# PKI Service for Large Scale IPSec Aggregation

**Revised: April 17, 2009, OL-19467-01**

## Contents


- PKI for Large Scale IPSec Introduction 2
- PKI Overview 3
  - Public Key Algorithms 3
    - Asymmetric Key Pair 3
    - Digital Signatures 4
    - Digital Signature Verification 4
    - What is a Certificate? 5
    - How Does IKE use Certificates? 6
    - How do we Get a Certificate 6
  - Deployment Scenarios 7
    - Branch/WAN Deployment 7
    - Internet Edge Deployment 8
    - Remote Access Deployment 9
- PKI Design Components 10
  - CA Server Models 10
  - When to Use Multiple CAs 11
  - Enrollment of Certificates 12
  - Maintaining a Database of Certificates 12
  - Rollover of Certificates 12
  - Renewal of Certificates 12
  - Revocation of Certificates 13
- PKI Deployment Basics 13


**Corporate Headquarters:**
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

Prerequisites 13
What You Need to Know Before You Begin 13
IP Addressing Requirements 14
Scaling Information 14
Sample PKI Architecture 14
Requirements 15
PKI Implementation Guidelines 15
PKI 16
Setting up Root CA Server 16
Setting up the Subordinate CA Architecture 17
Certificate Renewal 20
Recovery of PKI Certificate During Outages 22
Best Practices for Deploying PKI for a Large-Scale IPSec Solution 27
DMVPN Enrollment Using PKI 28
Hub-and-Spoke Enrollment with the PKI 29
Spoke Enrollment with *ra-subca* 29
Hub Enrollment with a Single Subordinate CA (*ra-subca)* 32
DMVPN Configuration 34
DMVPN Hub Configuration 36
DMVPN with PKI using Multiple Subordinate CAs 36
Migration of DMVPN from Pre-shared to PKI 41
Revocation Checking 48
Troubleshooting PKI Deployment 56
Troubleshooting PKI 57
Problem: Storage Location Not Accessible 57
Problem: Mismatched Subordinate CA Names 58
Troubleshooting DMVPN with PKI 61
Problem: Certificate Invalid Due to Revocation or Expired Certificate 61
Problem: Clock not Synchronized or Certificate Expired 63

# PKI for Large Scale IPSec Introduction

Today's complex digital world features a level of collaboration that demands an analogous and heightened level of secure communications that grows minute by minute. Virtual private networking (VPN) is a key security technology that is an essential element for ensuring secure communications for the entire range of Cisco customers, partners, and vendors.

A key question that arises when an organization is considering VPN deployment is this: *How do we trust the identity of users?* Moreover, the peers with which any organization might want to communicate can use different VPN technologies, such as IP Security (IPSec), Dynamic Multipoint VPN (DMVPN),

Group Encrypted Transport VPN (GETVPN), or Secure Socket Layer (SSL) VPN. Furthermore, the same VPN gateway might have to terminate all these connections. The challenge then is this: How do we trust the identity of these different peers using different technologies?

# PKI Overview

There are two methods to deploy authentication for VPN technologies:

- Pre-shared keys (PSK)
- Public Key Infrastructure (PKI)

Pre-shared keys are easy to deploy and highly scalable, but become very difficult to manage. Any VPN peer with the knowledge of the password can establish the VPN session to the headend in a pre-shared key environment. Moreover, if there are multiple headends then the passwords must be synchronized between the headend—which makes it inherently more difficult to manage. If spoke-to-spoke communication is required then you must maintain passwords between the spokes. Taken together, these issues can confuse the process of securing a large deployment. To avoid this, some organizations can use wild card-based strategy for authentication, which involves matching any IP address with one password. This can create security risk for the solution. Another weakness in pre-shared key-based security is the difficulty in removing spokes off the network. Removing spokes requires the removal of each associated pair-wise password from the headends or the spokes.

PKI addresses all the issues encountered with a pre-shared key authentication strategy. The significant benefits of PKI solutions including the following:

- PKI supports hierarchical architectures, thereby scaling to large number of sessions.
- PKI is highly secure because it uses public key cryptography.
- PKI is easier to manage; certificates can be added, deleted, or revoked.

# Public Key Algorithms

The following sections provide a summary of public key algorithm technology:


- Asymmetric Key Pair, page 3
- Digital Signatures, page 4
- Digital Signature Verification, page 4
- What is a Certificate?, page 5
- How Does IKE use Certificates?, page 6
- How do we Get a Certificate, page 6


## Asymmetric Key Pair

There are two components involved in a public key-base encryption solution: the *public key* and the *private key*. The user gives a public key to other users (through certificates) and keeps the private key. Conceptually, data encrypted with a public key can only be decrypted with the corresponding private key and data encrypted with a specific private can only be de-encrypted with the corresponding public key. In most deployments, the data is encrypted with private key, which is decrypted with the public key. The problem with an asymmetric key pair is that anybody can encrypt with the receiver's public key. For example, if Alice wants to talk to Bob, she can choose a session key derived out of the Bob's public key

and then communicate with Bob. Since, Bob knows his private key (which he will not share with anybody), he can decrypt the message. However, another individual (a *man-in-middle* we'll call Trucker) might impersonate Alice by initiating communication to Bob as if he is Alice. To ensure secure communications, Alice needs something to share with Bob to verify that she is in fact Alice. She can prove it by using a *certificate* that is digitally signed by a third party—a Certificate Authority (CA) server. To prove her identity, Alice must have a certificate signed by a CA server that both Alice and Bob trust. In the subsequent sections, the descriptions of digital signatures and certificates address this topic.

## Digital Signatures

The objective of the digital signature is to provide two benefits:

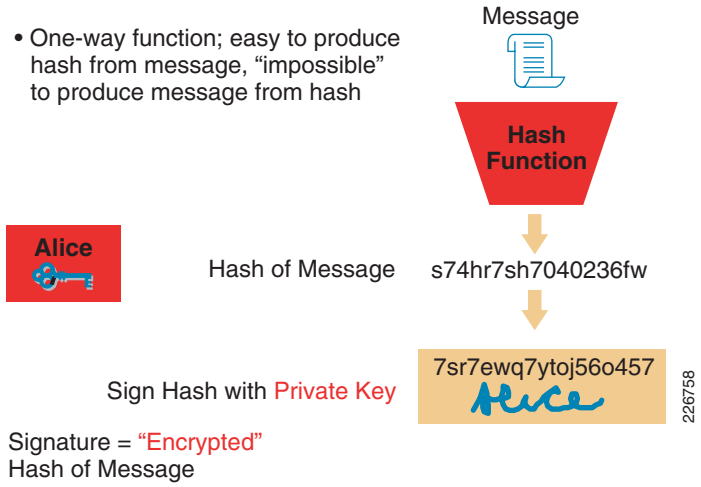
- Individual identity.
- Non-repudiation, so that a sender cannot deny having initiated a communication. This is very important for e-commerce applications.

Digital signatures are basically built in a two step process.

1. Take the message (which is normally the X.509 certificate) and generate the hash of the message.
2. Encrypt the message using the sender's private key.

Figure 1 depicts the process of building a digital signature.

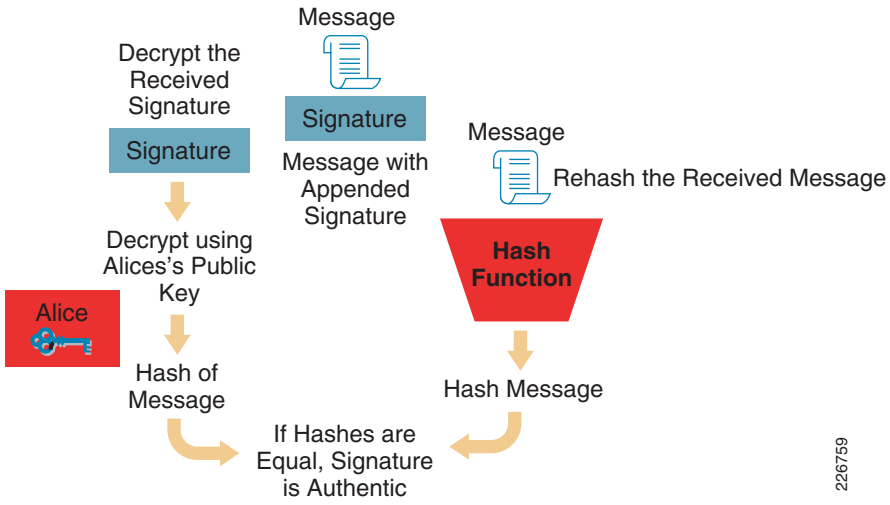
***Figure 1 Building Digital Signatures***




## Digital Signature Verification

Verification of the digital signature happens in the reverse process from which it was built. Figure 2 illustrates how the digital signature is verified.

*Figure 2 Digital Signature Verification Process*

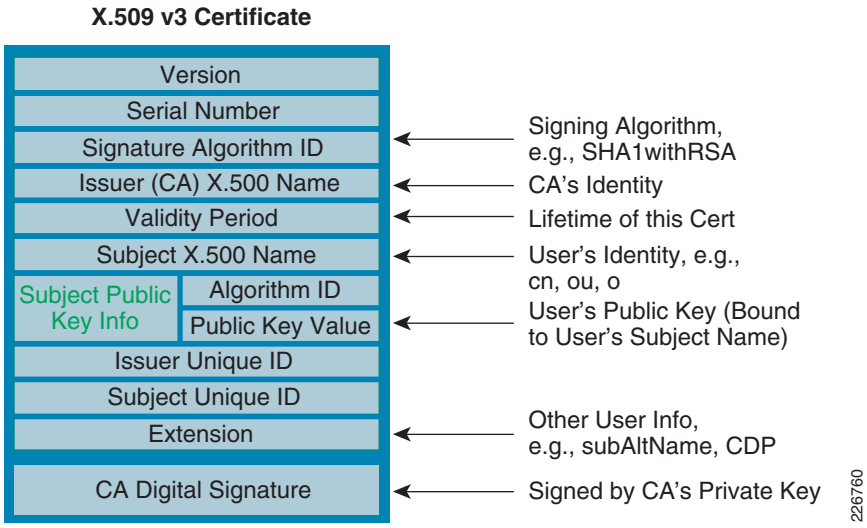



## What is a Certificate?

A *certificate* is basically composed of two elements:

- The user credentials—such as the hostname, common name (CN), serial number, and public key—and other elements as shown in Figure 3. The most important element of the certificate is the public key because all crypto-algorithms depend upon the public key.
- The digital signature of a CA server. The digital signature is in effect a certification from a trusted public authority that is trusted by both parties—it certifies that information provided in the certificate can be trusted.

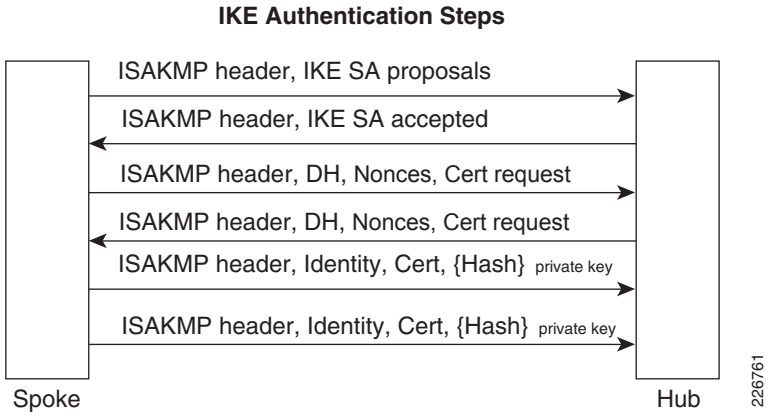
*Figure 3 Certificate Details*



## How Does IKE use Certificates?

Most of the VPN technologies, such as IPSec, DMVPN, GETVPN, and Easy VPN, use the Internet Key Exchange (IKE) protocol to derive session keys. Figure 4 illustrates how IKE uses certificates for authentication:

***Figure 4 How IKE Uses Certificates for Authentication***

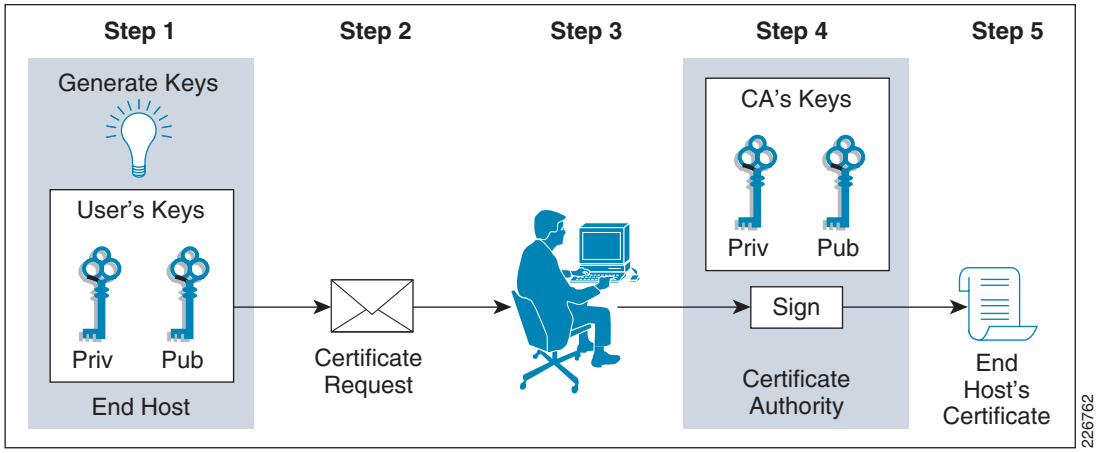



Certificates come into play in the fifth and sixth protocol steps shown at the bottom of Figure 4. If pre-shared key authentication was in use, the spoke-and-hub would have followed the similar steps—except for the fifth and sixth steps steps in which they would have computed hashes to authenticate each other. The hash of the sender is basically computing the hash of all the previous exchanges, IP address, pre-shared key, and other elements. When using certificates, the same hash is encrypted with each peer's private key—in addition to what is done in pre-shared keys. Both peers can decrypt the hash because they know the public key of the corresponding peer. How do they know the public key? This is provided via certificates. How do we trust the certificate? There is a digital signature in the certificate that is signed by a trusted authority.

## How do we Get a Certificate

Figure 5 illustrates the process of obtaining a certificate. The user (a router in this case) generates the public/private key pair and gets the appropriate public key signed by trusted third party. There are several methods for this enrollment process and these are addressed in the"PKI" section on page 16.

*Figure 5 Obtaining a Digital Certificate*




In general, the stages for obtaining a key as illustrated in Figure 5 is as follows:

- Each client generates a private/public key pair. With Cisco IOS, this is done using the command **crypto key generate rsa**.
- Each client would send its public key and other credentials (such as IP address, serial number, hostname, and so on) to a trusted third party.
- The trusted third party digitally signs the client's credentials and returns it
- The client presents this certificate during the authentication process with other peers

# Deployment Scenarios

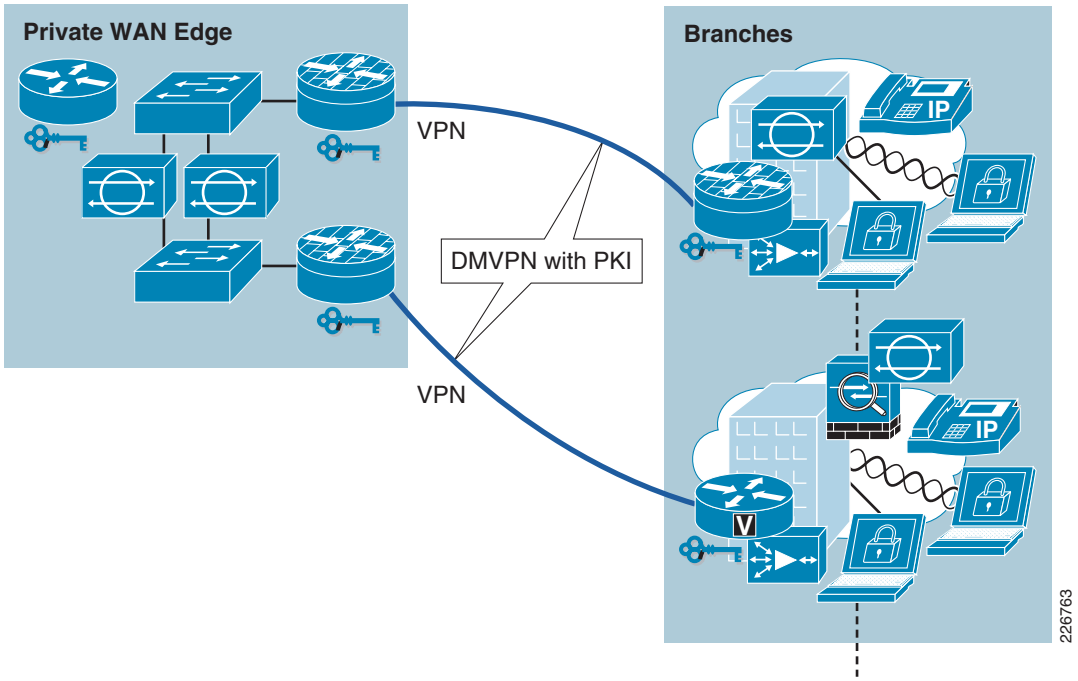
PKI can be used in several scenarios. The following sections describe some of the most common deployment scenarios:


- Branch/WAN Deployment, page 7
- Internet Edge Deployment, page 8
- Remote Access Deployment, page 9


## Branch/WAN Deployment

PKI can be used to provide better security for VPN connections such as GETVPN, DMVPN, IPSec/Generic Routing Encapsulation (GRE). Figure 6 depicts how PKI can be used in a branch/WAN deployment. A key design consideration is locating the CA server. Because all the branches must enroll with the CA server, it should be directly accessible on the WAN edge, as should the VPN gateway. The CA server should also be hardened as per the *Cisco SAFE Reference Guide*.

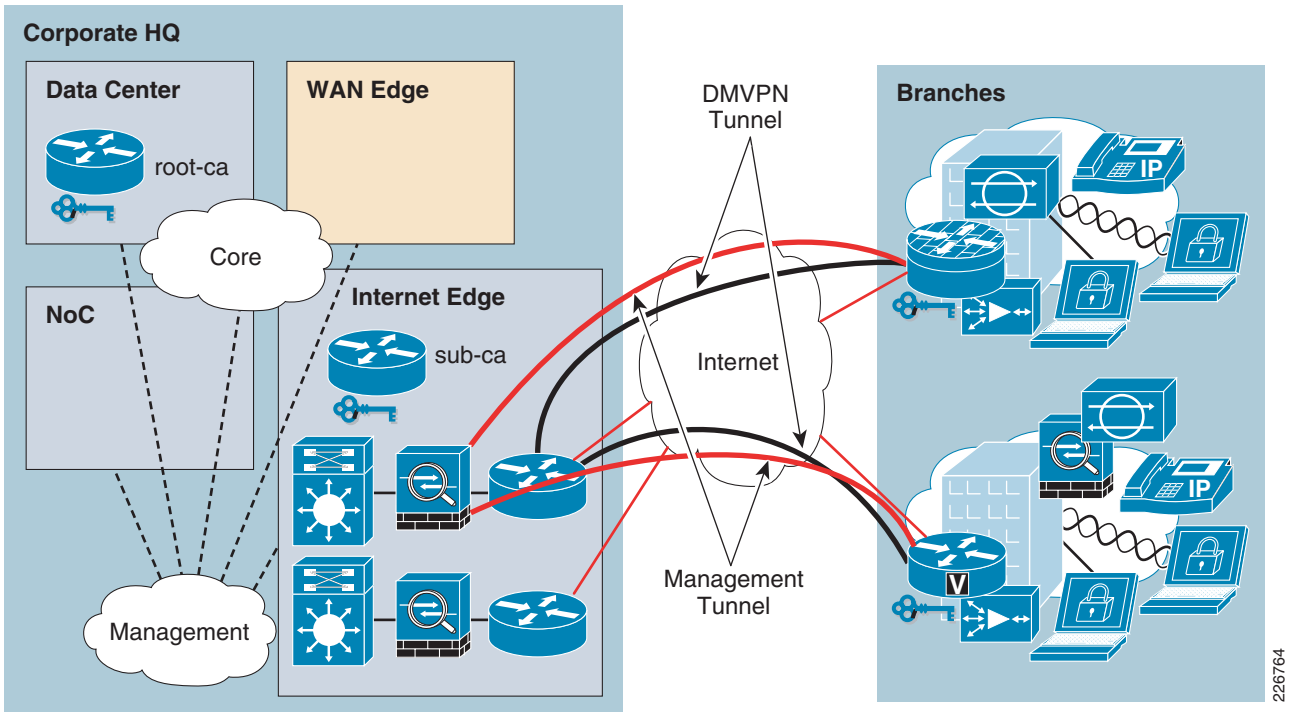
*Figure 6 PKI Used in a Branch/WAN Deployment*




## Internet Edge Deployment

The CA server can also reside in the Internet edge. Such deployments map to situations in which customers only use the Internet connection as the WAN interface. In that scenario, the recommendation is to have an out-of-band management tunnel using Easy VPN to the Internet edge and to then use that management tunnel to enroll certificates. See Figure 7.

