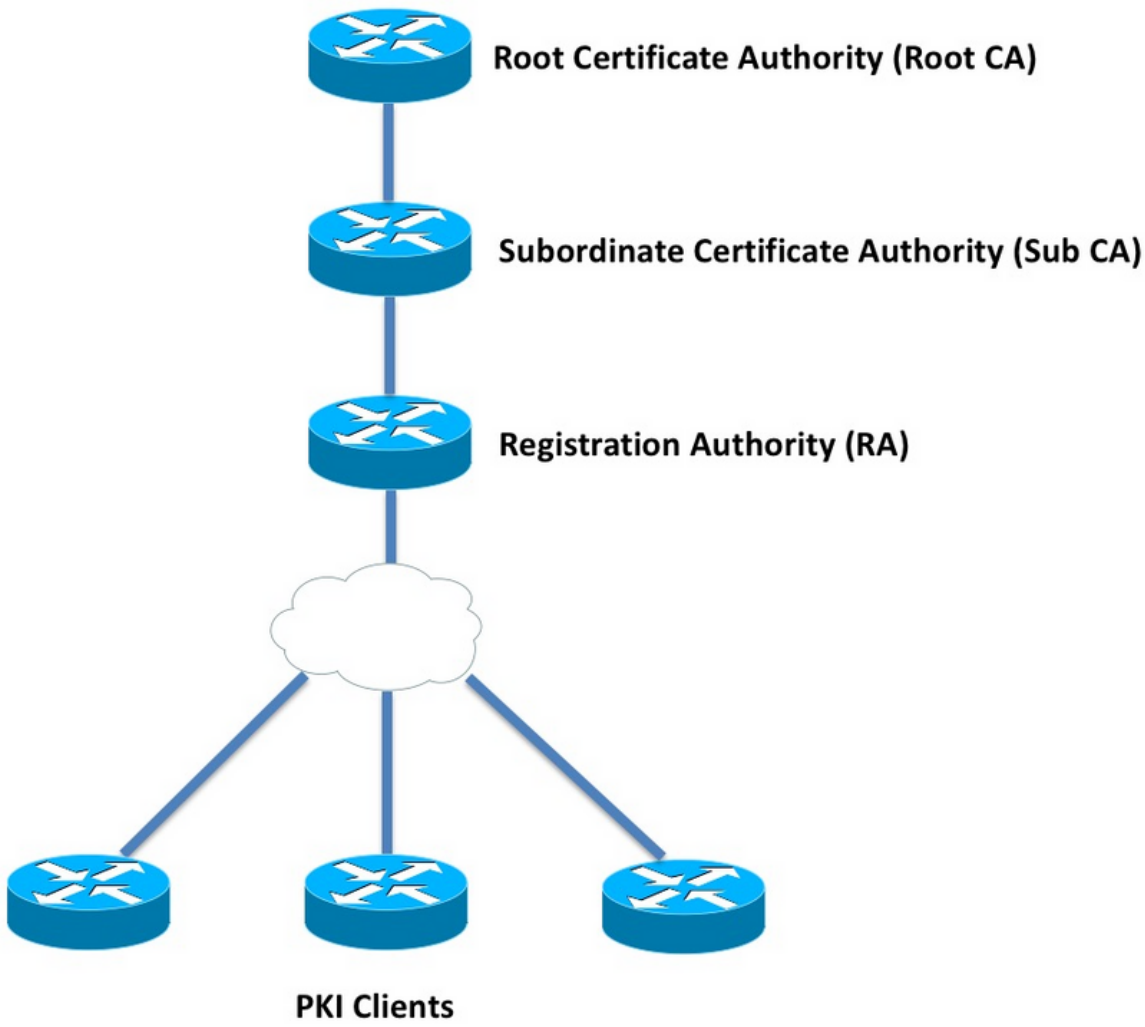
# IOS PKI部署指南：初始設計和部署

## 目錄

# 簡介

本文檔詳細介紹IOS PKI伺服器和客戶端功能。 它解決了IOS PKI初始設計和部署注意事項。

# PKI基礎設施




## 證書頒發機構

證書頒發機構(CA)，在文檔中也稱為PKI伺服器，是頒發證書的受信任實體。PKI以信任為基礎，信任層次結構從根證書頒發機構(Root-CA)開始。 因為根CA位於層次結構的頂端，所以它有一個自簽名的證書。

