configured using the `source interface` command and the interface addresses as described above.

```
crypto pki trustpoint ms-ca
```
enrollment url http://ms-ca:80/certsrv/mscep/mscep.dll
source interface fastethernet0
  interface fastethernet 0
description inside interface
  ip address 10.1.1.1 255.255.255.0
  interface fastethernet 1
description outside interface
  ip address 10.2.2.205 255.255.255.0
crypto map main-office

## Additional References

The following sections provide references related to the Source Interface Selection for Outgoing Traffic with Certificate Authority feature.

### Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring IPSec and certification authority</td>
<td>Security for VPNs with IPsec</td>
</tr>
<tr>
<td>IPSec and certification authority commands</td>
<td>Cisco IOS Security Command Reference</td>
</tr>
</tbody>
</table>

### Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified standards are supported by this feature.</td>
<td>-</td>
</tr>
</tbody>
</table>

### MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified MIBs are supported by this feature.</td>
<td>To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></td>
</tr>
</tbody>
</table>

### RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>No new or modified RFCs are supported by this feature.</td>
<td>-</td>
</tr>
</tbody>
</table>
Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
<tr>
<td>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</td>
<td></td>
</tr>
<tr>
<td>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</td>
<td></td>
</tr>
</tbody>
</table>

Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

**Table 7** Feature Information for Source Interface Selection for Outgoing Traffic with Certificate Authority

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Releases</th>
<th>Feature Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Interface Selection for Outgoing Traffic with Certificate Authority.</td>
<td>Cisco IOS XE Release 2.1</td>
<td>The following command was introduced: <code>source interface</code>.</td>
</tr>
</tbody>
</table>

Glossary

authenticate--To prove the identity of an entity using the certificate of an identity and a secret that the identity poses (usually the private key corresponding to the public key in the certificate).

CA --Certificate Authority. A CA is an entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

CA authentication --The user manually approves a certificate from a root CA. Usually a fingerprint of the certificate is presented to the user, and the user is asked to accept the certificate based on the fingerprint. The certificate of a root CA is signed by itself (self-signed) so that it cannot be automatically authenticated using the normal certificate verification process.
CRL -- certificate revocation list. A CRL is a data structure that enumerates digital certificates that have been invalidated by their issuer prior to when they were scheduled to expire.

enrollment -- A router receives its certificate via the enrollment process. The router generates a request for a certificate in a specific format (known as PKCS #10). The request is transmitted to a CA, which grants the request and generates a certificate encoded in the same format as the request. The router receives the granted certificate and stores it in an internal database for use during normal operations.

certificate -- A data structure defined in International Organization for Standardization (ISO) standard X.509 to associate an entity (machine or human) with the public key of that entity. The certificate contains specific fields, including the name of the entity. The certificate is normally issued by a CA on behalf of the entity. In this case the router will act as its own CA. Common fields within a certificate include the distinguished name (DN) of the entity, the DN of the authority issuing the certificate, and the public key of the entity.

LDAP -- Lightweight Directory Access Protocol. A LDAP is a protocol that provides access for management and browser applications that provide read-and-write interactive access to the X.500 directory.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.