*Figure 7 PKI Deployed in Internet Edge*

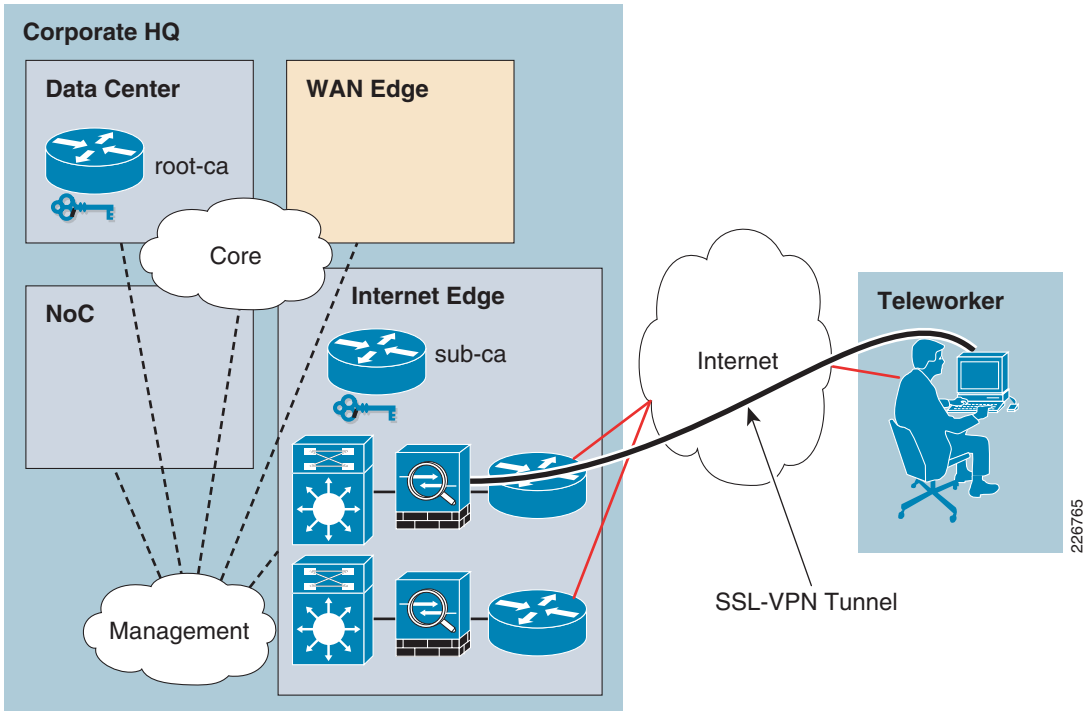



## Remote Access Deployment

PKI can be used to authenticate Easy VPN or SSL VPN sessions. For Easy VPN, the Cisco VPN client supports digital certificates; both the Easy VPN client and the Easy VPN server can use certificates to authenticate the sessions—rather then use group-based, pre-shared keys. The biggest advantage in using PKI for an Easy VPN sessions is that when an employee leaves the company any associated certificate can be revoked—thereby preventing continued resource access using a group-based, pre-shared key.

For SSL VPN sessions, both the SSL VPN gateway and the clients can use certificates to authenticate each other (in similar fashion as with Easy VPN). It is common for only the server to provide the certificate—not the client. However, having certificates at both client and server enhances the strength of authentication. See Figure 8.

*Figure 8 PKI Deployed for Remote Access*




# PKI Design Components

The key components of PKI described in the subsequent sections are as follows:

- CA Server Models, page 10
- When to Use Multiple CAs, page 11
- Enrollment of Certificates, page 12
- Maintaining a Database of Certificates, page 12
- Rollover of Certificates, page 12
- Renewal of Certificates, page 12
- Revocation of Certificates, page 13

## CA Server Models

There are different models for how CA servers can be organized. The following lists the most common models.

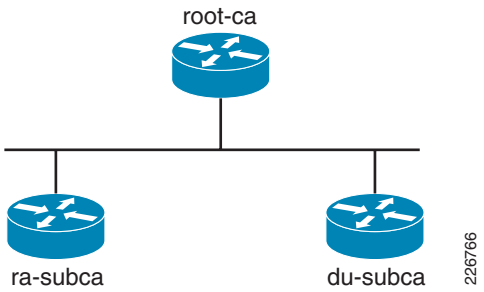
- *Standalone CA server*—This model is the simplest method of deployment. The CA server is the root server and directly communicates with the clients.

  **Note**—Even though this model is simple to deploy and manage, it has the following limitations:

  - Performance is affected because the entire PKI is dependent on this particular server.
  - If the root CA gets compromised, then all the keys are affected.

- *Registration Authority (RA) mode*—A Cisco IOS certificate server can be configured to run in RA mode. In this environment, the root CA offloads authentication and authorization responsibilities to an RA. When the RA receives a Simple Certificate Enrollment Protocol (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. 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. However, there are some limitations with this model. The RA model would create a hierarchical environment, but the root CA is still involved in answering every request. As a result, the performance of the PKI is still dependent on a single root CA. For hierarchical model, the subordinate CA (subordinate CA) model is recommended. This is defined in the model description that follows for multiple CAs.
- *Multiple CAs*—A 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 Rivest, Shamir, and Adelman (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. Multiple tiers of CAs are configured by either the root CA or with another subordinate CA. Within a hierarchical PKI, enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Figure 9 shows how a hierarchical CA can be designed.

***Figure 9 Hierarchical Architecture***




The root CA (*root-ca*) holds a self-signed certificate and the subordinate CAs obtain their certificates from the root CA. Once the subordinate CAs are operational, all requests come to the subordinate CA and root CA can be offline.

# When to Use Multiple CAs

Multiple CAs provide users with added flexibility and reliability. For example, subordinate CAs can be placed at the WAN edge, whereas the root CA is located at the data center. This way the root CA remains secure and well protected because the clients cannot (and don't need to) establish direct communication with it. 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 environment is recommended are as follows:

- Large and very active networks in which a large number of certificates are revoked and reissued. A multi-tier CA helps to control the size of the certificate revocation lists (CRL).
- When online enrollment protocols are used, the root CA can be kept offline except to issue subordinate CA certificates. This scenario provides added security for the root CA.

# Enrollment of Certificates

To obtain a certificate, clients must enroll with the CA server. This enrollment can happen in several ways. The following list summarizes the current enrollment methods:

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

**Note** SCEP is used for all enrollments in this publication.

- *PKCS12*—The router imports certificates in PKCS12 format from an external server.
- *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 the CA.

Among these methods, enrollment using SCEP is easiest to deploy and use.

# Maintaining a Database of Certificates

Keeping a repository of certificates is an important part of the infrastructure. This is useful for following reasons:

- Keeping a central database tracks the clients that have received certificates.
- If a certificate must be revoked, it will be easier to revoke with the serial number of the certificate.
- By default, the CA saves certificates locally within the router, but it is recommended to archive the certificate outside the CA server.

# Rollover of Certificates

This concept is mainly applicable to servers—not clients. The certificates come with a specific expiration time and expire after the specific period has elapsed. Once the certificate on the root CA or subordinate CA expires, then the server will stop operation and will not be able to issue new certificates. To prevent the above scenario, both subordinate CA and IOS CA servers support roll over to the new certificate before the current certificate expires. A more detailed description about this feature is found in the "PKI" section on page 16.

**Note** Auto-roll over is not enabled by default. This feature must be explicitly configured.

# Renewal of Certificates

This concept is mainly applicable to clients. The certificates come with a specific expiration time and expire after the specified period has elapsed. Once the certificate on the client expires, the client cannot present the expired certificate as its identity. The only way to obtain a certificate once one has expired is to request a new one. To prevent that from happening, spokes can request renewal for an existing certificate before it expires.

## Revocation of Certificates

One of the advantages of PKI is its ability to revoke certificates. The CA server can revoke a certificate using its serial number. As a result, the VPN gateway which checks the CRL list periodically would reject the tunnel if it finds the serial number of the certificate is found in the CRL list. A more detailed description about how this can be used in DMVPN scenarios provided in the PKI deployment section that follows.

# PKI Deployment Basics

The architecture described in this publication provides detailed steps for setting up a scalable PKI for enterprise IPSec VPN deployments. This design document provides guidelines for migrating from existing pre-shared key-based environments to PKI-based certificates. It also addresses a hybrid model involving both pre-shared keys and PKI certificates.

In the current phase of this architecture, DMVPN is used as the underlying encryption system, but in latter phases the GETVPN will be also tested. This design document complements the *Digital Certificates/PKI for IPSec VPNs*, which addresses PKI and digital certificates for IPsec VPNs. The *Digital Certificates/PKI for IPSec VPNs* explains how to set up a basic PKI for VPN services.

This publication enhances the *Digital Certificates/PKI for IPSec VPNs* by adding the following components:

- Setting up the subordinate CA architecture.
- Renewing the certificate for the root and subordinate CAs when the certificate expires.
- Performing the recovery for the root and subordinate CAs during outages.
- Explaining how DMVPN clients can use the PKI for authentication.

## Prerequisites

This section summarizes the following basic considerations before you start your implementation activity:

- What You Need to Know Before You Begin
- IP Addressing Requirements
- Scaling Information

### What You Need to Know Before You Begin

Refer to the *Digital Certificates/PKI for IPSec VPNs*. It describes setting up the basic components of PKI. It is expected that anyone using this publication to implement PKI have a fundamental understanding of the root CA and the following components:

- CRL checking
- Various types of files in CA server

## IP Addressing Requirements

Subordinate CA servers should be reachable via the WAN. The root CA server should be located privately and not be reachable via WAN—this way it is protected. In the design presented in this publication, the recommended storage location for certificates is external. Hence a FTP server and a HTTP server should be deployed.

The FTP and HTTP servers are located in the data center. In this design, only the subordinate CAs and root CA can access the FTP and HTTP servers.

All the branch routers using certificates must have clocks synchronized. It is recommended to have the Network Time Protocol (NTP) server at the main office and to provide access to it from all the branch routers.

## Scaling Information

In this design, the main scalability burden is on the WAN aggregation routers, which use PKI to perform authentication. The WAN aggregation architecture should be designed such that there are enough routers for high scalability. In previous testing, it was determined that a Cisco 7200 NPE-G2 with a Cisco VPN Services Adapter (VSA) can support up to 600 DMVPN tunnels. With PKI authentication and certificate revocation list (CRL) checking, the Cisco 7200 router should support a similar number of tunnels; the only difference would be that the tunnels would be slower to initialize when compared to pre-shared key authentication tunnels. Based on the number of branches, a sufficient number of WAN aggregation routers should be planned.

Low values for lifetime of certificates were chosen to illustrate relevant examples of auto-enrollment and roll over. In the "Best Practices for Deploying PKI for a Large-Scale IPSec Solution" section on page 27, configuration of the subordinate CAs and root CA are included with recommended lifetimes.

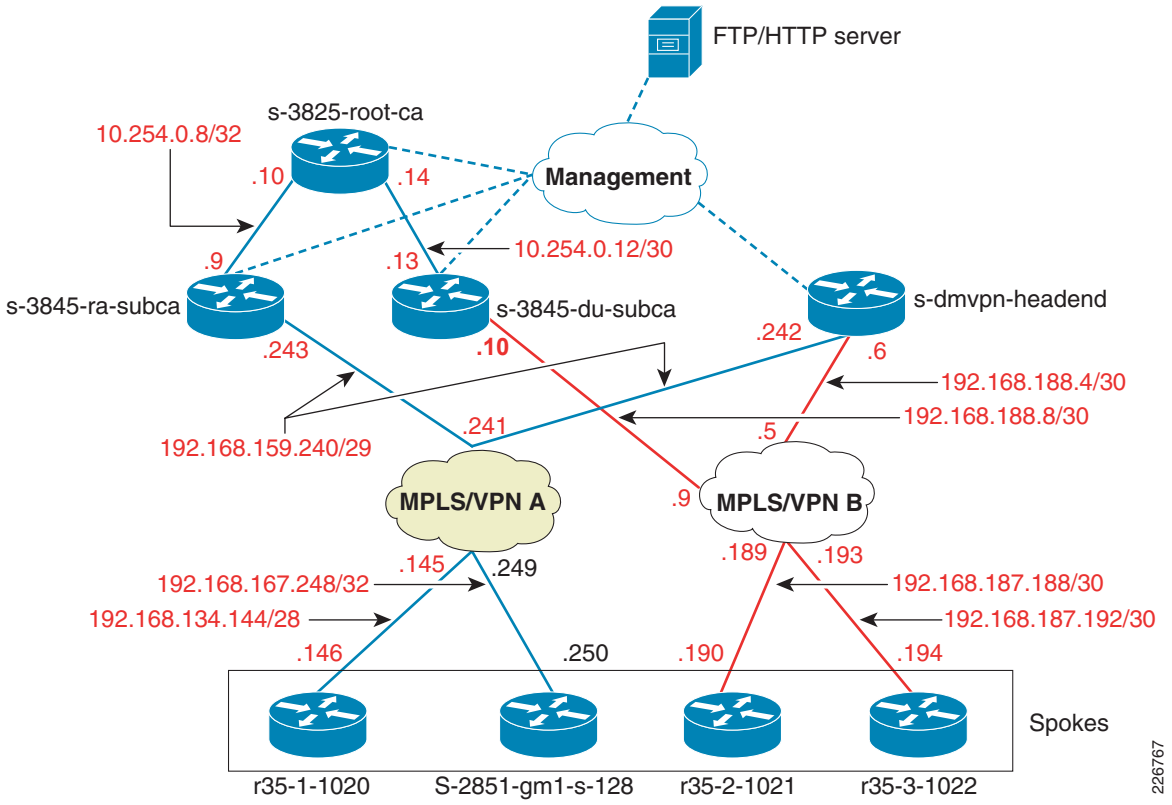
# Sample PKI Architecture

The following diagram shows the basic architecture for deploying a PKI-based infrastructure. This architecture has following characteristics:

- Regional subordinate CAs
- Subordinate CA certificates are signed by the root CA
- Provides layer of separation between root CA and clients
- Multiple subordinate CAs provide higher scalability

Figure 10 illustrates the topology of the architecture presented in this publication.

*Figure 10 Topology Diagram*




# Requirements

The architecture design presented in this publication, features the following products and software elements:

- Cisco 7200 VXR VPN aggregation router
- Cisco 2811 and Cisco 3845 branch routers
- Cisco 3845/3825 as root CA
- Cisco 3845/3825 as subordinate CAs
- Microsoft Windows 2003-based server as FTP/HTTP server

# PKI Implementation Guidelines

PKI implementation is done in two parts. The first part is setting up PKI and the second part is actually using the infrastructure for deploying the solution. As mentioned in the "Deployment Scenarios" section on page 7, PKI can be used in a variety of ways, but in this publication a PKI-based design with a branch/WAN arrangement is validated. The second part of the implementation is actually using the infrastructure for setting up DMVPN sessions.

# PKI

The PKI ca be done in several ways. In this publication, we recommend a hierarchical design with a root CA and subordinate CAs. The subsequent section address setting up this model.

Specific CLI-based configuration procedures described include the following:


- Setting up Root CA Server, page 16
- Setting up the Subordinate CA Architecture, page 17
- Certificate Renewal, page 20
- Recovery of PKI Certificate During Outages, page 22


## Setting up Root CA Server

Follow these steps to setup the root CA server.

**Step 1** Enable the NTP server.

```
ntp server xxx.26.129.252
```

**Step 2** Verify that NTP server is synchronized.

```
S-3825-root-ca# show ntp associations
      address         ref clock     st  when  poll reach  delay  offset    disp
*~xxx.26.129.252   10.81.254.202     2    28    64  377     0.6    0.01     0.0
 * master (synced), # master (unsynced), + selected, - candidate, ~ configured

S-3825-root-ca# show clock
06:43:43.259 EST Thu Nov 13 2008
S-3825-root-ca#
```

**Step 3** Enable the HTTP server.

```
S-3825-root-ca# conf t
Enter configuration commands, one per line. End with CNTL/Z.
S-3825-root-ca(config)# ip http server
S-3825-root-ca(config)# end
```

**Step 4** Generate the RSA keys.

```
S-3825-root-ca(config)# crypto key generate rsa label root-ca
```

**Step 5** Configure the certificate server. It should have the same name as the RSA key pair.

```
S-3825-root-ca(config)# crypto pki server root-ca
```

**Step 6** Specify the database level. In this example, all the contents of the certificate will be archived.

```
S-3825-root-ca(cs-server)# database level complete
```

**Step 7** Specify the archive password. This will ensure that the certificate and keys will be encrypted with password.

```
S-3825-root-ca(cs-server)# database archive pkcs12 password 7 104D000A061843595F
```

**Step 8** Granting mode. The grant mode *auto* makes the CA server issue certificates automatically.

```
S-3825-root-ca(cs-server)# grant auto
```

**Step 9** Specify the storage location.

```
S-3825-root-ca(cs-server)# database url ftp://xxx.26.185.99
```

The storage location can be NVRAM as well, but the external server is recommended.

**Step 10** Specify the lifetime for the self signed and issued certificates

```
S-3825-root-ca(cs-server)# lifetime certificate 730
S-3825-root-ca(cs-server)# lifetime ca-certificate 1825
```

**Step 11** Configure auto-rollover. This will enable the root CA to roll over to a new certificate when the current certificate expires.

```
S-3825-root-ca(cs-server)# auto-rollover 90
```

**Step 12** Enable the root CA to push this rollover certificate to the subordinate CA when its certificate is about to expire.

```
S-3825-root-ca(cs-server)# grant auto rollover ca-cert
```

**Step 13** Specify the CRL location. This will point the location where the CA server looks to verify the CRL.

```
S-3825-root-ca(cs-server)# cdp-url http://xxx.26.185.99/root-ca.crl
```

**Step 14** For self-signage, create the local trustpoint.

```
S-3825-root-ca(config)# crypto pki trustpoint root-ca
S-3825-root-ca(cs-server)# revocation-check crl none
S-3825-root-ca(cs-server)# rsakeypair root-ca
```

**Step 15** Apply **no shutdown** command to the CA server.

```
S-3825-root-ca(config)# crypto pki server root-ca
S-3825-root-ca(cs-server)# no shut
```

**Step 16** Verify that CA server is active and enabled with the following **show** command.

```
S-3825-root-ca# show crypto pki server
Certificate Server root-ca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=root-ca
    CA cert fingerprint: ABD85DC7 C152AE90 4949A459 B91F0A39
    Granting mode is: auto
    Last certificate issued serial number (hex): 9
    CA certificate expiration timer: 17:34:05 EST Nov 14 2013
    CRL NextUpdate timer: 21:58:11 EST Feb 12 2009
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
S-3825-root-ca#
```

## Setting up the Subordinate CA Architecture

This section describes how to set up subordinate CA

**Step 1** Set up the subordinate CA.

```
S-3825-du-subca# conf t
```

Enter configuration commands, one per line. End with Cntl-Z.

```
S-3825-du-subca(config)# crypto pki server du-subca
S-3825-du-subca(cs-server)# database level complete
S-3825-du-subca(cs-server)# database archive pkcs12 password 7 13061E010803557878
S-3825-du-subca(cs-server)# database url [ftp://xxx.26.185.99]
S-3825-du-subca(cs-server)# grant auto rollover ca-cert
S-3825-du-subca(cs-server)# grant auto
S-3825-du-subca(cs-server)# lifetime crl 0 5
S-3825-du-subca(cs-server)# lifetime certificate 0 0 30
S-3825-du-subca(cs-server)# cdp-url [http://xxx.26.185.99/du-subca.crl]
S-3825-du-subca(cs-server)# mode sub-cs
S-3825-du-subca(cs-server)# auto-rollover
S-3825-du-subca(cs-server)#
```

**Step 2** Set up the subordinate CA trust point.

```
S-3825-du-subca(cs-server)# crypto pki trustpoint du-subca
S-3825-du-subca(ca-trustpoint)# enrollment url [http://10.254.0.14:80]
S-3825-du-subca(ca-trustpoint)# revocation-check crl none
S-3825-du-subca(ca-trustpoint)# rsakeypair du-subca
S-3825-du-subca(ca-trustpoint)# exit
S-3825-du-subca(config)# end
```

**Step 3** Verifying the communication to the FTP server.

```
S-3825-du-subca# ping xxx.26.185.99
```

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to xxx.26.185.99, timeout is 2 seconds:
.!!!!
```

**Step 4** Apply the no shutdown command to the subordinate CA server

This will make the subordinate CA enroll with the root CA. To enroll with the root CA, the subordinate CA must accept the root CA certificate. As soon as no shutdown is applied the subordinate CA will start the enrollment process.