### 從屬證書頒發機構

在PKI Trust-hierarchy中，Root之下的所有證書頒發機構稱為從屬證書頒發機構(Sub-CA)。 顯然，子CA證書由CA頒發，其級別高於一級。

PKI不限制給定層次結構中的子CA的數量。但是，在具有3級以上證書頒發機構的企業部署中，可能難以管理。

### 註冊機構

PKI定義了一個特殊的證書頒發機構，稱為註冊機構(RA)，負責授權PKI客戶端註冊到給定的子CA或根CA。RA不向PKI客戶端頒發證書，而是決定哪個PKI客戶端可以或不能由子CA或根CA頒發證書。

RA的主要角色是從CA解除安裝基本客戶端證書請求驗證，並保護CA避免直接暴露給客戶端。這樣，RA就位於PKI客戶端和CA之間，從而保護CA免受任何型別的拒絕服務攻擊。

## PKI客戶端

任何基於駐留的公鑰 — 私鑰對請求證書以向其他裝置證明其身份的裝置，稱為PKI客戶端。

PKI客戶端必須能夠生成或儲存公鑰 — 私鑰對，例如RSA、DSA或ECDSA。

證書是給定公鑰的身份和有效性的證明，前提是裝置上存在相應的私鑰。

## IOS PKI伺服器

表1. IOS PKI伺服器功能的演變

| 功能 | IOS [ISR-G1、ISR-G2] | IOS-XE [ASR1K、ISR4K] |
|---|---|---|
| **IOS CA/PKI伺服器** | 12.3(4)公噸 | XE 3.14.0/15.5(1)S |
| **IOS PKI伺服器證書滾動更新** | 12.4(1)公噸 | XE 3.14.0/15.5(1)S |
| **IOS PKI HA** | 15.0(1)米 | NA [提供隱式RP間冗餘] |
| **適用於第三方CA的IOS RA** | 15.1(3)公噸 | XE 3.14.0/15.5(1)S |

在進入PKI伺服器配置之前，管理員必須瞭解這些核心概念。

### 權威時間來源

PKI基礎設施的基礎之一是時間。系統時鐘定義證書是否有效。因此，在IOS中，時鐘必須具有權威性或可信性。如果沒有權威的時間來源，PKI伺服器可能無法按預期運行，強烈建議使用以下方法使IOS上的時鐘具有權威性：

#### NTP（網路時間協定）

將系統時鐘與時間伺服器同步是使系統時鐘可信的唯一真正方法。IOS路由器可以配置為網路中公認且穩定的NTP伺服器的NTP客戶端：

```
configure terminal
ntp server <NTP Server IP address>
ntp source <source interface name>
ntp update-calendar

!! optional, if the NTP Server requires the clients to authenticate themselves
ntp authenticate
ntp authentication-key 1 md5 <key>

!! optionally an access-list can be configured to restrict time-updates from a specifc NTP
server
access-list 1 permit <NTP Server IP address>
ntp access-group peer 1
```

IOS還可以配置為NTP伺服器，它將本地系統時鐘標籤為授權時鐘。在小型PKI部署中，PKI伺服器可配置為其PKI客戶端的NTP伺服器：

```
configure terminal
ntp master <stratum-number>

!! optionally, NTP authentication can be enforced
ntp authenticate
ntp authentication-key 1 md5 <key-1>
ntp authentication-key 2 md5 <key-2>
ntp authentication-key 2 md5 <key-2>
ntp trusted-key 1 - 3

!! optinally, an access-list can be configured to restrict NTP clients
!! first allow the local router to synchronize with the local time-server
access-list 1 permit 127.127.7.1
ntp access-group peer 1

!! define an access-list to which the local time-server will serve time-synchronization services
access-list 2 permit <NTP-Client-IP>
ntp access-group serve-only 2
```

**將硬體時鐘標籤為可信**

在IOS中，可以使用以下命令將硬體時鐘標籤為授權時鐘：

```
config terminal
clock calendar-valid
```

這可以與NTP一起配置，這樣做的主要原因是當路由器重新載入（例如由於斷電）時，以及NTP伺服器不可訪問時，保持系統時鐘的權威。在這個階段，PKI計時器將停止工作，這進而導致證書續訂/翻轉失敗。**clock calendar-valid在此類**情況下充當安全保障。