```
S-3825-du-subca(config)# crypto pki server du-subca
S-3825-du-subca(cs-server)# no shut
% Some server settings cannot be changed after CA certificate generation.
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Writing du-subca.ser !
The certificate has the following attributes:
Fingerprint MD5: 7F626D1E 07C1C3AC 30220222 25F76AE2
Fingerprint SHA1: 8CFD5BB1 60ECBF8B 10BB4188 9CB11BDC E8829B6E
```

**Step 5** Accept the root CA certificate.

```
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.%
% 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: xxxxxxx
Re-enter password: xxxxxxxx

% Certificate request sent to Certificate Authority

% Enrollment in progress...
```

**Step 6** Go to the root CA and manually grant the subordinate CA enrollment request.

```
S-3825-root-ca# crypto pki server root-ca grant all
Writing 1D0.crt !
Writing 1D0.cnm !
Writing root-ca.ser !
S-3825-root-ca#
```

**Step 7** Go to the subordinate CA and verify that it receives the certificate.

```
S-3825-du-subca#
Writing du-subca.crl !
% Exporting Certificate Server signing certificate and keys...

Writing du-subca_00002.p12 !

storing the serial number to ftp server
Loading du-subca.ser
[OK - 32/4096 bytes]
storing the crl file to the ftp server
Loading du-subca.crl
[OK - 218/4096 bytes]

S-3825-du-subca# show crypto pki server
Certificate Server du-subca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=du-subca
    CA cert fingerprint: 42A3E048 6CFE2607 2D6E47B8 83A14556
    Server configured in subordinate server mode
    Upper CA cert fingerprint: 7F626D1E 07C1C3AC 30220222 25F76AE2
    Granting mode is: auto
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 16:03:07 EST May 3 2008
    CRL NextUpdate timer: 15:04:17 EST May 3 2008
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 30 days
```

**Step 8** Verify that certificates are on the subordinate CA.

```
S-3825-du-subca# show crypto pki certificates
Certificate (subordinate CA certificate)
  Status: Available
  Certificate Serial Number: 0x1D0
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=du-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 14:58:59 EST May 3 2008
    end   date: 16:03:07 EST May 3 2008
  Associated Trustpoints: du-subca

CA Certificate
  Status: Available
  Certificate Serial Number: 0x1CE
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=root-ca
```

```
Validity Date:
  start date: 12:03:07 EST May 3 2008
  end   date: 16:03:07 EST May 3 2008
Associated Trustpoints: du-subca
```

## Certificate Renewal

This section presents examples that describe how the root CA and subordinate CA handle the renewal of certificates.

**Note** The time specifications shown in the following examples are for the illustration purposes. The actual values are presented in the "Best Practices for Deploying PKI for a Large-Scale IPSec Solution" section on page 27.

### Root CA Certificate Renewal

As mentioned in the "PKI Design Components" section on page 10, the root CA can perform its certificate renewal before its certificate expires. To prevent outages, the root CA creates a shadow certificate called the *rollover certificate* before the current certificate expires. This shadow certificate becomes the active certificate after the current certificate expires. As shown in the following configuration example, the auto-rollover parameter is configured for one hour. One hour before the expiration of the current certificate, the root CA creates a rollover certificate. The clients enrolling with the root after the rollover certificate is generated have both the rollover certificate and the current certificate. Once the current certificate expires, the clients and the root CA server make the rollover certificate the primary certificate.

The following procedure illustrates the process of configuring auto-rollover for root CA.

**Step 1** Enable the root CA to roll over to its new certificate when its current certificate expires.

```
crypto pki server root-ca
auto-rollover 0 1
! The roll over is created one hour before the current certificate expires
```

**Step 2** Verify the clock.

```
S-3825-root-ca# show clock
10:19:36.955 EST Tue May 27 2008 ! The clock right now is 10:19
S-3825-root-ca#
```

**Step 3** Verify that the root CA generates its shadow certificate when its current certificate expires.

```
S-3825-root-ca# show crypto pki certificates
CA Certificate (Rollover)
  Status: Available
  Certificate Serial Number: 0x0A0
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    Name: root-ca
    cn=root-ca
  Validity Date:
    start date: 10:34:12 EST May 27 2008
```

**Note** The rollover certificate is created and will be sent to the clients along with the current certificate.

```
    end   date: 14:34:12 EST May 27 2008
  Associated Trustpoints: root-ca

CA Certificate
  Status: Available
  Certificate Serial Number: 0x09F
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=root-ca
  Validity Date:
    start date: 06:34:12 EST May 27 2008
    end   date: 10:34:12 EST May 27 2008
```

**Note** The current certificate is about to expire in 15 minutes.

```
  Associated Trustpoints: root-ca
```

The roll-over certificate changes to the current certificate once the current certificate expires.

### Subordinate CA Certificate Renewal

The following steps describe the process of creating a rollover certificate. When the subordinate CA certificate expires, it will roll over to the new certificate.

**Step 1** Enable the root CA to roll over to the new certificate when its current certificate expires.

```
crypto pki server du-subca
auto-rollover 0 0 30
```

**Step 2** Verify clock setting.

```
S-3825-du-subca# show clock
10:27:10.713 EST Thu May 15 2008
S-3825-du-subca#
```

**Step 3** Verify that the subordinate CA generates a shadow certificate in its database before the current certificate expires.

```
S-3825-du-subca# show crypto pki certificates
Certificate (subordinate CA certificate, Rollover)
  Status: Available
  Certificate Serial Number: 0x21
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=du-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 11:59:41 EST May 15 2008
```

```
    end   date: 13:34:12 EST May 15 2008
  Associated Trustpoints: du-subca

Certificate (subordinate CA certificate)
  Status: Available
  Certificate Serial Number: 0x20
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=du-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 09:59:41 EST May 15 2008
    end   date: 11:59:41 EST May 15 2008
  Associated Trustpoints: du-subca

CA Certificate
  Status: Available
  Certificate Serial Number: 0x1D
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=root-ca
  Validity Date:
    start date: 09:34:12 EST May 15 2008
    end   date: 13:34:12 EST May 15 2008
  Associated Trustpoints: du-subca
```

**Step 4** Enable the rollover functionality of the subordinate CA server via **auto-rollover** command.

**Step 5** Verify that the subordinate CA creates a shadow certificate before the current certificate expires.

**Step 6** Verify that the subordinate CA re-enrolls with the shadow certificate on the root CA before the current certificate expires.

**Step 7** Verify that the subordinate CA rolls over to the new certificate, once the root CA authorizes its current request.

### Spoke Certificate Renewal

The auto-enroll can be enabled on the spoke with the **auto-enroll 80 regenerate** command. The *regenerate* key would create a new RSA key pair when the spoke renews its certificate. Detailed examples are provided in the DMVPN deployment section that address spoke certificate enrollment (which also includes renewal).

Verify that the spoke enrolls with the subordinate CA server before 80 percent t of its current expiration time has elapsed.

## Recovery of PKI Certificate During Outages

The two procedures described in the following sections illustrate the disaster recovery scenario and how to recover the root CA or subordinate CA when they completely fail and must be replaced with a new router.

## Recovery of Root CA

The following procedure describes how to recover when the root CA fails and you are trying to recover from the scratch.

**Step 1** Remove the root CA configuration.

```
S-3825-root-ca(config)# no crypto pki server root-ca
```

**Step 2** Remove the certificate chain.

```
S-3825-root-ca(config)# no crypto pki certificate chain root-ca
This will remove all certificates for trustpoint root-ca
Are you sure you want to do this? \[yes/no\]: yes
S-3825-root-ca(config)# end
```

**Step 3** Import the latest pkcs12 from the FTP server.

```
S-3825-root-ca(config)# crypto pki import root-ca pkcs12 [ftp://xxx.26.185.99] ?
Passphrase used to protect the pkcs12 file

S-3825-root-ca(config)# crypto pki import root-ca pkcs12 [ftp://xxx.26.185.99] cisco123
% Importing pkcs12...
Address or name of remote host \[xxx.26.185.99\]?
Source filename \[root-ca\]? root-ca_00237.p12
Reading file from [ftp://xxx.26.185.99/root-ca_00237.p12]
Loading root-ca_00237.p12
\[OK - 1515/4096 bytes\]

CRYPTO_PKI: Imported PKCS12 file successfully.
S-3825-root-ca(config)# end
```

**Step 4** Configure the root CA configuration.

```
S-3825-root-ca# conf t
Enter configuration commands, one per line. End with CNTL/Z.
S-3825-root-ca(config)# crypto pki server root-ca
S-3825-root-ca(cs-server)# database level complete
S-3825-root-ca(cs-server)# database archive pkcs12 password 7 104D000A061843595F
S-3825-root-ca(cs-server)# grant auto rollover ca-cert
S-3825-root-ca(cs-server)# grant auto
S-3825-root-ca(cs-server)# lifetime crl 0 10
S-3825-root-ca(cs-server)# lifetime certificate 0 2
S-3825-root-ca(cs-server)# lifetime ca-certificate 0 4
% The CA certificate lifetime change will take effect after existing CA certs expire.
S-3825-root-ca(cs-server)# cdp-url [http://xxx.26.185.99/root-ca.crl]
S-3825-root-ca(cs-server)# auto-rollover 0 1
S-3825-root-ca(cs-server)# database url [ftp://xxx.26.185.99]
% Server database url was changed. You need to move the
% existing database to the new location.
S-3825-root-ca(cs-server)#
S-3825-root-ca(cs-server)# crypto pki trustpoint root-ca
S-3825-root-ca(ca-trustpoint)# revocation-check crl none
S-3825-root-ca(ca-trustpoint)# rsakeypair root-ca
S-3825-root-ca(ca-trustpoint)# end
```

**Step 5** Verify that the root CA server is active.

```
S-3825-root-ca# show crypto pki server
Certificate Server root-ca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=root-ca
```

```
    CA cert fingerprint: 1B560375 BF45735E 2FB04670 0310F406
    Granting mode is: auto
    Last certificate issued serial number: 0x1
    CA certificate expiration timer: 14:34:12 EST May 13 2008
    CRL NextUpdate timer: 10:55:45 EST May 13 2008
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 0 days
    Autorollover timer: 13:34:12 EST May 13 2008
```

## Recovering a Subordinate CA

The following procedure describes how to recover when the subordinate CA fails.

The recovery of the certificate follows these high level steps:

- Import the pkcs12 file.
- Restore the configuration.

**Step 1** Verify that the subordinate CA is active.

```
S-3825-du-subca# show crypto pki server
Certificate Server du-subca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=du-subca
    CA cert fingerprint: F25C6E76 18C3DE5C F014EB72 DE8A148B
    Server configured in subordinate server mode
    Upper CA cert fingerprint: CC59C214 DD402575 0BA87324 A809F682
    Granting mode is: auto
    Last certificate issued serial number: 0x6D
    CA certificate expiration timer: 09:34:12 EST May 14 2008
    CRL NextUpdate timer: 09:32:56 EST May 14 2008
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Rollover status: available for rollover
    Rollover CA certificate fingerprint: CEA006FB 8ECDD846 801B8C4B C757210D
    Rollover CA certificate expiration time: 11:18:05 EST May 14 2008
    Auto-Rollover configured, overlap period 30 days
```

**Step 2** Remove the server configuration to simulate an outage.

```
S-3825-du-subca# conf t
Enter configuration commands, one per line. End with CNTL/Z.
S-3825-du-subca(config)# no crypto pki server du-subca
Certificate server 'remove server' event has been queued for processing.
S-3825-du-subca(config)#
% The server file [ftp://xxx.26.185.99/du-subca.ser] could not be deleted.
% You need to delete it manually.
```

**Step 3** Remove the certificate chain.

```
S-3825-du-subca(config)# no crypto pki certificate chain du-subca
This will remove all certificates for trustpoint du-subca
Are you sure you want to do this? \[yes/no\]: yes
S-3825-du-subca(config)#
```

**Step 4** Verify that there are no server configurations.

```
S-3825-du-subca# show crypto pki server
%Cannot find Certificate Server
```

```
S-3825-du-subca#
```

**Step 5** Configure the subordinate CA server.

```
S-3825-du-subca(config)# crypto pki server du-subca
S-3825-du-subca(cs-server)# database level complete
S-3825-du-subca(cs-server)# database archive pkcs12 password 7 13061E010803557878
S-3825-du-subca(cs-server)# grant auto rollover ca-cert
S-3825-du-subca(cs-server)# grant auto
S-3825-du-subca(cs-server)# lifetime crl 0 5
S-3825-du-subca(cs-server)# lifetime certificate 0 0 30
S-3825-du-subca(cs-server)# cdp-url [http://xxx.26.185.99/du-subca.crl]
S-3825-du-subca(cs-server)# mode sub-cs
S-3825-du-subca(cs-server)# auto-rollover
```

**Step 6** Configure the trust point for the subordinate CA.

```
S-3825-du-subca# conf t
Enter configuration commands, one per line. End with CNTL/Z.
S-3825-du-subca(config)# crypto pki trustpoint du-subca
S-3825-du-subca(ca-trustpoint)# revocation-check crl none
S-3825-du-subca(ca-trustpoint)# rsakeypair du-subca
S-3825-du-subca(ca-trustpoint)# end
```

**Note** Don't configure the enrollment URL yet.

**Step 7** Import the PKCS12 certificate.

This certificate should be at FTP server. The name should be something like *du-subca_xxxxx.p12*

```
S-3825-du-subca(config)# crypto pki import du-subca ?
 certificate Import a certificate from a TFTP server or the terminal
Import from PEM files
Import from PKCS12 file

S-3825-du-subca(config)# crypto pki import du-subca pk
S-3825-du-subca(config)# crypto pki import du-subca pkcs12 ?
Import from archive: file system
Import from cns: file system
Import from flash: file system
Import from ftp: file system
Import from http: file system
Import from https: file system
Import from null: file system
Import from nvram: file system
Import from pram: file system
Import from rcp: file system
Import from scp: file system
Import from system: file system
Import from tar: file system
Input pkcs12 file from the terminal
Import from tftp: file system
mport from tmpsys: file system
Import from xmodem: file system
Import from ymodem: file system

S-3825-du-subca(config)# crypto pki import du-subca pkcs12 [ftp://xxx.26.185.99]
LINE Passphrase used to protect the pkcs12 file

S-3825-du-subca(config)# crypto pki import du-subca pkcs12 [ftp://xxx.26.185.99] cisco123
% Importing pkcs12...
Address or name of remote host \[xxx.26.185.99\]?
```

```
Source filename \[du-subca\]? du-subca_00016.p12
Reading file from [ftp://xxx.26.185.99/du-subca_00016.p12]
Loading du-subca_00016.p12
\[OK - 2163/4096 bytes\]

CRYPTO_PKI: Imported PKCS12 file successfully.
S-3825-du-subca(config)# end
S-3825-du-subca#
```

**Step 8** Configure the enrollment URL.

```
S-3825-du-subca(config)# crypto pki trustpoint du-subca
S-3825-du-subca(ca-trustpoint)# enrollment url [http://10.254.0.14:80]
```

**Step 9** Enable the *du-subca* server.

```
S-3825-du-subca(config)# crypto pki server du-subca
S-3825-du-subca(cs-server)# no shut
S-3825-du-subca(cs-server)# end

S-3825-du-subca# show crypto pki server
Certificate Server du-subca:
 Status: enabled
 State: enabled
 Server's configuration is locked (enter "shut" to unlock it)
 Issuer name: CN=du-subca
 CA cert fingerprint: 3C038827 6EBBF0BA DA15C1D3 8071AD39
 Server configured in subordinate server mode
 Upper CA cert fingerprint: 39B754AC 8820E36C 950D4F5C BE9F4629
 Granting mode is: auto
 Last certificate issued serial number: 0x1
 CA certificate expiration timer: 13:18:05 EST May 14 2008
 CRL NextUpdate timer: 10:05:16 EST May 14 2008
 Current primary storage dir: nvram:
 Database Level: Complete - all issued certs written as <serialnum>.cer
 Auto-Rollover configured, overlap period 30 days
 Autorollover timer: 10:03:27 EST May 14 2008
```

**Note** The subordinate CA sends the request to root CA.

```
S-3825-du-subca(cs-server)# end
S-3825-du-subca#

% Certificate request sent to Certificate Authority
```

**Step 10** Go to the root CA and grant the certificate to subordinate CA.

```
ca-console# 2
\[Resuming connection 2 to root-ca ... \]

Writing root-ca.crl
S-3825-root-ca#
S-3825-root-ca# crypto pki server root-ca ?
grant Grant enrollment requests
 info Display info
 password One Time Password for SCEP enrollment
 reject Reject enrollment requests
 remove Remove enrollment requests from database
 request Retrieve an enrollment request
 revoke Revoke certificate

S-3825-root-ca# crypto pki server root-ca info ?
 crl Certificate Revocation List
```

```
 requests Enrollment Requests

S-3825-root-ca# crypto pki server root-ca info requests
Enrollment Request Database:

Subordinate CA certificate requests:
ReqID State Fingerprint SubjectName
\--------------------------------------------------------------\-
4 pending 939481D9AFA039DDB5BF7377ABA167DF cn=du-subca

RA certificate requests:
ReqID State Fingerprint SubjectName
\--------------------------------------------------------------\-

Router certificates requests:
ReqID State Fingerprint SubjectName
\--------------------------------------------------------------\-

S-3825-root-ca# crypto pki server root-ca ?
 grant Grant enrollment requests
 info Display info
 password One Time Password for SCEP enrollment
 reject Reject enrollment requests
 remove Remove enrollment requests from database
 request Retrieve an enrollment request
 revoke Revoke certificate

S-3825-root-ca# crypto pki server root-ca grant ?
 <1-999> Request ID
 all all pending requests

S-3825-root-ca# crypto pki server root-ca grant all
Writing B.crt
Writing B.cnm
Writing root-ca.ser
```

# Best Practices for Deploying PKI for a Large-Scale IPSec Solution

The following are the best practices for deploying PKI for IPSec:

- The first and foremost design consideration is placement of the root and subordinate CAs. The root CA is the most criticial element in the overall PKI. It must be placed securely in the data center. The spokes must contact the subordinated CA over the WAN and the subordinate CAs must be connected to the WAN, so that the spokes can contact them directly for enrollment.

**Note** In this design, only manual enrollment is used for the spokes.

- While considering the policy for granting certificates, it is better for the subordinate CA to not be placed in “grant auto” mode because this causes the subordinate CA to issue certificates automatically to any spoke that might attempt to connect. It is advisable to put it in manual mode. The following configuration would list it.

```
ra-subca(config)# crypto pki server ra-subca
Certificate server 'shut' event has been queued for processing.
ra-subca(cs-server)#no grant auto
ra-subca(cs-server)#no shut
```

Verify that the subordinate CA is in grant mode manual with the **show crypto pki server** command as follows:

```
ra-subca# show crypto pki server
Certificate Server ra-subca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=ra-subca
    CA cert fingerprint: ECE8BE9E 9C5179A5 ABD983A2 6E5F5DE8
    Server configured in subordinate server mode
    Upper CA cert fingerprint: ECE8BE9E 9C5179A5 ABD983A2 6E5F5DE8
    Granting mode is: manual
    Last certificate issued serial number (hex): 8
    CA certificate expiration timer: 12:21:19 EST Jan 28 2011
    CRL NextUpdate timer: 17:15:23 EST Mar 6 2009
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 90 days
    Autorollover timer: 12:21:18 EST Oct 30 2010
ra-subca#
```

- *Redundant subordinate CAs*—From a reliability perspective, it is a good practice to have redundant subordinate CAs as illustrated in Figure 10.
- *SCEP*— SCEP, the protocol used for enrollments, runs over HTTP. Since the default port is 80, it is advisable to use a non-standard port to strengthen the security of the subordinate CAs. The following configuration fragment shows how to implement: a non-standard port for this purpose.