在配置此項時，關鍵是要瞭解如果系統電池電量耗盡，系統時鐘將不同步，PKI將開始信任不同步時鐘。但是，與完全沒有權威的時間源相比，配置這一點相對更安全。

**附註**：**clock calendar-valid命**令是在IOS-XE 3.10.0/15.3(3)S版中新增的。

### 主機名和域名

建議首先在Cisco IOS上配置主機名和域名，然後再配置任何與PKI相關的服務。路由器主機名和域名在以下情況下使用：

- 預設RSA金鑰對名稱通過組合主機名和域名而得出
- 註冊證書時，預設的subject-name由hostname屬性和unstructured-name組成，後者是hostname和domain-name之和。

對於PKI伺服器，未使用主機名和域名：

- 預設金鑰對名稱將與PKI伺服器名稱相同
- 預設的Subject-name由CN組成，與PKI伺服器名稱相同。

一般建議是配置合適的主機名和域名。

```
config terminal
hostname <string>
ip domain name <domain>
```

## HTTP伺服器

IOS PKI Server僅在啟用HTTP Server時啟用。必須注意的是，如果由於HTTP伺服器被禁用而禁用了PKI伺服器，則它可以繼續通過[終端]授予離線請求。處理SCEP請求和傳送SCEP響應需要HTTP伺服器功能。

IOS HTTP Server使用以下命令啟用：

```
ip http server
```

可以使用以下方法將預設HTTP伺服器埠從80更改為任何有效的埠號：

```
ip http port 8080
```

#### HTTP最大連線

在使用SCEP將IOS部署為PKI伺服器時，瓶頸之一是最大併發HTTP連線數和平均HTTP連線數（每分鐘）。
目前，預設情況下，IOS HTTP伺服器上的最大併發連線數限製為5，並且可以增加到16，這在中等規模部署中是強烈建議的：

```
ip http max-connections 16
```

此IOS安裝最多允許1000個併發HTTP連線：

- 採用uck9許可證集的UniversalK9 IOS

CLI會自動更改，以接受介於1和1000之間的數字引數

```
ip http max-connections 1000
```

IOS HTTP伺服器允許每分鐘80個連線[在最大HTTP併發會話可以增加到1000的IOS版本中為580]，並且在一分鐘內達到此限制時，IOS HTTP監聽器通過關閉監聽程式15秒開始限制傳入的HTTP連線。這會導致客戶端連線請求因達到TCP連線隊列**限制而被丟棄**。有關這方面的更多資訊，請參閱

## RSA金鑰對

IOS上PKI伺服器功能的RSA金鑰對可以自動生成或手動生成。
配置PKI伺服器時，IOS自動建立與PKI伺服器同名的信任點，以便儲存PKI伺服器證書。

### 手動生成PKI伺服器RSA金鑰對：

步驟1.建立與PKI伺服器同名的RSA金鑰對：

```
crypto key generate rsa general-keys label <LABEL> modulus 2048
```

步驟2.在啟用PKI伺服器之前，修改PKI伺服器信任點：

```
crypto pki trustpoint <PKI-SERVER-Name>
  rsakeypair <LABEL>
```

> **附註**：在IOS版本15.4(3)M4之前，不會考慮在PKI伺服器信任點下提到的RSA金鑰對係數值，這是已知警告。預設金鑰係數為1024位。

### 自動生成PKI伺服器RSA金鑰對：

啟用PKI伺服器時，IOS會自動生成與PKI伺服器同名的RSA金鑰對，金鑰係數大小為1024位。

從IOS版本15.4(3)M5開始，此配置將建立一個以<LABEL>為名稱的RSA金鑰對，並且金鑰強度將按定義的<MOD>模數確定。

```
crypto pki trustpoint <PKI-SERVER-Name>
rsakeypair <LABEL> <MOD>
```

擾流器

CSCuu73408 IOS PKI伺服器應允許使用非預設金鑰大小來進行滾動驗證。
CSCuu73408 IOS PKI伺服器應允許使用非預設金鑰大小來進行滾動更新證書。

目前的行業標準是使用2048位的RSA金鑰對。

## 自動滾動更新計時器注意事項

目前，IOS PKI Server在預設情況下不生成翻轉證書，並且必須在PKI伺服器下使用**auto-rollover <days-before-expiry>**命令顯式啟用該證書。有關證書滾動的詳細資訊，請參閱