```
ra-subca# show running-config | include ip http
ip http server
ip http port 12345
```

Similarly, you must change the enrollment port on the spokes as follows:

```
crypto pki trustpoint ra
 enrollment url http://192.168.159.243:12345
 serial-number
 ip-address 192.168.159.243
 revocation-check none
 rsakeypair ra
 auto-enroll 80 regenerate
```

# DMVPN Enrollment Using PKI

The following sections use the same underlying topology to validate the PKI component and describe how DMVPN authentication can use PKI for authentication. Specific descriptions provided here address the following topics:


- Hub-and-Spoke Enrollment with the PKI, page 29
- Spoke Enrollment with ra-subca, page 29
- Hub Enrollment with a Single Subordinate CA (ra-subca), page 32
- DMVPN Configuration, page 34
- DMVPN Hub Configuration, page 36
- DMVPN with PKI using Multiple Subordinate CAs, page 36
- Migration of DMVPN from Pre-shared to PKI, page 41
- Revocation Checking, page 48

## Hub-and-Spoke Enrollment with the PKI

There are two deployment models in DMVPN:

- Hub-and-Spoke Deployment
- Spoke-to-Spoke Deployment

**Note** In this design, only the hub-and-spoke deployment is covered in detail.

### Hub-and-Spoke Deployment

Key considerations in hub-and-spoke deployments:

- The hub router should enroll with both the subordinate CAs *ra* and *du*.
- The spoke routers can enroll with either of the subordinate CAs.
- The hub router should be configured with ISAKMP profiles that would match pre-shared keys and certificates.

### Spoke-to-Spoke Deployment

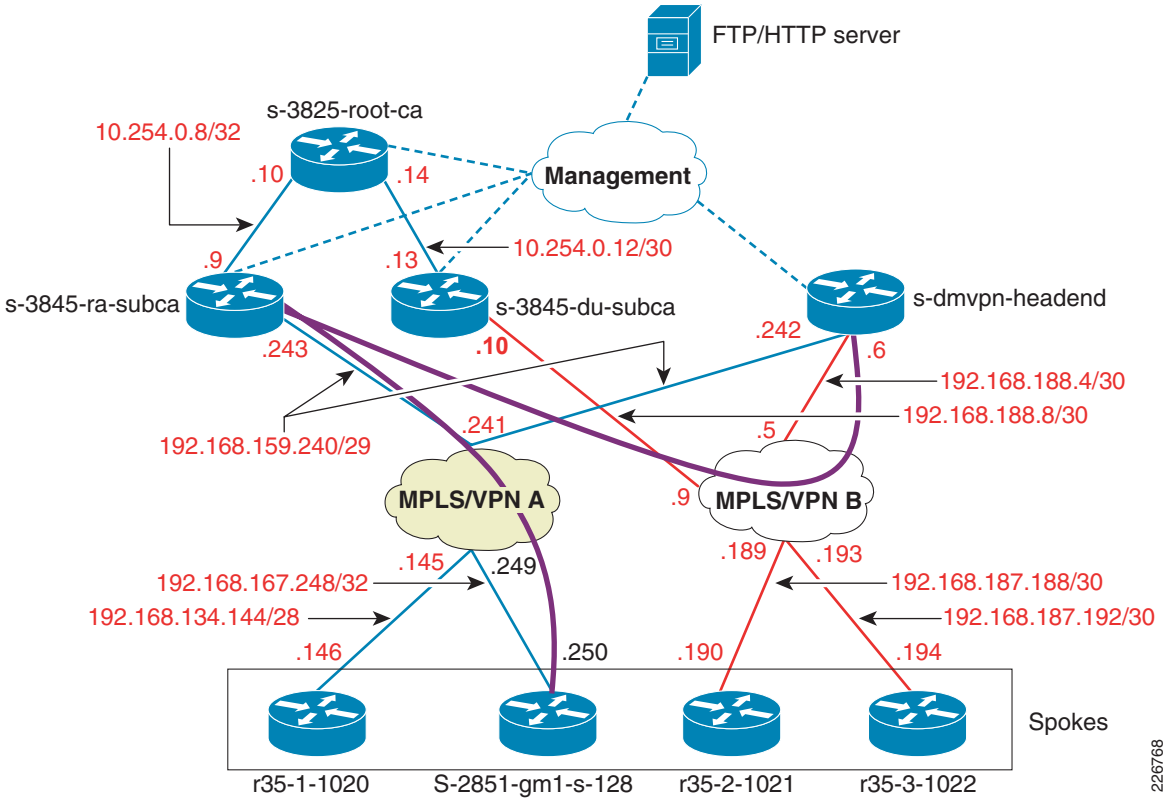
Key considerations in spoke-to-spoke deployments:

- The hub router should enroll with both the subordinate CAs *ra* and *du*.
- The spoke routers can enroll with either of the subordinate CAs.
- Verify that spoke routers can establish tunnels with hub router and that spoke routers use certificates as their identity.

## Spoke Enrollment with *ra-subca*

Figure 11 illustrates the lab topology used to test and validate the design described in this publication; it illustrates how *s-dmvpn-headend* and *S-2851-gm1-s-128* are enrolling with the subordinate CA (*ra-subca*).

*Figure 11 Spoke and Hub Enrollment with ra-subca*




The following procedure describes the process of spoke-and-hub enrollment.

**Step 1** Verify the NTP association. The certificates are dependent on having properly synchronized clocks.

```
S-2851-gm1-s-128# show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
+~xxx.26.156.10   10.81.254.202    2    439   1024   377  0.000   0.905 18.685
*~xxx.26.129.252  10.81.254.202    2    456   1024   377  0.000   0.857 18.673
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
S-2851-gm1-s-128#
S-2851-gm1-s-128# show clock
15:53:40.863 EST Thu Mar 19 2009
```

**Step 2** Generate the RSA key pair.

```
S-2851-gm1-s-128(config)# crypto key generate rsa label spoke-keys
The name for the keys will be: spoke-keys
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, keys will be non-exportable...[OK]
```

**Step 3** Configure enrollment on the spoke.

```
S-2851-gm1-s-128(config)# crypto pki trustpoint ra
S-2851-gm1-s-128(ca-trustpoint)# enrollment url http://192.168.159.243:12345
S-2851-gm1-s-128(ca-trustpoint)# revocation-check none
```

```
S-2851-gm1-s-128(ca-trustpoint)# auto-enroll 70 regenerate
S-2851-gm1-s-128(ca-trustpoint)# rsakeypair spoke-keys
S-2851-gm1-s-128(ca-trustpoint)# exit
```

**Note** Revocation check is disabled for spokes in this design. Only the hub would perform CRL checking.

**Note** The **auto-enroll 70** command specification would cause the spoke to request a new certificate.

**Step 4** Start the enrollment process. This is a two step process:

- Authenticate the subordinate CA certificate.
- Enroll with subordinate CA.

```
S-2851-gm1-s-128(config)# crypto pki authenticate ra
Trustpoint 'ra' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
       Fingerprint MD5: ECE8BE9E 9C5179A5 ABD983A2 6E5F5DE8
      Fingerprint SHA1: 0A86F03E 077E587B 2DB4644A 5BA55F0F FC57D2EF

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
```

The preceding step ensures that the subordinate CA certificate is accepted. This step can be avoided if you know the finger print of the subordinate CA.

**Step 5** Verify the certificate received from the subordinate CA. Key considerations to note include the following:

- The issuer of the subordinate CA (root CA in the example presented)
- The associated trust point (*ra* in the example presented)

```
S-2851-gm1-s-128# show crypto pki certificates
CA Certificate
  Status: Available
  Certificate Serial Number (hex): 08
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=ra-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 11:21:19 EST Jan 28 2009
    end   date: 11:21:19 EST Jan 28 2011
  Associated Trustpoints: ra
```

**Step 6** Because auto-enrollment is configured, the spoke attempts to get the certificate from the subordinate CA.

```
S-2851-gm1-s-128# %
% Start certificate enrollment ..

% The subject name in the certificate will include: S-2851-gm1-s-128.cisco.com
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate ra verbose' commandwill show the fingerprint.
c
Mar 19 16:41:01.747 EST: %PKI-6-CERTRENEWAUTO: Renewing the router certificate for
trustpoint rao
Mar 19 21:41:01.763: CRYPTO_PKI:  Certificate Request Fingerprint MD5: 32C06B4E 09E29D63
B94D051E 749DBB13
```

```
Mar 19 21:41:01.763: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: A757CADE 3BD2BE48
8FD4A4C1 8DF51141 8A15DAC6
```

**Step 7** As mentioned in the best practice description, the subordinate CA is configured in manual mode. The certificate must be manually granted at the subordinate CA.

```
ra-subca# crypto pki server ra-subca grant all
Writing A.crt !
Writing A.cnm !
Writing ra-subca.ser !
ra-subca#
```

**Step 8** Verify the certificate in the spoke. Some of the important fields to look for are the *Certificate Serial Number*, *Subject Name*, and *Validity Date* interval.

```
S-2851-gm1-s-128# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number (hex): 0A
  Certificate Usage: General Purpose
  Issuer:
    cn=ra-subca
  Subject:
    Name: S-2851-gm1-s-128.cisco.com
    hostname=S-2851-gm1-s-128.cisco.com
  CRL Distribution Points:
    http://xxx.26.185.99/ra-subca.crl
  Validity Date:
    start date: 16:43:30 EST Mar 19 2009
    end   date: 16:43:30 EST Sep 15 2009
    renew date: 16:43:30 EST Jul 23 2009
  Associated Trustpoints: ra

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 08
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=ra-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 11:21:19 EST Jan 28 2009
    end   date: 11:21:19 EST Jan 28 2011
  Associated Trustpoints: ra
```

## Hub Enrollment with a Single Subordinate CA (*ra-subca)*

The following procedure describes the process of hub enrollment with subordinate CAs.

**Step 1** As shown in spoke enrollment example, the first step is to verify NTP association and the clock setting.

```
s-dmvpn-headend# show ntp associations

  address         ref clock       st   when   poll reach  delay  offset   disp
*~xxx.26.129.252  10.81.254.202    2    910   1024   377  0.000  -0.692 14.848
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
s-dmvpn-headend# show clo
s-dmvpn-headend# show clock
10:44:25.941 EST
Fri Mar 20 2009
s-dmvpn-headend#
```

**Step 2** Generate the RSA key pair.

```
s-dmvpn-headend(config)# crypto key generate rsa label hub-keys
The name for the keys will be: hub-keys
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, keys will be non-exportable...[OK]

s-dmvpn-headend(config)#
```

**Step 3** Create the enrollment trust point for the hub.

```
crypto pki trustpoint ra
 enrollment url http://192.168.159.243:12345
 revocation-check crl
 rsakeypair hub-keys
 auto-enroll 70 regenerate
```

**Note** On the hub side, it is important to review CRL checking.

**Step 4** Authenticate the subordinate CA certificate.

```
s-dmvpn-headend(config)# crypto pki authenticate ra
Trustpoint 'ra' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: ECE8BE9E 9C5179A5 ABD983A2 6E5F5DE8
      Fingerprint SHA1: 0A86F03E 077E587B 2DB4644A 5BA55F0F FC57D2EF

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
s-dmvpn-headend(config)#
s-dmvpn-headend(config)# %
% Start certificate enrollment ..

% The subject name in the certificate will include: s-dmvpn-headend.cisco.com
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate ra verbose' commandwill show the fingerprint.

Mar 20 14:55:32.777: %PKI-6-CERTRENEWAUTO: Renewing the router certificate for trustpoint
ra
Mar 20 14:55:32.781: CRYPTO_PKI:  Certificate Request Fingerprint MD5: A2FD052B 2DAD2C7E
E853C13A A74C77A5
Mar 20 14:55:32.781: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 6F3863DF 4B97CDDF
1E2F7E60 53092E85 AE9685C2
```

**Step 5** As mentioned with the spoke enrollment process, the certificate must be approved on the subordinate CA because it is in manual mode.

```
ra-subca# crypto pki server ra-subca grant all
Writing C.crt !
Writing C.cnm !
Writing ra-subca.ser !
ra-subca#
```

**Step 6** Verify the certificates on the hub side.

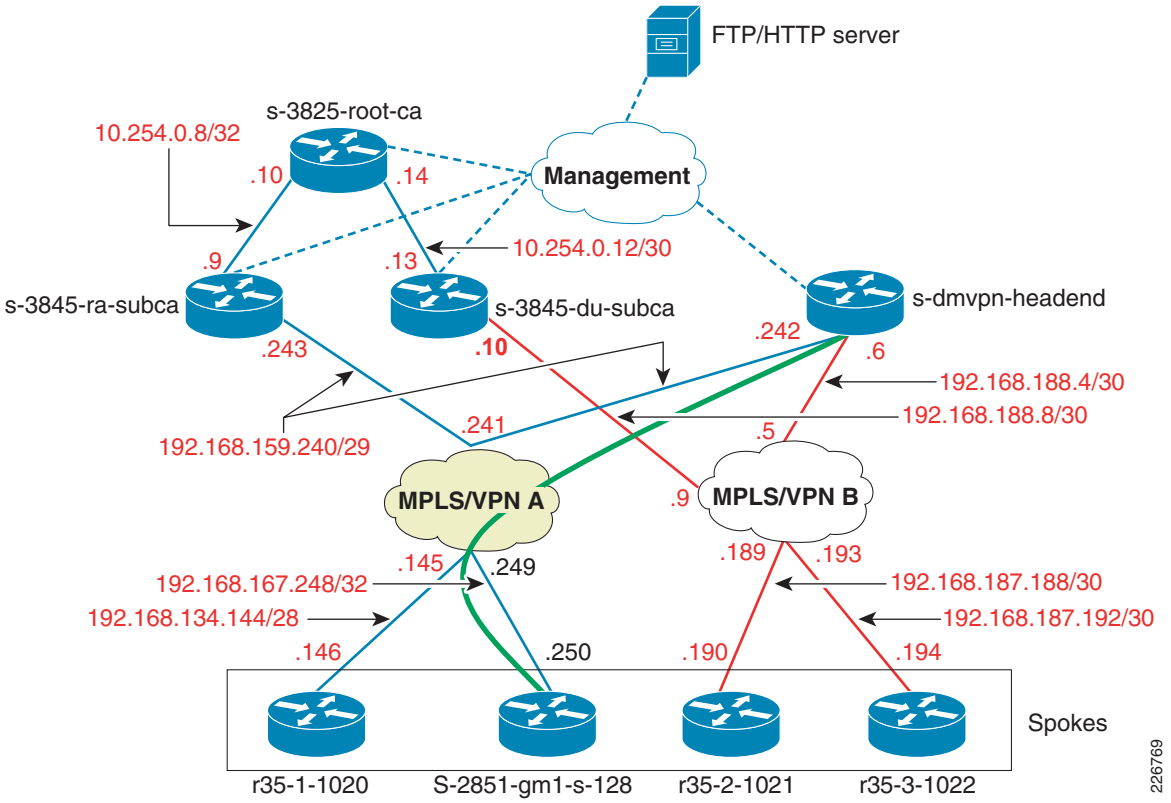
```
s-dmvpn-headend# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number (hex): 0C
  Certificate Usage: General Purpose
  Issuer:
    cn=ra-subca
  Subject:
    Name: s-dmvpn-headend.cisco.com
    hostname=s-dmvpn-headend.cisco.com
  CRL Distribution Points:
    http://xxx.26.185.99/ra-subca.crl
  Validity Date:
    start date: 10:58:07 EST Mar 20 2009
    end   date: 10:58:07 EST Sep 16 2009
    renew date: 10:58:07 EST Jul 24 2009
  Associated Trustpoints: ra

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 08
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=ra-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 11:21:19 EST Jan 28 2009
    end   date: 11:21:19 EST Jan 28 2011
  Associated Trustpoints: ra
```

## DMVPN Configuration

After both hub and spoke are enrolled with the subordinate CA, a DMVPN tunnel can be created to connect them. Figure 12 illustrates a DMVPN tunnel between *S-2851-gm1-s-128* and *s-dmvpn-headend*.

*Figure 12 DMVPN Tunnel Between the Spoke and Hub*




The following is the process for configuring the DMVPN spoke.

**Step 1** Create the **crypto isakmp policy** configuration.

```
crypto isakmp policy 1
 encr 3des
 hash md5
```

**Step 2** Create an **isakmp-profile** configuration which helps in matching identities such as pre-shared keys or certificates. To get more information about the **isakmp-profile** command please refer to the following URL:
http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6635/prod_white_paper0900aecd8034bd59.html

```
S-2851-gm1-s-128(config)# crypto isakmp profile spoke-profile
% A profile is deemed incomplete until it has match identity statements
S-2851-gm1-s-128(conf-isa-prof)# match identity host domain cisco.com
S-2851-gm1-s-128(conf-isa-prof)# exit
```

**Step 3** Create the **transform-set** configuration.

```
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
 mode transport
```

**Step 4** Define the **ipsec profile** configuration.

```
crypto ipsec profile SPOKE_DMVPN
 set transform-set 3DES_MD5
 set isakmp-profile spoke-profile
```

**Step 5** Define the tunnel interface on the spoke.

```
interface Tunnel1
 ip address 20.0.0.5 255.255.255.0
 ip mtu 1400
 ip flow ingress
 ip nhrp authentication DMVPNGET
 ip nhrp map 20.0.0.4 192.168.159.242
 ip nhrp map multicast 20.0.0.4
 ip nhrp network-id 2345
 ip nhrp nhs 10.0.0.4
 tunnel source 192.168.167.250
 tunnel destination 192.168.159.242
 tunnel key 1234
 tunnel protection ipsec profile SPOKE_DMVPN
```

## DMVPN Hub Configuration

Complete steps to configure the hub in the same way in which you configured the DMVPN spoke. The following is the complete configuration of the hub.

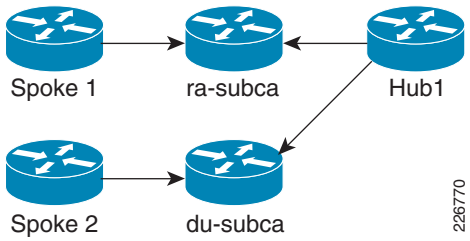
```
crypto isakmp policy 2
 encr 3des
 hash md5
 group 2
!
crypto isakmp profile dmvpn-profile
   match identity host domain cisco.com
!
!
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile ESE_DMVPN
 set transform-set 3DES_MD5
 set isakmp-profile dmvpn-profile

!
interface Tunnel1
 ip address 20.0.0.4 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication DMVPNGET
 ip nhrp map multicast dynamic
 ip nhrp network-id 2345
 no ip split-horizon eigrp 100
 tunnel source 192.168.159.242
 tunnel mode gre multipoint
 tunnel key 1234
 tunnel protection ipsec profile ESE_DMVPN
```

## DMVPN with PKI using Multiple Subordinate CAs

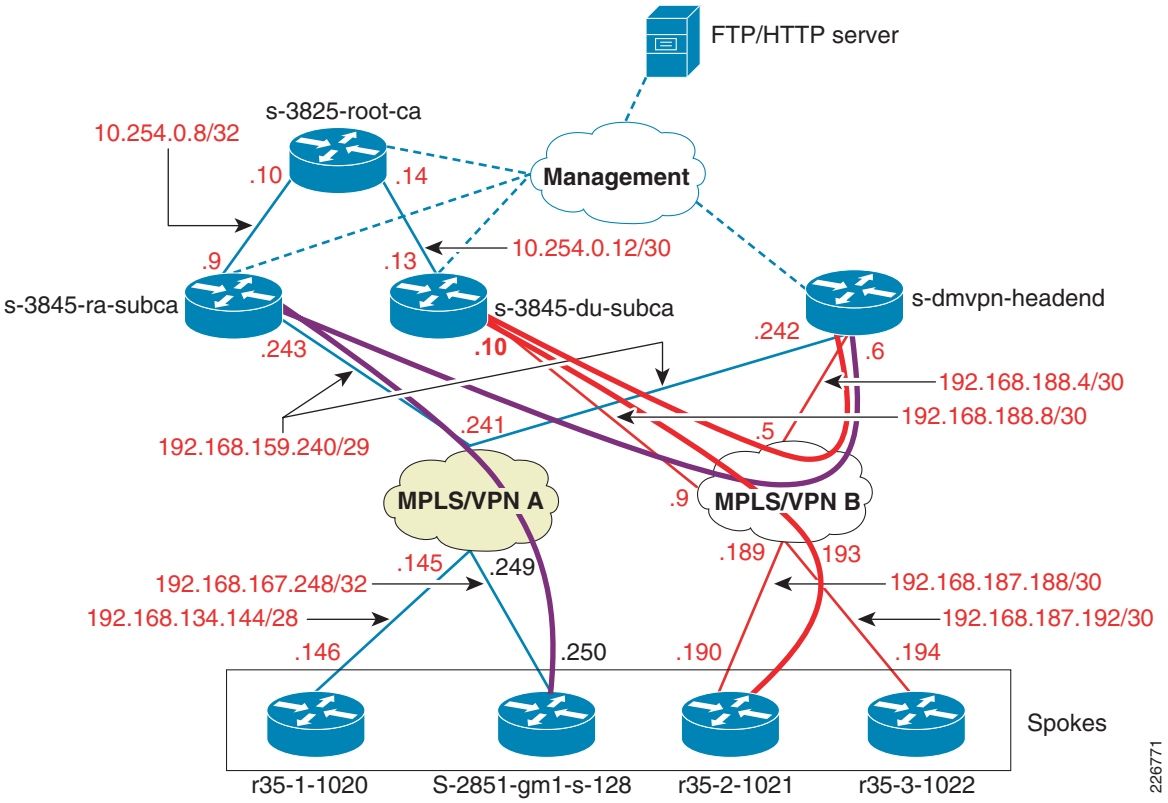
The following steps illustrate how the DMVPN hub can service the spokes enrolled in different subordinate CAs. If the entire communication is only between hub and spoke, then the spoke can enroll with either of the subordinate CAs, but the hub must enroll with both the subordinate CAs. Figure 13 illustrates this environment.

*Figure 13 DMVPN Hub Enrolling with Both Subordinate CAs*




The lab topology used to create the architecture addressed in this publications show how this can be implemented. Figure 14 depicts the hub and spoke enrolling with the subordinate CAs:

*Figure 14 Lab Topology Showing DMVPN Hub Enrolling with ra-subca and du-subca*




The following procedure details steps showing how the DMVPN hub router can authenticate spokes that have enrolled in different subordinate CAs. The DMVPN headend has previously established a DMVPN tunnel to *S-2851-gm1-s-128* which is enrolled with *ra-subca*. In the description that follows, *r35-2-1021* enrolls with *du-subca* and establishes a DMVPN tunnel with the hub router.

**Step 1** Verify the *du-subca* server configuration.

```
crypto pki server du-subca
 database level complete
 database archive pkcs12 password 7 00071A1507545A545C
 grant auto rollover ca-cert
 lifetime certificate 180
 lifetime ca-certificate 730
```

```
 cdp-url http://xxx.26.185.99/du-subca.crl
 mode sub-cs
 auto-rollover 90
 database url ftp://xxx.26.185.99
!
crypto pki trustpoint du-subca
 enrollment url http://10.254.0.14:80
 revocation-check crl
 rsakeypair du-subca
```

**Step 2** Verify that *du-subca* server is active and able to grant certificates to the branches. As noted in the best practices description, the grant mode should be set to manual—which is default behavior.

```
S-3825-du-subca# show crypto pki server
Certificate Server du-subca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=du-subca
    CA cert fingerprint: A6298B11 A50948FF C170D745 CD7DFABC
    Server configured in subordinate server mode
    Upper CA cert fingerprint: ABD85DC7 C152AE90 4949A459 B91F0A39
    Granting mode is: manual
    Last certificate issued serial number (hex): 1
    CA certificate expiration timer: 16:06:24 EST Feb 12 2011
    CRL NextUpdate timer: 16:18:57 EST Mar 27 2009
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 90 days
    Autorollover timer: 16:06:24 EST Nov 14 2010
```

**Step 3** Configure the trust point pointing to *du-subca*.

```
crypto pki trustpoint du
 enrollment url http://192.168.188.10:12345
 revocation-check crl
 rsakeypair hub-keys
 auto-enroll 70 regenerate
```

**Step 4** Authenticate the trust point.

```
s-dmvpn-headend(config)# crypto pki authenticate du
Trustpoint 'du' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
       Fingerprint MD5: A6298B11 A50948FF C170D745 CD7DFABC
      Fingerprint SHA1: AEAB0945 31F309DF 7D4DCEAF D4198981 B5B8FAE9

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
```

**Step 5** Since auto-enroll is configured, the hub router will start the enrollment request.

```
s-dmvpn-headend# show crypto pki c%
% Start certificate enrollment ..

% The subject name in the certificate will include: s-dmvpn-headend.cisco.com
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate du verbose' commandwill show the fingerprint.

s-dmvpn-headend# show crypto pki certificates
Mar 27 14:40:45.498: %PKI-6-CERTRENEWAUTO: Renewing the router certificate for trustpoint
du
Mar 27 14:40:45.502: CRYPTO_PKI:  Certificate Request Fingerprint MD5: A2FD052B 2DAD2C7E
E853C13A A74C77A5
```

```
Mar 27 14:40:45.502: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 6F3863DF 4B97CDDF
1E2F7E60 53092E85 AE9685C2
```

**Step 6** The subordinate CA is configured in manual mode. You must approve the certificate on subordinate CA.

```
S-3825-du-subca# crypto pki server du-subca grant all
Writing 2.crt !
Writing 2.cnm !
Writing du-subca.ser !
S-3825-du-subca#
```

**Step 7** Verify that hub router received the certificate.

```
s-dmvpn-headend# show crypto pki certificates du
Certificate
  Status: Available
  Certificate Serial Number (hex): 02
  Certificate Usage: General Purpose
  Issuer:
    cn=du-subca
  Subject:
    Name: s-dmvpn-headend.cisco.com
    hostname=s-dmvpn-headend.cisco.com
  CRL Distribution Points:
    http://xxx.26.185.99/du-subca.crl
  Validity Date:
    start date: 10:42:35 EST Mar 27 2009
    end   date: 10:42:35 EST Sep 23 2009
    renew date: 10:42:35 EST Jul 31 2009
  Associated Trustpoints: du
  Storage: nvram:du-subca#7.cer

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 09
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=du-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 16:06:24 EST Feb 12 2009
    end   date: 16:06:24 EST Feb 12 2011
  Associated Trustpoints: du
  Storage: nvram:root-ca#11CA.cer
```

**Step 8** Repeat Steps 3 through 8 on *r35-2-1021* for the enrollment with *du-subca*.

**Step 9** Configure the enrollment request on *r35-2-1021*.

```
crypto pki trustpoint du
 enrollment url http://192.168.188.10:12345
 revocation-check none
 rsakeypair r35-2-keys
 auto-enroll 70 regenerate
```

**Step 10** Authenticate the subordinate CA.

```
r35-2-1021(config)# crypto pki authenticate du
Trustpoint 'du' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
       Fingerprint MD5: A6298B11 A50948FF C170D745 CD7DFABC
      Fingerprint SHA1: AEAB0945 31F309DF 7D4DCEAF D4198981 B5B8FAE9
```

```
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
```

The spoke will try to renew the certificate with the subordinate CA.

```
*Mar 27 13:18:01.505 EDT: %PKI-6-CERTRENEWAUTO: Renewing the router certificate for
trustpoint du
*Mar 27 13:18:02.485 EDT: %CRYPTO-6-AUTOGEN: Generated new 512 bit key pair
*Mar 27 13:18:02.509 EDT: CRYPTO_PKI:  Certificate Request Fingerprint MD5: F4E358C9
58F20C5A C6EEC862 1639D4CA
*Mar 27 13:18:02.509 EDT: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 5B4D5429
B6CFFE9E B01154EE 41189B4A 78375D20
```

**Step 11** Grant the certificate at the subordinate CA.

```
S-3825-du-subca# crypto pki server du-subca grant all
Writing 3.crt !
Writing 3.cnm !
Writing du-subca.ser !
S-3825-du-subca#
```

**Step 12** Verify that the certificate was granted at subordinate CA.

```
r35-2-1021# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 0x3
  Certificate Usage: General Purpose
  Issuer:
    cn=du-subca
  Subject:
    Name: r35-2-1021
    hostname=r35-2-1021
  CRL Distribution Points:
    http://xxx.26.185.99/du-subca.crl
  Validity Date:
    start date: 12:39:56 EDT Mar 27 2009
    end   date: 12:39:56 EDT Sep 23 2009
    renew date: 12:39:56 EDT Jul 31 2009
  Associated Trustpoints: du

CA Certificate
  Status: Available
  Certificate Serial Number: 0x9
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=du-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 16:06:24 EST Feb 12 2009
    end   date: 16:06:24 EST Feb 12 2011
  Associated Trustpoints: du
```

**Step 13** Configure the DMVPN configuration on the *r35-2-1021*.

```
crypto isakmp policy 1
 encr 3des
 hash md5
 group 2
crypto isakmp identity hostname
crypto isakmp profile r35-profile
```

```
   match identity host domain cisco.com
!
!
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile ESE_DMVPN
 set transform-set 3DES_MD5
 set isakmp-profile r35-profile
!
interface Tunnel1
 ip address 20.0.0.2 255.255.255.0
 ip mtu 1400
 ip nhrp authentication DMVPNGET
 ip nhrp map 20.0.0.4 192.168.188.6
 ip nhrp map multicast 20.0.0.4
 ip nhrp network-id 2345
 ip nhrp nhs 20.0.0.4
 tunnel source 192.168.187.190
 tunnel destination 192.168.188.6
 tunnel key 1234
 tunnel protection ipsec profile ESE_DMVPN
```

**Step 14** Verify DMVPN tunnel with PKI is enabled.

Use the following **show crypto** commands on either end of the tunnel to verify that the DMVPN tunnel with PKI is enabled.

Verification at spoke:

```
S-2851-gm1-s-128# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
192.168.159.242 192.168.167.250 QM_IDLE           4218    0 ACTIVE spoke-profile

IPv6 Crypto ISAKMP SA
```

Verification at the headend:

```
s-dmvpn-headend# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
192.168.188.6   192.168.187.186 QM_IDLE          13060    0 ACTIVE
192.168.159.242 192.168.167.250 QM_IDLE          13063    0 ACTIVE
192.168.188.6   192.168.187.190 QM_IDLE          13064    0 ACTIVE
```

## Migration of DMVPN from Pre-shared to PKI

This section describes the process of migrating a spoke router that implements a DMVPN tunnel to the hub (using pre-shared key) into having a tunnel that uses certificates for authentication. This example features spoke router *r35-1-1020*. The following is the relevant configuration for *r35-1-1020*.

```
crypto keyring giraffe
  pre-shared-key address 192.168.159.242 key <strong_psk>
!
crypto isakmp policy 1
 encr 3des
 hash md5

 authentication pre-share
 group 2
crypto isakmp profile r35-profile
```

```
   keyring giraffe
   match identity host domain cisco.com
   match identity address 192.168.159.242 255.255.255.255
!
!
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile ESE_DMVPN
 set transform-set 3DES_MD5
 set isakmp-profile r35-profile


interface Tunnel1
 ip address 20.0.0.1 255.255.255.0
 ip mtu 1400
 ip nhrp authentication DMVPNGET
 ip nhrp map 20.0.0.4 192.168.159.242
 ip nhrp map multicast 20.0.0.4
 ip nhrp network-id 2345
 ip nhrp nhs 20.0.0.4
 ip route-cache flow
 tunnel source 192.168.134.146
 tunnel destination 192.168.159.242
 tunnel key 1234
 tunnel protection ipsec profile ESE_DMVPN
!
```

With the preceding configuration, the spoke will establish the DMVPN tunnel with the hub router. The following **show** commands verify two things:

- DMVPN tunnel is up.
- Tunnel is using pre-shared key for authentication.

```
r35-1-1020# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
192.168.159.242 192.168.134.146 QM_IDLE           4001    0 ACTIVE

r35-1-1020# show crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

4001  192.168.134.146 192.168.159.242          ACTIVE 3des md5  psk  2  23:51:08
       Engine-id:Conn-id =  AIM-VPN/EPII-PLUS:1

IPv6 Crypto ISAKMP SA

r35-1-1020#
```

The following configuration listing illustrates the relevant DMVPN hub configuration.

```
crypto keyring zebra
  pre-shared-key address 192.168.187.190 key CISCO
  pre-shared-key address 192.168.187.194 key CISCO
  pre-shared-key address 192.168.134.146 key CISCO
!

crypto isakmp policy 1 ! The first policy is meant for pre-shared authentication.
```

```
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto isakmp policy 2 ! The second policy for rsa-signature authentication
 encr 3des
 hash md5
 group 2
```

**Tip** The ISAKMP profile provides a template for matching identities. In the following example, the ISAKMP profile matches either domain name or pre-shared secret.

```
crypto isakmp profile dmvpn-profile
! The isakmp profile will match either pre-shared or keyring zebra.
   match identity host domain cisco.com
   match identity address 192.168.134.146 255.255.255.255
!
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile ESE_DMVPN
 set transform-set 3DES_MD5
 set isakmp-profile dmvpn-profile
```

With this initial setup in place with the preceding configuration examples, the following steps illustrate the process of migrating from the pre-shared key implementation to a certificate-based authentication environment.

**Step 1** As shown in spoke enrollment example, *r35-1-1020* must to enroll with *sub-ca* in order to obtain a certificate. Because this spoke is connected to MPLS/VPN A, it would register with *ra-subca*. The following is the configuration for enrollment:

```
crypto pki trustpoint ra
 enrollment url http://192.168.159.243:12345
 serial-number
 ip-address 192.168.159.243
 revocation-check none
 rsakeypair ra
 auto-enroll 80 regenerate
!
```

**Step 2** Once the spoke is enrolled with the *sub-ca*, the following show command presents the content of the spoke (*r35-1-1020*) certificate:

```
r35-1-1020# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 0x9
  Certificate Usage: General Purpose
  Issuer:
    cn=ra-subca
  Subject:
    Name: r35-1-1020.cisco.com
    IP Address: 192.168.159.243
    Serial Number: FTX1048A6QA
    serialNumber=FTX1048A6QA+ipaddress=192.168.159.243+hostname=r35-1-1020.cisco.com
  CRL Distribution Points:
    http://xxx.26.185.99/ra-subca.crl
  Validity Date:
```

```
    start date: 10:44:55 EST Mar 6 2009
    end   date: 11:44:55 EDT Sep 2 2009
    renew date: 11:44:55 EDT Jul 28 2009
  Associated Trustpoints: ra
  Storage: nvram:ra-subca#9.cer

CA Certificate
  Status: Available
  Certificate Serial Number: 0x8
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=ra-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 11:21:19 EST Jan 28 2009
    end   date: 11:21:19 EST Jan 28 2011
  Associated Trustpoints: ra
  Storage: nvram:root-ca#8CA.cer


r35-1-1020#
```

**Step 3** Change the ISAKMP policy on the spoke to use RSA signatures.

```
crypto isakmp policy 1
 encr 3des
 hash md5
 group 2
```

**Step 4** Flap the tunnel.

r35-1-1020(config)# interface tunnel 1

```
shut
no shut
```

**Step 5** With the above steps, the DMVPN tunnel should be up with certificate-based authentication. The following **debug** command output is an example of output captured from the spoke router.

```
Apr  2 17:36:24.796 EDT: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
Apr  2 17:36:26.792 EDT: %LINK-5-CHANGED: Interface Tunnel1, changed state to
administratively down
Apr  2 17:36:27.792 EDT: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed
state to down
Apr  2 17:36:29.188 EDT: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
Apr  2 17:36:29.192 EDT: ISAKMP:(0): SA request profile is r35-profile
Apr  2 17:36:29.192 EDT: ISAKMP: Created a peer struct for 192.168.159.242, peer port 500
Apr  2 17:36:29.192 EDT: ISAKMP: New peer created peer = 0x48B7D7B4 peer_handle =
0x80000004
Apr  2 17:36:29.192 EDT: ISAKMP: Locking peer struct 0x48B7D7B4, refcount 1 for
isakmp_initiator
Apr  2 17:36:29.192 EDT: ISAKMP: local port 500, remote port 500
Apr  2 17:36:29.192 EDT: ISAKMP: set new node 0 to QM_IDLE
Apr  2 17:36:29.192 EDT: insert sa successfully sa = 4BEBCE20
Apr  2 17:36:29.192 EDT: ISAKMP:(0):Can not start Aggressive mode, trying Main mode.
Apr  2 17:36:29.192 EDT: ISAKMP:(0):Found ADDRESS key in keyring giraffe
Apr  2 17:36:29.192 EDT: ISAKMP:(0): constructed NAT-T vendor-rfc3947 ID
Apr  2 17:36:29.192 EDT: ISAKMP:(0): constructed NAT-T vendor-07 ID
Apr  2 17:36:29.192 EDT: ISAKMP:(0): constructed NAT-T vendor-03 ID
Apr  2 17:36:29.192 EDT: ISAKMP:(0): constructed NAT-T vendor-02 ID
Apr  2 17:36:29.192 EDT: ISAKMP:(0):Input = IKE_MESG_FROM_IPSEC, IKE_SA_REQ_MM
Apr  2 17:36:29.192 EDT: ISAKMP:(0):Old State = IKE_READY  New State = IKE_I_MM1
```

```
Apr  2 17:36:29.192 EDT: ISAKMP:(0): beginning Main Mode exchange
Apr  2 17:36:29.196 EDT: ISAKMP:(0): sending packet to 192.168.159.242 my_port 500
peer_port 500 (I) MM_NO_STATE
Apr  2 17:36:29.196 EDT: ISAKMP:(0):Sending an IKE IPv4 Packet.
Apr  2 17:36:29.196 EDT: ISAKMP (0:0): received packet from 192.168.159.242 dport 500
sport 500 Global (I) MM_NO_STATE
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Old State = IKE_I_MM1  New State = IKE_I_MM2

Apr  2 17:36:29.200 EDT: ISAKMP:(0): processing SA payload. message ID = 0
Apr  2 17:36:29.200 EDT: ISAKMP:(0): processing vendor id payload
Apr  2 17:36:29.200 EDT: ISAKMP:(0): vendor ID seems Unity/DPD but major 69 mismatch
Apr  2 17:36:29.200 EDT: ISAKMP (0:0): vendor ID is NAT-T RFC 3947
Apr  2 17:36:29.200 EDT: ISAKMP : Looking for xauth in profile r35-profile
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
Apr  2 17:36:29.200 EDT: ISAKMP:      encryption 3DES-CBC
Apr  2 17:36:29.200 EDT: ISAKMP:      hash MD5
Apr  2 17:36:29.200 EDT: ISAKMP:      default group 2
Apr  2 17:36:29.200 EDT: ISAKMP:      auth RSA sig
Apr  2 17:36:29.200 EDT: ISAKMP:      life type in seconds
Apr  2 17:36:29.200 EDT: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
Apr  2 17:36:29.200 EDT: ISAKMP:(0):atts are acceptable. Next payload is 0
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Acceptable atts:actual life: 0
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Acceptable atts:life: 0
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Fill atts in sa vpi_length:4
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Fill atts in sa life_in_seconds:86400
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Returning Actual lifetime: 86400
Apr  2 17:36:29.200 EDT: ISAKMP:(0)::Started lifetime timer: 86400.