此命令指定如果IOS建立滾動更新CA證書，PKI伺服器/CA證書到期前多長時間。請注意，在當前活動CA證書過期後，將啟用滾動更新CA證書。當前預設值為30天。根據CA證書生存期，應將此值設定為合理的值，這進而影響PKI客戶端上的自動註冊計時器配置。

> **附註**：在CA和客戶端證書滾動期間，在客戶端上自動註冊計時器之前，應始終觸發自動滾動計時器[稱為]

## CRL注意事項

IOS PKI基礎設施支援兩種分發CRL的方法：

### 將CRL發佈到HTTP伺服器

IOS PKI Server可以在PKI Server下使用以下命令將CRL檔案發佈到HTTP伺服器上的特定位置：

```
crypto pki server <PKI-SERVER-Name>
 database crl publish <URL>
```

並且，在PKI伺服器下使用以下命令可以將PKI伺服器配置為將此CRL位置嵌入到所有PKI客戶端證書中：

```
crypto pki server <PKI-SERVER-Name>
cdp-url <CRL file location>
```

### SCEP GetCRL方法

IOS PKI Server自動將CRL檔案儲存在特定的資料庫位置（預設為nvram），強烈建議在PKI Server下使用以下命令在SCP/FTP/TFTP伺服器上保留副本：

```
crypto pki server <PKI-SERVER-Name>
database url <URL>
or
database crl <URL>
```

預設情況下，IOS PKI伺服器不會將CDP位置嵌入到PKI客戶端證書中。如果將IOS PKI客戶端配置為執行撤銷檢查，但正在驗證的證書沒有內建CDP，並且已使用CA位置配置驗證CA信任點(使用http://<CA-Server-IP或FQDN>)，則預設情況下IOS將回退到基於SCEP的GetCRL方法。
SCEP GetCRL通過對此URL執行HTTP GET來執行CRL檢索：

```
http:://<CA-Server-IP/FQDN>/cgi-bin/pkiclient.exe?operation=GetCRL
```

**附註**：在IOS CLI中，輸入**?**之前，按**Ctrl + V**組合鍵。

IOS PKI Server還可以將此URL嵌入為CDP位置。這樣做的好處有兩方面：

- 它確保所有非IOS SCEP型PKI客戶端都可以執行CRL檢索。
- 如果沒有嵌入式CDP，則會按照SCEP草案中的定義對IOS SCEP GetCRL請求消息進行簽名（使用臨時自簽名證書）。但是，CRL檢索請求不需要簽名，並且通過為GetCRL方法嵌入CDP URL，可以避免對CRL請求進行簽名。

## CRL的生存期

在PKI伺服器下使用以下命令可以控制IOS PKI伺服器的CRL生存期：

```
crypto pki server <PKI-SERVER-Name>
 lifetime crl <0 - 360>
```

值以小時為單位。預設情況下，CRL的生命週期設定為6小時。根據撤銷證書的頻率，將CRL生存期調整為最佳值可以提高網路中的CRL檢索效能。

## 資料庫注意事項

IOS PKI伺服器使用nvram作為預設資料庫位置，強烈建議使用FTP、TFTP或SCP伺服器作為資料庫位置。預設情況下，IOS PKI伺服器建立兩個檔案：

- <Server-Name>.ser — 它包含CA以十六進位制形式發出的最後序列號。檔案採用純文字檔案格式，包含以下資訊：
  db_version = 1
  last_serial = 0x4
- <Server-Name>.crl — 這是CA發佈的DER編碼的CRL檔案

IOS PKI Server在資料庫中以3個可配置級別儲存資訊：

- 最小值 — 這是預設級別，在此級別不會在資料庫中建立任何檔案，因此CA伺服器上沒有有關過去授予的客戶端證書的資訊。
- 名稱 — 在此級別，IOS PKI伺服器為每個頒發的客戶端證書建立一個名為<Serial-Number>.cnm的檔案，其中名稱<Serial-Number>是指頒發的客戶端證書的序列號。此cnm檔案包含主題名稱和客戶端證書的到期日期。
- 完成 — 在此級別，IOS PKI伺服器為每個頒發的客戶端證書建立兩個檔案：
- <Serial-Number>.cnm
- <Serial-Number>.crt