Apr  2 17:36:29.200 EDT: ISAKMP:(0): processing vendor id payload
Apr  2 17:36:29.200 EDT: ISAKMP:(0): vendor ID seems Unity/DPD but major 69 mismatch
Apr  2 17:36:29.200 EDT: ISAKMP (0:0): vendor ID is NAT-T RFC 3947
Apr  2 17:36:29.200 EDT: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  2 17:36:29.204 EDT: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM2

Apr  2 17:36:29.204 EDT: ISAKMP (0:0): constructing CERT_REQ for issuer cn=ra-subca
Apr  2 17:36:29.204 EDT: ISAKMP:(0): sending packet to 192.168.159.242 my_port 500
peer_port 500 (I) MM_SA_SETUP
Apr  2 17:36:29.204 EDT: ISAKMP:(0):Sending an IKE IPv4 Packet.
Apr  2 17:36:29.204 EDT: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Apr  2 17:36:29.204 EDT: ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3

Apr  2 17:36:29.212 EDT: ISAKMP (0:0): received packet from 192.168.159.242 dport 500
sport 500 Global (I) MM_SA_SETUP
Apr  2 17:36:29.212 EDT: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  2 17:36:29.212 EDT: ISAKMP:(0):Old State = IKE_I_MM3  New State = IKE_I_MM4

Apr  2 17:36:29.212 EDT: ISAKMP:(0): processing KE payload. message ID = 0
Apr  2 17:36:29.220 EDT: ISAKMP:(0): processing NONCE payload. message ID = 0
Apr  2 17:36:29.224 EDT: ISAKMP:(4003): processing CERT_REQ payload. message ID = 0
Apr  2 17:36:29.224 EDT: ISAKMP:(4003): peer wants a CT_X509_SIGNATURE cert
Apr  2 17:36:29.224 EDT: ISAKMP:(4003): peer wants cert issued by cn=du-subca
Apr  2 17:36:29.224 EDT: ISAKMP:(4003): issuer name is not a trusted root.
Apr  2 17:36:29.224 EDT: ISAKMP:(4003): processing CERT_REQ payload. message ID = 0
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): peer wants a CT_X509_SIGNATURE cert
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): peer wants cert issued by cn=ra-subca
Apr  2 17:36:29.228 EDT:  Choosing trustpoint ra as issuer
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): processing vendor id payload
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): vendor ID is Unity
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): processing vendor id payload
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): vendor ID is DPD
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): processing vendor id payload
```

```
Apr  2 17:36:29.228 EDT: ISAKMP:(4003): speaking to another IOS box!

Apr  2 17:36:29.228 EDT: ISAKMP:received payload type 20
Apr  2 17:36:29.228 EDT: ISAKMP:received payload type 20
Apr  2 17:36:29.228 EDT: ISAKMP:(4003):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  2 17:36:29.228 EDT: ISAKMP:(4003):Old State = IKE_I_MM4  New State = IKE_I_MM4

Apr  2 17:36:29.228 EDT: ISAKMP:(4003):Send initial contact
Apr  2 17:36:29.232 EDT: ISAKMP:(4003):My ID configured as IPv4 Addr, but Addr not in
Cert!
Apr  2 17:36:29.232 EDT: ISAKMP:(4003):Using FQDN as My ID
Apr  2 17:36:29.232 EDT: ISAKMP:(4003):SA is doing RSA signature authentication using id
type ID_FQDN
Apr  2 17:36:29.232 EDT: ISAKMP (0:4003): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : r35-1-1020.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
Apr  2 17:36:29.232 EDT: ISAKMP:(4003):Total payload length: 28
Apr  2 17:36:29.240 EDT: ISAKMP (0:4003): constructing CERT payload for
serialNumber=FTX1048A6QA+ipaddress=192.168.159.243+hostname=r35-1-1020.cisco.com
Apr  2 17:36:29.240 EDT: ISAKMP:(4003): using the ra trustpoint's keypair to sign
Apr  2 17:36:29.252 EDT: ISAKMP:(4003): sending packet to 192.168.159.242 my_port 500
peer_port 500 (I) MM_KEY_EXCH
Apr  2 17:36:29.252 EDT: ISAKMP:(4003):Sending an IKE IPv4 Packet.
Apr  2 17:36:29.252 EDT: ISAKMP:(4003):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Apr  2 17:36:29.252 EDT: ISAKMP:(4003):Old State = IKE_I_MM4  New State = IKE_I_MM5

Apr  2 17:36:29.380 EDT: ISAKMP (0:4003): received packet from 192.168.159.242 dport 500
sport 500 Global (I) MM_KEY_EXCH
Apr  2 17:36:29.380 EDT: ISAKMP:(4003): processing ID payload. message ID = 0
Apr  2 17:36:29.380 EDT: ISAKMP (0:4003): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : s-dmvpn-headend.cisco.com
        protocol     : 17
        port         : 500
        length       : 33
Apr  2 17:36:29.380 EDT: ISAKMP:(4003): processing CERT payload. message ID = 0
Apr  2 17:36:29.380 EDT: ISAKMP:(4003): processing a CT_X509_SIGNATURE cert
Apr  2 17:36:29.384 EDT: ISAKMP:(4003): peer's pubkey isn't cached
Apr  2 17:36:29.392 EDT: ISAKMP:(4003): Unable to get DN from certificate!
Apr  2 17:36:29.392 EDT: ISAKMP:(4003): Cert presented by peer contains no OU field.
Apr  2 17:36:29.396 EDT: ISAKMP:(4003): processing SIG payload. message ID = 0
Apr  2 17:36:29.400 EDT: ISAKMP:(4003):SA authentication status:
        authenticated
Apr  2 17:36:29.400 EDT: ISAKMP:(4003):SA has been authenticated with 192.168.159.242
Apr  2 17:36:29.400 EDT: ISAKMP: Trying to insert a peer
192.168.134.146/192.168.159.242/500/,  and inserted successfully 48B7D7B4.
Apr  2 17:36:29.400 EDT: ISAKMP:(4003):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  2 17:36:29.400 EDT: ISAKMP:(4003):Old State = IKE_I_MM5  New State = IKE_I_MM6

Apr  2 17:36:29.400 EDT: ISAKMP (0:4003): received packet from 192.168.159.242 dport 500
sport 500 Global (I) MM_KEY_EXCH
Apr  2 17:36:29.400 EDT: ISAKMP: set new node -1008416615 to QM_IDLE
Apr  2 17:36:29.404 EDT: ISAKMP:(4003): processing HASH payload. message ID = -1008416615
Apr  2 17:36:29.404 EDT: ISAKMP:(4003): processing DELETE payload. message ID =
-1008416615
Apr  2 17:36:29.404 EDT: ISAKMP:(4003):peer does not do paranoid keepalives.

Apr  2 17:36:29.404 EDT: ISAKMP:(4003):deleting node -1008416615 error FALSE reason
"Informational (in) state 1"
```

```
Apr  2 17:36:29.404 EDT: ISAKMP:(4003):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  2 17:36:29.404 EDT: ISAKMP:(4003):Old State = IKE_I_MM6  New State = IKE_I_MM6


Apr  2 17:36:29.404 EDT: ISAKMP:(4003):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Apr  2 17:36:29.404 EDT: ISAKMP:(4003):Old State = IKE_I_MM6  New State = IKE_P1_COMPLETE

Apr  2 17:36:29.404 EDT: ISAKMP:(4003):beginning Quick Mode exchange, M-ID of -1182603547
Apr  2 17:36:29.404 EDT: ISAKMP:(4003):QM Initiator gets spi
Apr  2 17:36:29.408 EDT: ISAKMP:(4003): sending packet to 192.168.159.242 my_port 500
peer_port 500 (I) QM_IDLE
Apr  2 17:36:29.408 EDT: ISAKMP:(4003):Sending an IKE IPv4 Packet.
Apr  2 17:36:29.408 EDT: ISAKMP:(4003):Node -1182603547, Input = IKE_MESG_INTERNAL,
IKE_INIT_QM
Apr  2 17:36:29.408 EDT: ISAKMP:(4003):Old State = IKE_QM_READY  New State = IKE_QM_I_QM1
Apr  2 17:36:29.412 EDT: ISAKMP:(4003):Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
Apr  2 17:36:29.412 EDT: ISAKMP:(4003):Old State = IKE_P1_COMPLETE  New State =
IKE_P1_COMPLETE

Apr  2 17:36:29.416 EDT: ISAKMP (0:4003): received packet from 192.168.159.242 dport 500
sport 500 Global (I) QM_IDLE
Apr  2 17:36:29.420 EDT: ISAKMP:(4003): processing HASH payload. message ID = -1182603547
Apr  2 17:36:29.420 EDT: ISAKMP:(4003): processing SA payload. message ID = -1182603547
Apr  2 17:36:29.420 EDT: ISAKMP:(4003):Checking IPSec proposal 1
Apr  2 17:36:29.420 EDT: ISAKMP: transform 1, ESP_3DES
Apr  2 17:36:29.420 EDT: ISAKMP:   attributes in transform:
Apr  2 17:36:29.420 EDT: ISAKMP:      encaps is 2 (Transport)
Apr  2 17:36:29.420 EDT: ISAKMP:      SA life type in seconds
Apr  2 17:36:29.420 EDT: ISAKMP:      SA life duration (basic) of 3600
Apr  2 17:36:29.420 EDT: ISAKMP:      SA life type in kilobytes
Apr  2 17:36:29.420 EDT: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
Apr  2 17:36:29.420 EDT: ISAKMP:      authenticator is HMAC-MD5
Apr  2 17:36:29.420 EDT: ISAKMP:(4003):atts are acceptable.
Apr  2 17:36:29.424 EDT: ISAKMP:(4003): processing NONCE payload. message ID = -1182603547
Apr  2 17:36:29.424 EDT: ISAKMP:(4003): processing ID payload. message ID = -1182603547
Apr  2 17:36:29.424 EDT: ISAKMP:(4003): processing ID payload. message ID = -1182603547
Apr  2 17:36:29.432 EDT: ISAKMP:(4003): Creating IPSec SAs
Apr  2 17:36:29.432 EDT:         inbound SA from 192.168.159.242 to 192.168.134.146 (f/i)
0/ 0
        (proxy 192.168.159.242 to 192.168.134.146)
Apr  2 17:36:29.432 EDT:          has spi 0x11689A5F and conn_id 0
Apr  2 17:36:29.432 EDT:          lifetime of 3600 seconds
Apr  2 17:36:29.432 EDT:          lifetime of 4608000 kilobytes
Apr  2 17:36:29.432 EDT:         outbound SA from 192.168.134.146 to 192.168.159.242 (f/i)
0/0
        (proxy 192.168.134.146 to 192.168.159.242)
Apr  2 17:36:29.432 EDT:          has spi  0x9375DD09 and conn_id 0
Apr  2 17:36:29.432 EDT:          lifetime of 3600 seconds
Apr  2 17:36:29.432 EDT:          lifetime of 4608000 kilobytes
Apr  2 17:36:29.436 EDT: ISAKMP:(4003): sending packet to 192.168.159.242 my_port 500
peer_port 500 (I) QM_IDLE
Apr  2 17:36:29.436 EDT: ISAKMP:(4003):Sending an IKE IPv4 Packet.
Apr  2 17:36:29.436 EDT: ISAKMP:(4003):deleting node -1182603547 error FALSE reason "No
Error"
Apr  2 17:36:29.436 EDT: ISAKMP:(4003):Node -1182603547, Input = IKE_MESG_FROM_PEER,
IKE_QM_EXCH
Apr  2 17:36:29.436 EDT: ISAKMP:(4003):Old State = IKE_QM_I_QM1  New State =
IKE_QM_PHASE2_COMPLETE
Apr  2 17:36:31.180 EDT: %LINK-3-UPDOWN: Interface Tunnel1, changed state to up
Apr  2 17:36:31.332 EDT: %SYS-5-CONFIG_I: Configured from console by vty1 (64.102.87.234)
Apr  2 17:36:32.180 EDT: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed
state to up
```

**Step 6** Use the following **show** commands to verify the DMVPN tunnel status.

```
r35-1-1020# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status

192.168.159.242 192.168.134.146 QM_IDLE           4003    0 ACTIVE r35-profile

IPv6 Crypto ISAKMP SA

r35-1-1020# show crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

4003  192.168.134.146 192.168.159.242          ACTIVE 3des md5  rsig 2  23:53:33
       Engine-id:Conn-id =  AIM-VPN/EPII-PLUS:3

IPv6 Crypto ISAKMP SA

r35-1-1020#
```

## Revocation Checking

The following section describes the process of revocation used in the network deployment. To illustrate this process, *r35-3-1022* is revoked. After revoking the certificate, the spoke will not be able to establish the tunnel to the hub router.

**Step 1** Before beginning a revocation, verify that the spoke has a valid certificate and that it has an established DMVPN tunnel to the hub router. The following is the configuration of *r35-3-1022* before its certificate is revoked.

```
crypto pki trustpoint du
 enrollment url http://192.168.188.10:12345
 serial-number
 ip-address 192.168.187.194
 revocation-check none
 rsakeypair r35-3-keys
 auto-enroll 70 regenerate
!
!
crypto isakmp policy 1
 encr 3des
 hash md5
 group 2
crypto isakmp identity hostname
crypto isakmp profile r35-profile
   match identity host domain cisco.com
!
!

crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
 mode transport
!
crypto ipsec profile ESE_DMVPN
```

```
 set transform-set 3DES_MD5
 set isakmp-profile r35-profile
!
interface Tunnel1
 ip address 20.0.0.3 255.255.255.0
 ip mtu 1400
 ip nhrp authentication DMVPNGET
 ip nhrp map 20.0.0.4 192.168.188.6
 ip nhrp map multicast 20.0.0.4
 ip nhrp network-id 2345
 ip nhrp nhs 20.0.0.4
 ip route-cache flow
 tunnel source 192.168.187.194
 tunnel destination 192.168.188.6
 tunnel key 1234
 tunnel protection ipsec profile ESE_DMVPN
!
```

**Step 2** Verify the certificates in *r35-3-1021*.

```
r35-3-1022# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 0x4 ! Serial number of the certificate
  Certificate Usage: General Purpose
  Issuer:
    cn=du-subca
  Subject:
    Name: r35-3-1022
    IP Address: 192.168.187.194
    Serial Number: FTX1048A6DW
    serialNumber=FTX1048A6DW+hostname=r35-3-1022+ipaddress=192.168.187.194
  CRL Distribution Points:
    http://xxx.26.185.99/du-subca.crl
  Validity Date:
    start date: 10:16:36 EDT Apr 5 2009
    end   date: 10:16:36 EDT Oct 2 2009
    renew date: 10:16:36 EDT Aug 9 2009
  Associated Trustpoints: du

CA Certificate
  Status: Available
  Certificate Serial Number: 0x9
  Certificate Usage: Signature
  Issuer:
    cn=root-ca
  Subject:
    cn=du-subca
  CRL Distribution Points:
    http://xxx.26.185.99/root-ca.crl
  Validity Date:
    start date: 16:06:24 EST Feb 12 2009
    end   date: 16:06:24 EST Feb 12 2011
  Associated Trustpoints: du


r35-3-1022#
```

**Step 3** Verify the DMVPN tunnel.

```
r35-3-1022# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
192.168.188.6   192.168.187.194 QM_IDLE           5670    0 ACTIVE r35-profile
```

```
IPv6 Crypto ISAKMP SA

r35-3-1022#
r35-3-1022# show crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

5670  192.168.187.194 192.168.188.6            ACTIVE 3des md5  rsig 2  23:39:58
       Engine-id:Conn-id =  AIM-VPN/EPII-PLUS:1670

IPv6 Crypto ISAKMP SA

r35-3-1022# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
192.168.188.6   192.168.187.194 QM_IDLE           5670    0 ACTIVE r35-profile

IPv6 Crypto ISAKMP SA
```

**Step 4** Verify the DMVPN tunnel at the headend.

```
s-dmvpn-headend# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
192.168.188.6   192.168.187.190 QM_IDLE          13017    0 ACTIVE
192.168.188.6   192.168.187.194 QM_IDLE          13020    0 ACTIVE
192.168.159.242 192.168.134.146 QM_IDLE          13018    0 ACTIVE
192.168.159.242 192.168.167.250 QM_IDLE          13016    0 ACTIVE

IPv6 Crypto ISAKMP SA
```

**Step 5** Verify CRL functionality and revoke 0x4, which is *r35-3-1022*.

```
S-3825-du-subca# crypto pki server du-subca revoke ?
  <0x0-0xFFFFFFFF>  Serial Number in Hexadecimal.

S-3825-du-subca# crypto pki server du-subca revoke 0X4
Writing du-subca.crl !
Writing du-subca.crl !
% Certificate 04 succesfully revoked.
```

**Step 6** The hub router CRL list is updated every six hours. Until the headend requests the CRL list, the spoke can still establish the VPN tunnel. The following command shows the current CRL state of the headend router.

```
s-dmvpn-headend# show crypto pki crls
CRL Issuer Name:
    cn=du-subca
    LastUpdate: 10:03:28 EST Apr 5 2009
    NextUpdate: 16:03:28 EST Apr 5 2009
    Retrieved from CRL Distribution Point:
 CRL DER is 218 bytes
 CRL is stored in parsed CRL cache


Parsed CRL cache current size is 218 bytes
Parsed CRL cache maximum size is 65536 bytes
```

```
s-dmvpn-headend#
s-dmvpn-headend# show clock
15:06:42.055 EST Sun Apr 5 2009
s-dmvpn-headend#
```

**Step 7** The headend has one more hour until the CRL timer expires. You can manually update the CRL list. The following commands illustrate manual CRL updating.

```
s-dmvpn-headend(config)# crypto pki crl request du
s-dmvpn-headend(config)# end
s-dmvpn-headend#show
Apr  5 19:09:41.511: %SYS-5-CONFIG_I: Configured from console by consolecry
s-dmvpn-headend# show crypto pki crls
CRL Issuer Name:
    cn=du-subca
    LastUpdate: 13:28:25 EST Apr 5 2009
    NextUpdate: 19:28:25 EST Apr 5 2009
    Retrieved from CRL Distribution Point:
 CRL DER is 240 bytes
 CRL is stored in parsed CRL cache


Parsed CRL cache current size is 458 bytes
Parsed CRL cache maximum size is 65536 bytes

s-dmvpn-headend#
```

**Step 8** To verify whether the headend rejects the tunnel, flap the tunnel at the spoke and enable the **debug** command at the headend as follows:

```
s-dmvpn-headend# debug crypto isakmp
Crypto ISAKMP debugging is on
s-dmvpn-headend#
Apr  5 19:14:23.325: ISAKMP:(13017):purging node -1291694641
Apr  5 19:14:30.114: ISAKMP:(13021):purging node -1518526702
Apr  5 19:14:31.722: ISAKMP:(13016):purging node 372434734
Apr  5 19:14:34.446: ISAKMP (0:13021): received packet from 192.168.187.194 dport 500
sport 500 Global (R) QM_IDLE
Apr  5 19:14:34.450: ISAKMP: set new node -784080745 to QM_IDLE
Apr  5 19:14:34.450: ISAKMP:(13021): processing HASH payload. message ID = -784080745
Apr  5 19:14:34.450: ISAKMP:received payload type 18
Apr  5 19:14:34.450: ISAKMP:(13021):Processing delete with reason payload
Apr  5 19:14:34.450: ISAKMP:(13021):delete doi = 1
Apr  5 19:14:34.450: ISAKMP:(13021):delete protocol id = 1
Apr  5 19:14:34.450: ISAKMP:(13021):delete spi_size =  16
Apr  5 19:14:34.450: ISAKMP:(13021):delete num spis = 1
Apr  5 19:14:34.450: ISAKMP:(13021):delete_reason = 6
Apr  5 19:14:34.450: ISAKMP:(13021): processing DELETE_WITH_REASON payload, message ID =
-784080745, reason: Unknown delete reason!
Apr  5 19:14:34.450: ISAKMP:(13021):peer does not do paranoid keepalives.

Apr  5 19:14:34.450: ISAKMP:(13021):deleting SA reason "P1 delete notify (in)" state (R)
QM_IDLE       (peer 192.168.187.194)
Apr  5 19:14:34.450: ISAKMP:(13021):deleting node -784080745 error FALSE reason
"Informational (in) state 1"
Apr  5 19:14:34.450: ISAKMP: set new node 890235939 to QM_IDLE
Apr  5 19:14:34.450: ISAKMP:(13021): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) QM_IDLE
Apr  5 19:14:34.450: ISAKMP:(13021):Sending an IKE IPv4 Packet.
Apr  5 19:14:34.450: ISAKMP:(13021):purging node 890235939
Apr  5 19:14:34.450: ISAKMP:(13021):Input = IKE_MESG_INTERNAL, IKE_PHASE1_DEL
Apr  5 19:14:34.450: ISAKMP:(13021):Old State = IKE_P1_COMPLETE  New State = IKE_DEST_SA
```

```
Apr  5 19:14:34.450: ISAKMP:(13021):deleting SA reason "P1 delete notify (in)" state (R)
QM_IDLE       (peer 192.168.187.194)
Apr  5 19:14:34.450: ISAKMP:(0):Can't decrement IKE Call Admission Control stat
incoming_active since it's already 0.
Apr  5 19:14:34.450: ISAKMP: Unlocking peer struct 0x8101A6C for isadb_mark_sa_deleted(),
count 0
Apr  5 19:14:34.450: ISAKMP: Deleting peer node by peer_reap for 192.168.187.194: 8101A6C
Apr  5 19:14:34.450: ISAKMP:(13021):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:34.450: ISAKMP:(13021):Old State = IKE_DEST_SA  New State = IKE_DEST_SA

Apr  5 19:14:38.506: ISAKMP (0:0): received packet from 192.168.187.194 dport 500 sport
500 Global (N) NEW SA
Apr  5 19:14:38.506: ISAKMP: Created a peer struct for 192.168.187.194, peer port 500
Apr  5 19:14:38.506: ISAKMP: New peer created peer = 0x8101A6C peer_handle = 0x8000001F
Apr  5 19:14:38.506: ISAKMP: Locking peer struct 0x8101A6C, refcount 1 for
crypto_isakmp_process_block
Apr  5 19:14:38.506: ISAKMP: local port 500, remote port 500
Apr  5 19:14:38.506: ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa
= 8353390
Apr  5 19:14:38.506: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:38.506: ISAKMP:(0):Old State = IKE_READY  New State = IKE_R_MM1

Apr  5 19:14:38.506: ISAKMP:(0): processing SA payload. message ID = 0
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 69 mismatch
Apr  5 19:14:38.506: ISAKMP (0:0): vendor ID is NAT-T RFC 3947
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
Apr  5 19:14:38.506: ISAKMP (0:0): vendor ID is NAT-T v7
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 157 mismatch
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID is NAT-T v3
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID is NAT-T v2
Apr  5 19:14:38.506: ISAKMP:(0):found peer pre-shared key matching 192.168.187.194
Apr  5 19:14:38.506: ISAKMP:(0): local preshared key found
Apr  5 19:14:38.506: ISAKMP : Scanning profiles for xauth ... dmvpn-profile
Apr  5 19:14:38.506: ISAKMP:(0):Checking ISAKMP transform 1 against priority 1 policy
Apr  5 19:14:38.506: ISAKMP:      encryption 3DES-CBC
Apr  5 19:14:38.506: ISAKMP:      hash MD5
Apr  5 19:14:38.506: ISAKMP:      default group 2
Apr  5 19:14:38.506: ISAKMP:      auth RSA sig
Apr  5 19:14:38.506: ISAKMP:      life type in seconds
Apr  5 19:14:38.506: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
Apr  5 19:14:38.506: ISAKMP:(0):Authentication method offered does not match policy!
Apr  5 19:14:38.506: ISAKMP:(0):atts are not acceptable. Next payload is 3
Apr  5 19:14:38.506: ISAKMP:(0):Checking ISAKMP transform 2 against priority 1 policy
Apr  5 19:14:38.506: ISAKMP:      encryption DES-CBC
Apr  5 19:14:38.506: ISAKMP:      hash SHA
Apr  5 19:14:38.506: ISAKMP:      default group 1
Apr  5 19:14:38.506: ISAKMP:      auth RSA sig
Apr  5 19:14:38.506: ISAKMP:      life type in seconds
Apr  5 19:14:38.506: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
Apr  5 19:14:38.506: ISAKMP:(0):Encryption algorithm offered does not match policy!
Apr  5 19:14:38.506: ISAKMP:(0):atts are not acceptable. Next payload is 0
Apr  5 19:14:38.506: ISAKMP:(0):Checking ISAKMP transform 1 against priority 2 policy
Apr  5 19:14:38.506: ISAKMP:      encryption 3DES-CBC
Apr  5 19:14:38.506: ISAKMP:      hash MD5
Apr  5 19:14:38.506: ISAKMP:      default group 2
Apr  5 19:14:38.506: ISAKMP:      auth RSA sig
Apr  5 19:14:38.506: ISAKMP:      life type in seconds
Apr  5 19:14:38.506: ISAKMP:      life duration (VPI) of  0x0 0x1 0x51 0x80
Apr  5 19:14:38.506: ISAKMP:(0):atts are acceptable. Next payload is 3
```

```
Apr  5 19:14:38.506: ISAKMP:(0):Acceptable atts:actual life: 0
Apr  5 19:14:38.506: ISAKMP:(0):Acceptable atts:life: 0
Apr  5 19:14:38.506: ISAKMP:(0):Fill atts in sa vpi_length:4
Apr  5 19:14:38.506: ISAKMP:(0):Fill atts in sa life_in_seconds:86400
Apr  5 19:14:38.506: ISAKMP:(0):Returning Actual lifetime: 86400
Apr  5 19:14:38.506: ISAKMP:(0)::Started lifetime timer: 86400.

Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 69 mismatch
Apr  5 19:14:38.506: ISAKMP (0:0): vendor ID is NAT-T RFC 3947
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 245 mismatch
Apr  5 19:14:38.506: ISAKMP (0:0): vendor ID is NAT-T v7
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 157 mismatch
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID is NAT-T v3
Apr  5 19:14:38.506: ISAKMP:(0): processing vendor id payload
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
Apr  5 19:14:38.506: ISAKMP:(0): vendor ID is NAT-T v2
Apr  5 19:14:38.506: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:38.506: ISAKMP:(0):Old State = IKE_R_MM1  New State = IKE_R_MM1

Apr  5 19:14:38.506: ISAKMP:(0): constructed NAT-T vendor-rfc3947 ID
Apr  5 19:14:38.506: ISAKMP:(0): sending packet to 192.168.187.194 my_port 500 peer_port
500 (R) MM_SA_SETUP
Apr  5 19:14:38.506: ISAKMP:(0):Sending an IKE IPv4 Packet.
Apr  5 19:14:38.506: ISAKMP:(0):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Apr  5 19:14:38.506: ISAKMP:(0):Old State = IKE_R_MM1  New State = IKE_R_MM2

Apr  5 19:14:38.514: ISAKMP (0:0): received packet from 192.168.187.194 dport 500 sport
500 Global (R) MM_SA_SETUP
Apr  5 19:14:38.514: ISAKMP:(0):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:38.514: ISAKMP:(0):Old State = IKE_R_MM2  New State = IKE_R_MM3

Apr  5 19:14:38.514: ISAKMP:(0): processing KE payload. message ID = 0
Apr  5 19:14:38.518: ISAKMP:(0): processing NONCE payload. message ID = 0
Apr  5 19:14:38.518: ISAKMP:(13022): processing CERT_REQ payload. message ID = 0
Apr  5 19:14:38.522: ISAKMP:(13022): peer wants a CT_X509_SIGNATURE cert
Apr  5 19:14:38.522: ISAKMP:(13022): peer wants cert issued by cn=du-subca
Apr  5 19:14:38.522:  Choosing trustpoint du as issuer
Apr  5 19:14:38.522: ISAKMP:(13022): processing vendor id payload
Apr  5 19:14:38.522: ISAKMP:(13022): vendor ID is Unity
Apr  5 19:14:38.522: ISAKMP:(13022): processing vendor id payload
Apr  5 19:14:38.522: ISAKMP:(13022): vendor ID is DPD
Apr  5 19:14:38.522: ISAKMP:(13022): processing vendor id payload
Apr  5 19:14:38.522: ISAKMP:(13022): speaking to another IOS box!
Apr  5 19:14:38.522: ISAKMP:received payload type 20
Apr  5 19:14:38.522: ISAKMP (13022): His hash no match - this node outside NAT
Apr  5 19:14:38.522: ISAKMP:received payload type 20
Apr  5 19:14:38.522: ISAKMP (13022): No NAT Found for self or peer
Apr  5 19:14:38.522: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:38.522: ISAKMP:(13022):Old State = IKE_R_MM3  New State = IKE_R_MM3

Apr  5 19:14:38.522: ISAKMP (0:13022): constructing CERT_REQ for issuer cn=du-subca
Apr  5 19:14:38.522: ISAKMP (0:13022): constructing CERT_REQ for issuer cn=ra-subca
Apr  5 19:14:38.522: ISAKMP:(13022): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) MM_KEY_EXCH
Apr  5 19:14:38.522: ISAKMP:(13022):Sending an IKE IPv4 Packet.
Apr  5 19:14:38.522: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
Apr  5 19:14:38.522: ISAKMP:(13022):Old State = IKE_R_MM3  New State = IKE_R_MM4

Apr  5 19:14:38.562: ISAKMP (0:13022): received packet from 192.168.187.194 dport 500
sport 500 Global (R) MM_KEY_EXCH
Apr  5 19:14:38.562: ISAKMP:(13022):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
```

```
Apr  5 19:14:38.562: ISAKMP:(13022):Old State = IKE_R_MM4  New State = IKE_R_MM5

Apr  5 19:14:38.562: ISAKMP:(13022): processing ID payload. message ID = 0
Apr  5 19:14:38.562: ISAKMP (0:13022): ID payload
    next-payload : 6
    type         : 1
    address      : 192.168.187.194
    protocol     : 17
    port         : 500
    length       : 12
Apr  5 19:14:38.562: ISAKMP:(0):: peer matches *none* of the profiles
Apr  5 19:14:38.562: ISAKMP:(13022): processing CERT payload. message ID = 0
Apr  5 19:14:38.562: ISAKMP:(13022): processing a CT_X509_SIGNATURE cert
Apr  5 19:14:38.566: ISAKMP:(13022): peer's pubkey isn't cached
Apr  5 19:14:38.566: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.168.187.194
is bad: CA request failed!
Apr  5 19:14:38.566: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:38.566: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM5

Apr  5 19:14:38.566: ISAKMP (0:13022): incrementing error counter on sa, attempt 1 of 5:
reset_retransmission
Apr  5 19:14:38.566: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_ERROR
Apr  5 19:14:38.566: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM4

Apr  5 19:14:39.566: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH...
Apr  5 19:14:39.566: ISAKMP (0:13022): incrementing error counter on sa, attempt 2 of 5:
retransmit phase 1
Apr  5 19:14:39.566: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH
Apr  5 19:14:39.566: ISAKMP:(13022): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) MM_KEY_EXCH
Apr  5 19:14:39.566: ISAKMP:(13022):Sending an IKE IPv4 Packet.
Apr  5 19:14:40.066: ISAKMP (0:13022): received packet from 192.168.187.194 dport 500
sport 500 Global (R) MM_KEY_EXCH
Apr  5 19:14:40.066: ISAKMP:(13022):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:40.066: ISAKMP:(13022):Old State = IKE_R_MM4  New State = IKE_R_MM5

Apr  5 19:14:40.066: ISAKMP:(13022): processing CERT payload. message ID = 0
Apr  5 19:14:40.066: ISAKMP:(13022): processing a CT_X509_SIGNATURE cert
Apr  5 19:14:40.066: ISAKMP:(13022): peer's pubkey isn't cached
Apr  5 19:14:40.070: ISAKMP:(13022): Unable to get DN from certificate!
Apr  5 19:14:40.070: ISAKMP:(13022): Cert presented by peer contains no OU field.
Apr  5 19:14:40.070: ISAKMP:(0):: peer matches *none* of the profiles
Apr  5 19:14:40.070: ISAKMP (13022): adding peer's pubkey to cache
Apr  5 19:14:40.070: ISAKMP:(13022): processing SIG payload. message ID = 0
Apr  5 19:14:40.070: %CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.
Apr  5 19:14:40.070:  ISAKMP (0:13022): process_rsa_sig: Querying key pair failed.
Apr  5 19:14:40.070: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:40.070: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM5

Apr  5 19:14:40.070: ISAKMP (0:13022): incrementing error counter on sa, attempt 1 of 5:
reset_retransmission
Apr  5 19:14:40.070: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_ERROR
Apr  5 19:14:40.070: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM4

Apr  5 19:14:41.071: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH...
Apr  5 19:14:41.071: ISAKMP (0:13022): incrementing error counter on sa, attempt 2 of 5:
retransmit phase 1
Apr  5 19:14:41.071: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH
Apr  5 19:14:41.071: ISAKMP:(13022): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) MM_KEY_EXCH
Apr  5 19:14:41.071: ISAKMP:(13022):Sending an IKE IPv4 Packet.
Apr  5 19:14:41.571: ISAKMP (0:13022): received packet from 192.168.187.194 dport 500
sport 500 Global (R) MM_KEY_EXCH
Apr  5 19:14:41.571: ISAKMP:(13022):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
```

```
Apr  5 19:14:41.571: ISAKMP:(13022):Old State = IKE_R_MM4  New State = IKE_R_MM5

Apr  5 19:14:41.571: ISAKMP:(13022): processing CERT payload. message ID = 0
Apr  5 19:14:41.571: ISAKMP:(13022): processing a CT_X509_SIGNATURE cert
Apr  5 19:14:41.571: ISAKMP:(13022): peer's pubkey isn't cached
Apr  5 19:14:41.575: ISAKMP:(13022): Unable to get DN from certificate!
Apr  5 19:14:41.575: ISAKMP:(13022): Cert presented by peer contains no OU field.
Apr  5 19:14:41.575: ISAKMP:(0):: peer matches *none* of the profiles
Apr  5 19:14:41.575: ISAKMP (13022): adding peer's pubkey to cache
Apr  5 19:14:41.575: ISAKMP:(13022): processing SIG payload. message ID = 0
Apr  5 19:14:41.575: %CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.
Apr  5 19:14:41.575:  ISAKMP (0:13022): process_rsa_sig: Querying key pair failed.
Apr  5 19:14:41.575: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:41.575: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM5

Apr  5 19:14:41.575: ISAKMP (0:13022): incrementing error counter on sa, attempt 1 of 5:
reset_retransmission
Apr  5 19:14:41.575: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_ERROR
Apr  5 19:14:41.575: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM4

Apr  5 19:14:42.575: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH...
Apr  5 19:14:42.575: ISAKMP (0:13022): incrementing error counter on sa, attempt 2 of 5:
retransmit phase 1
Apr  5 19:14:42.575: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH
Apr  5 19:14:42.575: ISAKMP:(13022): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) MM_KEY_EXCH
Apr  5 19:14:42.575: ISAKMP:(13022):Sending an IKE IPv4 Packet.
Apr  5 19:14:43.075: ISAKMP (0:13022): received packet from 192.168.187.194 dport 500
sport 500 Global (R) MM_KEY_EXCH
Apr  5 19:14:43.075: ISAKMP:(13022):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:43.075: ISAKMP:(13022):Old State = IKE_R_MM4  New State = IKE_R_MM5

Apr  5 19:14:43.075: ISAKMP:(13022): processing CERT payload. message ID = 0
Apr  5 19:14:43.075: ISAKMP:(13022): processing a CT_X509_SIGNATURE cert
Apr  5 19:14:43.075: ISAKMP:(13022): peer's pubkey isn't cached
Apr  5 19:14:43.079: ISAKMP:(13022): Unable to get DN from certificate!
Apr  5 19:14:43.079: ISAKMP:(13022): Cert presented by peer contains no OU field.
Apr  5 19:14:43.079: ISAKMP:(0):: peer matches *none* of the profiles
Apr  5 19:14:43.079: ISAKMP (13022): adding peer's pubkey to cache
Apr  5 19:14:43.079: ISAKMP:(13022): processing SIG payload. message ID = 0
Apr  5 19:14:43.079: %CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.
Apr  5 19:14:43.079:  ISAKMP (0:13022): process_rsa_sig: Querying key pair failed.
Apr  5 19:14:43.079: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:43.079: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM5

Apr  5 19:14:43.079: ISAKMP (0:13022): incrementing error counter on sa, attempt 1 of 5:
reset_retransmission
Apr  5 19:14:43.079: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_ERROR
Apr  5 19:14:43.079: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM4

Apr  5 19:14:44.079: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH...
Apr  5 19:14:44.079: ISAKMP (0:13022): incrementing error counter on sa, attempt 2 of 5:
retransmit phase 1
Apr  5 19:14:44.079: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH
Apr  5 19:14:44.079: ISAKMP:(13022): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) MM_KEY_EXCH
Apr  5 19:14:44.079: ISAKMP:(13022):Sending an IKE IPv4 Packet.
Apr  5 19:14:44.579: ISAKMP (0:13022): received packet from 192.168.187.194 dport 500
sport 500 Global (R) MM_KEY_EXCH
Apr  5 19:14:44.579: ISAKMP:(13022):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:44.579: ISAKMP:(13022):Old State = IKE_R_MM4  New State = IKE_R_MM5

Apr  5 19:14:44.579: ISAKMP:(13022): processing CERT payload. message ID = 0
Apr  5 19:14:44.579: ISAKMP:(13022): processing a CT_X509_SIGNATURE cert
```

```
Apr  5 19:14:44.579: ISAKMP:(13022): peer's pubkey isn't cached
Apr  5 19:14:44.583: ISAKMP:(13022): Unable to get DN from certificate!
Apr  5 19:14:44.583: ISAKMP:(13022): Cert presented by peer contains no OU field.
Apr  5 19:14:44.583: ISAKMP:(0):: peer matches *none* of the profiles
Apr  5 19:14:44.583: ISAKMP (13022): adding peer's pubkey to cache
Apr  5 19:14:44.583: ISAKMP:(13022): processing SIG payload. message ID = 0
Apr  5 19:14:44.583: %CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.
Apr  5 19:14:44.583:  ISAKMP (0:13022): process_rsa_sig: Querying key pair failed.
Apr  5 19:14:44.583: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:44.583: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM5

Apr  5 19:14:44.583: ISAKMP (0:13022): incrementing error counter on sa, attempt 1 of 5:
reset_retransmission
Apr  5 19:14:44.583: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_ERROR
Apr  5 19:14:44.583: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM4

Apr  5 19:14:45.583: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH...
Apr  5 19:14:45.583: ISAKMP (0:13022): incrementing error counter on sa, attempt 2 of 5:
retransmit phase 1
Apr  5 19:14:45.583: ISAKMP:(13022): retransmitting phase 1 MM_KEY_EXCH
Apr  5 19:14:45.583: ISAKMP:(13022): sending packet to 192.168.187.194 my_port 500
peer_port 500 (R) MM_KEY_EXCH
Apr  5 19:14:45.583: ISAKMP:(13022):Sending an IKE IPv4 Packet.
Apr  5 19:14:46.083: ISAKMP (0:13022): received packet from 192.168.187.194 dport 500
sport 500 Global (R) MM_KEY_EXCH
Apr  5 19:14:46.083: ISAKMP:(13022):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
Apr  5 19:14:46.083: ISAKMP:(13022):Old State = IKE_R_MM4  New State = IKE_R_MM5

Apr  5 19:14:46.083: ISAKMP:(13022): processing CERT payload. message ID = 0
Apr  5 19:14:46.083: ISAKMP:(13022): processing a CT_X509_SIGNATURE cert
Apr  5 19:14:46.083: ISAKMP:(13022): peer's pubkey isn't cached
Apr  5 19:14:46.087: ISAKMP:(13022): Unable to get DN from certificate!
Apr  5 19:14:46.087: ISAKMP:(13022): Cert presented by peer contains no OU field.
Apr  5 19:14:46.087: ISAKMP:(0):: peer matches *none* of the profiles
Apr  5 19:14:46.087: ISAKMP (13022): adding peer's pubkey to cache
Apr  5 19:14:46.087: ISAKMP:(13022): processing SIG payload. message ID = 0
Apr  5 19:14:46.087: %CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.
Apr  5 19:14:46.087:  ISAKMP (0:13022): process_rsa_sig: Querying key pair failed.
Apr  5 19:14:46.087: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_MAIN_MODE
Apr  5 19:14:46.087: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM5

Apr  5 19:14:46.087: ISAKMP (0:13022): incrementing error counter on sa, attempt 1 of 5:
reset_retransmission
Apr  5 19:14:46.087: ISAKMP:(13022):Input = IKE_MESG_INTERNAL, IKE_PROCESS_ERROR
Apr  5 19:14:46.087: ISAKMP:(13022):Old State = IKE_R_MM5  New State = IKE_R_MM4
```

The preceding output example illustrates that the spoke was unable to establish a DMVPN tunnel with the headend.

# Troubleshooting PKI Deployment

This section presents troubleshooting content relevant to the following topics:


- Troubleshooting PKI, page 57
- Troubleshooting DMVPN with PKI, page 61

# Troubleshooting PKI

Suggestions for troubleshooting pertain to implementation of the PKI.

## Problem: Storage Location Not Accessible

**Symptom** The server is disabled.

**Possible Cause** Storage location is not accessible.

**Recommended Action** Make sure the external storage location, such as an FTP or HTTP server, is reachable and operational.

**Error Message** `Status: disabled, Storage not accessible`

The following procedure illustrates the process of resolving an inaccessible storage location problem.

**Step 1** To simulate the above problem, shut down the interface to the external storage for *du-subca*. By doing this, the server enters into that state. The server currently is active, operational as illustrated in the following **show** command output.

```
S-3825-du-subca# show crypto pki server
Certificate Server du-subca:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=du-subca
    CA cert fingerprint: A6298B11 A50948FF C170D745 CD7DFABC
    Server configured in subordinate server mode
    Upper CA cert fingerprint: ABD85DC7 C152AE90 4949A459 B91F0A39
    Granting mode is: manual
    Last certificate issued serial number (hex): 3
    CA certificate expiration timer: 16:06:24 EST Feb 12 2011
    CRL NextUpdate timer: 10:01:49 EST Apr 5 2009
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 90 days
    Autorollover timer: 16:06:14 EST Nov 14 2010
S-3825-du-subca#
```

**Step 2** Shut down the path to the external storage (an FTP server in this case). The interface connecting to external storage is gi0/0. You must also flap the server processes because the server will contact the external storage when it is issuing a new certificate or for a periodic CRL query. I n this case, we simulate the server contacting the external storage.

```
S-3825-du-subca(config)# interface gi
S-3825-du-subca(config)# interface gigabitEthernet 0/0
S-3825-du-subca(config-if)# shut
S-3825-du-subca(config-if)# end
S-3825-du-subca#
S-3825-du-subca(config)# crypto pki server du-subca
S-3825-du-subca(cs-server)# shut
Certificate server 'shut' event has been queued for processing.
S-3825-du-subca(cs-server)#
.Apr  5 13:58:17.524: %PKI-6-CS_DISABLED: Certificate server now disabled.
S-3825-du-subca(cs-server)# no shut
```

```
Certificate server 'no shut' event has been queued for processing.
S-3825-du-subca(cs-server)#
% There was a problem reading the file 'du-subca.ser'
%from certificate storage.
% Please verify storage accessibility
% and enable the server again.
S-3825-du-subca(cs-server)# end
S-3825-du-subca#
```

The server enters the disabled state because it is unable to reach the external storage.

**Step 3** To fix the problem, bring re-establish the path to the external storage server; flap the server process to make it reach the external location. The following commands illustrate these adjustments.

```
S-3825-du-subca(config)# interface gigabitEthernet 0/0
S-3825-du-subca(config-if)# no shut
S-3825-du-subca(config-if)#
.Apr  5 14:02:53.416: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to
reset
S-3825-du-subca(config-if)# end
S-3825-du-subca#
.Apr  5 14:02:56.680: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
.Apr  5 14:02:57.680: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0,
changed state to up
.Apr  5 14:02:57.972: %SYS-5-CONFIG_I: Configured from console by console
S-3825-du-subca# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
S-3825-du-subca(config)# crypto pki server du-subca
S-3825-du-subca(cs-server)# shut
Certificate server 'shut' event has been queued for processing.
S-3825-du-subca(cs-server)# no shut
Certificate server 'no shut' event has been queued for processing.
S-3825-du-subca(cs-server)# end
S-3825-du-subca#
.Apr  5 14:03:25.840: %SYS-5-CONFIG_I: Configured from console by console
Writing du-subca.crl !
Writing du-subca.crl !
Loading du-subca.ser
[OK - 32/4096 bytes]

Loading du-subca.crl
[OK - 218/4096 bytes]

.Apr  5 14:03:30.900: %PKI-6-CS_ENABLED: Certificate server now enabled
```

## Problem: Mismatched Subordinate CA Names

**Symptom** Certificate authority will not authenticate subordinate CA.

**Possible Cause** Mismatched subordinate CA trustpoint and enrollment names.

**Recommended Action** Fix the relevant configurations.

**Error Message** `% Failed to authenticate the Certificate Authority`

In order for the subordinate CA operate properly, the subordinate CA trustpoint and enrollment names must match. The following procedure illustrates finding this problem and making he appropriate adjustment.

**Step 1** Inspect the subordinate CA trustpoint name and enrollment name specifications in the configuration listing. They must match. The configuration listing that follows illustrates a problem.

```
 crypto pki server ra-subca
 database level complete
 database archive pkcs12 password 7 060506324F41584B56
 grant auto rollover ca-cert
 grant auto
 lifetime certificate 180
 lifetime ca-certificate 730
 mode sub-cs
 auto-rollover 90
 database url ftp://xxx.26.185.99
!
crypto pki trustpoint ra    ! <-- Mistake
 enrollment url http://10.254.0.10:80
 revocation-check crl
 rsakeypair ra-subca
```

**Step 2** With this error in your configuration, if you apply the **no shutdown** command, the subordinate CA generates the following error:

```
ra-subca(config)# crypto pki server ra-subca
ra-subca(cs-server)# no shut
%Some server settings cannot be changed after CA certificate generation.
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

Writing ra-subca.ser !% You must specify an enrollment URL for this CA before
you can authenticate it.

% Failed to authenticate the Certificate Authority
ra-subca(cs-server)# end
```

**Step 3** To fix this problem, make the following changes:

```
ra-subca(config)# no crypto pki trustpoint ra
% Removing an enrolled trustpoint will destroy all certificates
 received from the related Certificate Authority.

Are you sure you want to do this? [yes/no]: yes
% Be sure to ask the CA administrator to revoke your certificates.

No enrollment sessions are currently active.

ra-subca(config)# crypto pki trustpoint ra-subca
% You are not supposed to change the configuration of this
% trustpoint. It is being used by the IOS CA server.


ra-subca(config)# crypto pki server ra-subca
ra-subca(cs-server)# shut
Certificate server 'shut' event has been queued for processing.
ra-subca(cs-server)# exit
ra-subca(config)# crypto pki trustpoint ra-subca
ra-subca(ca-trustpoint)# enrollment url http://10.254.0.10:80
ra-subca(ca-trustpoint)# revocation-check crl
ra-subca(ca-trustpoint)# rsakeypair ra-subca
```

```
ra-subca(ca-trustpoint)# exit
ra-subca(config)# crypto pki server ra-subca
ra-subca(cs-server)# no shut
%Some server settings cannot be changed after CA certificate generation.

Certificate has the following attributes:
       Fingerprint MD5: ABD85DC7 C152AE90 4949A459 B91F0A39
      Fingerprint SHA1: 7754F54E A6547D55 182A6912 7F75EB6D DD1218E3

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.%
% 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: xxxxxx
Re-enter password: xxxxxxx


% Certificate request sent to Certificate Authority

% Enrollment in progress...
ra-subca(cs-server)#
*Jan 28 16:23:07.187: CRYPTO_PKI:  Certificate Request Fingerprint MD5: BFAF8350 A50AB8C6
364965C5 2CB5A996
*Jan 28 16:23:07.187: CRYPTO_PKI:  Certificate Request Fingerprint SHA1: 98A9CA2D 7FA2E275
5061735E 838FAD37 F104DE8F end
ra-subca# show cry
*Jan 28 16:23:10.019: %SYS-5-CONFIG_I: Configured from console by console
ra-subca# show crypto pki se
% Ambiguous command:  "show crypto pki se"
ra-subca#
ca-console# root-ca
Translating "root-ca"
Trying root-ca (1.1.1.1, 2034)... Open

S-3825-root-ca>
S-3825-root-ca> en
Password: xxxxxx
S-3825-root-ca# crypto pki server root-ca grant ?
  <1-999>  Request ID
  all      all pending requests

S-3825-root-ca# crypto pki server root-ca grant all
Writing 8.crt !
Writing 8.cnm !
Writing root-ca.ser !
S-3825-root-ca#
ca-console#1
[Resuming connection 1 to ra-subca ... ]

ra-subca#
ra-subca# show crypto pki server
Certificate Server ra-subca:
    Status: disabled, CA Certificate request is pending
    State: check failed
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=ra-subca
    CA cert fingerprint: -Not found-
    Server configured in subordinate server mode
    Upper CA cert fingerprint: ABD85DC7 C152AE90 4949A459 B91F0A39
    Granting mode is: auto
    Last certificate issued serial number (hex): 0
```

```
    CA certificate expiration timer: 20:00:00 EST Dec 31 1969
    CRL not present.
    Current primary storage dir: ftp://xxx.26.185.99
    Database Level: Complete - all issued certs written as <serialnum>.cer
    Auto-Rollover configured, overlap period 90 days
ra-subca#
Writing ra-subca.crl !
Writing ra-subca.crl !% Exporting Certificate Server signing certificate and keys...

Writing ra-subca_00002.p12 !
Loading ra-subca.ser
[OK - 32/4096 bytes]

Loading ra-subca.crl
[OK - 218/4096 bytes]

*Jan 28 16:24:10.515: %PKI-6-CERTRET: Certificate received from Certificate Authority
*Jan 28 16:24:10.839: %PKI-6-CS_ENABLED: Certificate server now enabled.
```

# Troubleshooting DMVPN with PKI

Suggestions for troubleshooting pertain to implementation of DMVPN with PKI.

## Problem: Certificate Invalid Due to Revocation or Expired Certificate

**Symptom** DMVPN tunnel is not being established.

**Possible Cause** Certificate invalid due to revocation or expired certificate.

**Recommended Action** Request a new certificate.

**Error Message** `%CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.`

To troubleshoot the above problem these are some of the debugs which could be turned ON at the router

```
s-dmvpn-headend# show debugging

PKI:
  Crypto PKI Msg debugging is on
  Crypto PKI Trans debugging is on
  Crypto PKI callbacks debugging is on
  Crypto PKI Validation Path debugging is on
s-dmvpn-headend#
Apr  6 16:18:33.909: %CRYPTO-3-IKMP_QUERY_KEY: Querying key pair failed.
Apr  6 16:18:36.469: CRYPTO_PKI: unlocked trustpoint du, refcount is 1
s-dmvpn-headend#
s-dmvpn-headend#u
Apr  6 16:18:56.346: CRYPTO_PKI: Trust-Point du picked up
Apr  6 16:18:56.346: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
Apr  6 16:18:56.346: CRYPTO_PKI: Found a subject match
Apr  6 16:18:56.346: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
Apr  6 16:18:56.346: CRYPTO_PKI: unlocked trustpoint du, refcount is 1
Apr  6 16:18:56.346: CRYPTO_PKI: locked trustpoint du, refcount is 2
Apr  6 16:18:56.386: CRYPTO_PKI: Adding peer certificate
Apr  6 16:18:56.386: CRYPTO_PKI: Added x509 peer certificate - (539) bytes
Apr  6 16:18:56.386: CRYPTO_PKI: crypto_pki_get_cert_record_by_subject()
```

```
Apr  6 16:18:56.386: CRYPTO_PKI: Found a subject match
Apr  6 16:18:56.386: CRYPTO_PKI: validation path has 1 certs

Apr  6 16:18:56.386: CRYPTO_PKI: Check for identical certs
Apr 6 16:18:56.386: CRYPTO_PKI(Cert Lookup) issuer="cn=du-subca" serial number= 04

Apr  6 16:18:56.386: CRYPTO_PKI: looking for cert in handle=736ECE8, digest=
 8F 0A C5 02 57 57 26 32 7C 81 D0 67 DC AB C4 DB

Apr  6 16:18:56.386: CRYPTO_PKI: Cert record not found, returning E_NOT_FOUND
Apr  6 16:18:56.386: CRYPTO_PKI: Create a list of suitable trustpoints
Apr  6 16:18:56.386: CRYPTO_PKI: crypto_pki_get_cert_record_by_issuer()
Apr  6 16:18:56.386: CRYPTO_PKI: Found a issuer match
Apr  6 16:18:56.386: CRYPTO_PKI: Suitable trustpoints are: du,
Apr  6 16:18:56.386: CRYPTO_PKI: Attempting to validate certificate using du
Apr  6 16:18:56.386: CRYPTO_PKI: Using du to validate certificate
Apr  6 16:18:56.386: CRYPTO_PKI(make trusted certs chain)
Apr  6 16:18:56.386: P11:C_CreateObject:
Apr  6 16:18:56.386:  CKA_CLASS: PUBLIC KEY
Apr  6 16:18:56.386:  CKA_KEY_TYPE: RSA
Apr  6 16:18:56.386:  CKA_MODULUS:
     AE 5D FC 46 E8 EB FC 20 8D ED F3 DA 50 9B 42 0C
     B5 B1 07 38 2A 5D 10 E3 90 C1 9E F5 0F 12 D0 DB
     7D 36 8C 79 23 6C 8F 64 56 3F 28 9B 4B 61 F0 41
     5B 14 BD 29 A2 A7 06 C8 67 10 3A ED 5F 2B 18 00
     7F C0 09 F3 1F 03 8B 5E E6 D4 D6 31 A6 9D 08 E0
     A8 A6 69 93 CB 3B 67 88 5C F5 83 DD 3E 71 6A EE
     45 EA E2 0E BB BA 78 AC 41 BC DE 19 64 92 F4 6B
     48 ED D5 4D 97 2B 57 AC 02 5C 66 A0 CB 7F 8D 91

Apr  6 16:18:56.390:  CKA_PUBLIC_EXPONENT:  01 00 01

Apr  6 16:18:56.390:  CKA_VERIFY_RECOVER:  01

Apr  6 16:18:56.390: P11:C_CreateObject: 139146472
Apr  6 16:18:56.390: P11:C_GetMechanismInfo slot 1 type 3 (invalid mechanism)
Apr  6 16:18:56.390: P11:C_GetMechanismInfo slot 1 type 1
Apr  6 16:18:56.390: P11:C_VerifyRecoverInit - 2711
Apr  6 16:18:56.390: P11:C_VerifyRecover - 2711
Apr  6 16:18:56.390: P11:found pubkey in cache using index = 2711
Apr  6 16:18:56.390: P11:public key found is :
     30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
     05 00 03 81 8D 00 30 81 89 02 81 81 00 AE 5D FC
     46 E8 EB FC 20 8D ED F3 DA 50 9B 42 0C B5 B1 07
     38 2A 5D 10 E3 90 C1 9E F5 0F 12 D0 DB 7D 36 8C
     79 23 6C 8F 64 56 3F 28 9B 4B 61 F0 41 5B 14 BD
     29 A2 A7 06 C8 67 10 3A ED 5F 2B 18 00 7F C0 09
     F3 1F 03 8B 5E E6 D4 D6 31 A6 9D 08 E0 A8 A6 69
     93 CB 3B 67 88 5C F5 83 DD 3E 71 6A EE 45 EA E2
     0E BB BA 78 AC 41 BC DE 19 64 92 F4 6B 48 ED D5
     4D 97 2B 57 AC 02 5C 66 A0 CB 7F 8D 91 02 03 01
     00 01

Apr  6 16:18:56.390: P11:CEAL:CRYPTO_NO_ERR
Apr  6 16:18:56.390: P11:C_DestroyObject 84C7868:A97
Apr  6 16:18:56.390: CRYPTO_PKI: Certificate is verified
Apr  6 16:18:56.390: CRYPTO_PKI: Checking certificate revocation
Apr  6 16:18:56.390: CRYPTO_PKI: Starting CRL revocation
Apr  6 16:18:56.390: CRYPTO_PKI: Select crl(cn=du-subca)
Apr  6 16:18:56.390: P11:C_CreateObject:
Apr  6 16:18:56.390:  CKA_CLASS: PUBLIC KEY
Apr  6 16:18:56.390:  CKA_KEY_TYPE: RSA
Apr  6 16:18:56.390:  CKA_MODULUS:
     AE 5D FC 46 E8 EB FC 20 8D ED F3 DA 50 9B 42 0C
```

```
      B5 B1 07 38 2A 5D 10 E3 90 C1 9E F5 0F 12 D0 DB
      7D 36 8C 79 23 6C 8F 64 56 3F 28 9B 4B 61 F0 41
      5B 14 BD 29 A2 A7 06 C8 67 10 3A ED 5F 2B 18 00
      7F C0 09 F3 1F 03 8B 5E E6 D4 D6 31 A6 9D 08 E0
      A8 A6 69 93 CB 3B 67 88 5C F5 83 DD 3E 71 6A EE
      45 EA E2 0E BB BA 78 AC 41 BC DE 19 64 92 F4 6B
      48 ED D5 4D 97 2B 57 AC 02 5C 66 A0 CB 7F 8D 91

Apr  6 16:18:56.390:  CKA_PUBLIC_EXPONENT:  01 00 01

Apr  6 16:18:56.390:  CKA_VERIFY_RECOVER:  01

Apr  6 16:18:56.390: P11:C_CreateObject: 139147368
Apr  6 16:18:56.390: P11:C_GetMechanismInfo slot 1 type 3 (invalid mechanism)
Apr  6 16:18:56.390: P11:C_GetMechanismInfo slot 1 type 1
Apr  6 16:18:56.390: P11:C_VerifyRecoverInit - 2712
Apr  6 16:18:56.390: P11:C_VerifyRecover - 2712
Apr  6 16:18:56.390: P11:found pubkey in cache using index = 2712
Apr  6 16:18:56.390: P11:public key found is :
      30 81 9F 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01
      05 00 03 81 8D 00 30 81 89 02 81 81 00 AE 5D FC
      46 E8 EB FC 20 8D ED F3 DA 50 9B 42 0C B5 B1 07
      38 2A 5D 10 E3 90 C1 9E F5 0F 12 D0 DB 7D 36 8C
      79 23 6C 8F 64 56 3F 28 9B 4B 61 F0 41 5B 14 BD
      29 A2 A7 06 C8 67 10 3A ED 5F 2B 18 00 7F C0 09
      F3 1F 03 8B 5E E6 D4 D6 31 A6 9D 08 E0 A8 A6 69
      93 CB 3B 67 88 5C F5 83 DD 3E 71 6A EE 45 EA E2
      0E BB BA 78 AC 41 BC DE 19 64 92 F4 6B 48 ED D5
      4D 97 2B 57 AC 02 5C 66 A0 CB 7F 8D 91 02 03 01
      00 01

Apr  6 16:18:56.394: P11:CEAL:CRYPTO_NO_ERR
Apr  6 16:18:56.394: P11:C_DestroyObject 84C7868:A98
Apr  6 16:18:56.394: ../crypto/ca/provider/revoke/crl/crlstat.c(205) : E_NOT_VALIDATED :
validation process failed (reason: 1)
Apr  6 16:18:56.394: CRYPTO_PKI: Certificate revoked
Apr  6 16:18:56.394: CRYPTO_PKI: Certificate validation failed
Apr  6 16:18:56.394: %CRYPTO-5-IKMP_INVAL_CERT: Certificate received from 192.168.187.194
is bad: CA request failed!
Apr  6 16:18:57.894: CRYPTO_PKI: Adding peer certificate
Apr 6 16:18:57.894: CRYPTO_PKI(Cert Lookup) issuer="cn=du-subca" serial number= 02

Apr  6 16:18:57.894: CRYPTO_PKI: looking for cert in handle=736ECE8, digest=
 82 83 D4 1E 27 D3 AF 97 F9 31 6B 24 36 09 48 47
```

## Problem: Clock not Synchronized or Certificate Expired

**Symptom** DMVPN tunnel not getting established.

**Possible Cause** The clock has not synchronized or the certificate has expired.

**Recommended Action** Synchronize the clocks if the certification is not yet expired.

**Error Message** `%PKI-3-CERTIFICATE_INVALID_NOT_YET_VALID: Certificate chain validation has failed.`

The following log messages illustrate the clock synchronization problem.

```
*Feb  1 06:20:00.095: CRYPTO_PKI: New CRL Not Yet Valid (router time not synched to CA?)
```

```
*Feb  1 06:20:00.095:    CRL published: 10:46:41 EST Apr 2 2009
*Feb  1 06:20:00.095:      Router time: 01:20:00 EST Feb 1 2002
*Feb  1 06:20:00.095: %PKI-4-CRLINSERTFAIL: Trustpoint "ra" unknown (error 1804:E_VALIDITY
: validity period start later than end)
*Feb  1 06:20:00.095: %PKI-3-CERTIFICATE_INVALID_NOT_YET_VALID: Certificate chain
validation has failed.
```