這裡，crt檔案是客戶端證書檔案，它採用DER編碼。

以下幾點很重要：

- 在頒發客戶端證書之前，IOS PKI Server會引用<Server-Name>.ser來確定和派生證書的序列號。
- 如果資料庫級別設定為「名稱」或「完成」，則在將已授予/已頒發的證書傳送到客戶端之前，需要將<Serial-Number>.cnm和<Serial-Number>.crt寫入資料庫
- 如果資料庫URL設定為「名稱」或「完整」，則資料庫URL必須擁有足夠的空間來儲存檔案。因此，建議將外部檔案伺服器[FTP或TFTP或SCP]配置為資料庫URL。
- 配置外部資料庫URL後，絕對有必要確保在證書授予過程中可以訪問檔案伺服器，否則會將CA伺服器標籤為禁用。並且需要手動干預，才能使CA伺服器重新聯機。

## 資料庫存檔

部署PKI伺服器時，必須考慮故障場景，並在出現硬體故障時做好準備。實現這一點有兩種方式：

1. 備援
   在這種情況下，兩個裝置或處理單元將充當主用 — 備用裝置以提供冗餘。
   IOS PKI伺服器的高可用性可通過使用兩個啟用HSRP的ISR路由器[ISR G1和ISR G2]來實現，如中所述

   基於IOS XE的系統[ISR4K和ASR1k]沒有可用的裝置冗餘選項。但是，在ASR1k中，RP間冗餘預設可用。
2. 存檔CA伺服器金鑰對和檔案
   IOS提供存檔PKI伺服器金鑰對和證書的功能。可使用兩種型別的檔案完成歸檔：
   PEM - IOS建立PEM格式的檔案以儲存RSA公鑰、加密的RSA私鑰和CA伺服器證書。滾動更新金鑰對和證書將自動存檔PKCS12 - IOS建立一個PKCS12檔案，該檔案包含CA伺服器證書和使用密碼加密的相應RSA私鑰。

可以在PKI伺服器下使用以下命令啟用資料庫存檔：

```
crypto pki server <PKI-SERVER-Name>
 database archive {pkcs12 | pem} password <password>
```

也可以將歸檔檔案儲存到單獨的伺服器，可能在PKI伺服器下使用下列命令使用安全協定(SCP):

```
crypto pki server <PKI-SERVER-Name>
 database url {p12 | pem} <URL>
```

除已存檔檔案和.Ser檔案外，資料庫中的所有其他檔案都是明文檔案，如果檔案丟失也不會造成真正的威脅，因此可以儲存在單獨的伺服器上，而不會在寫入檔案時（例如TFTP伺服器）產生大量開銷。

## IOS as Sub-CA

預設情況下，IOS PKI伺服器承擔根CA的角色。要配置從屬PKI伺服器（子CA），請先在PKI伺服器配置部分啟用此命令（在啟用PKI伺服器之前）：

```
crypto pki server <Sub-PKI-SERVER-Name>
 mode sub-cs
```

使用此命令在PKI伺服器的信任點下配置根CA的URL:

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
 enrollment url <Root-CA URL>
```
啟用此PKI伺服器現在將觸發以下事件：

- PKI Server trustpoint通過身份驗證以安裝根CA證書。
- 在根CA通過驗證後，IOS會為從屬CA [x509基本約束包含CA:TRUE標誌]生成CSR，並將其傳送到根CA

無論在根CA上配置何種授權模式，IOS都會將CA（或RA）證書請求放入掛起隊列。管理員必須手動授予CA證書。
要檢視待處理的證書請求和請求ID，請執行以下操作：

```
show crypto pki server <Server-Name> requests
```

要批准請求，請執行以下操作：

```
crypto pki server <Server-Name> grant <request-id>
```

- 使用此，後續的SCEP POLL(GetCertInitial)操作將下載子CA證書並將其安裝在路由器上，從而啟用從屬PKI伺服器

### IOS as RA

IOs PKI伺服器可以配置為給定從屬CA或根CA的註冊授權。要將PKI伺服器配置為註冊機構，請首先在PKI伺服器配置部分下啟用此命令（在啟用PKI伺服器之前）：

```
crypto pki server <RA-SERVER-Name>
 mode ra
```
然後，在PKI伺服器的信任點下配置CA的URL。這表示哪個CA受RA保護：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
註冊機構不會頒發證書，因此不需要在RA下配置**issuer-name**，即使進行了配置，該配置也無效。在RA信任點下使用**subject-name**命令配置RA的subject-name。將**OU = ioscs RA**配置為主題名稱的一部分非常重要，這樣IOS CA才能識別IOS RA，即識別IOS RA授權的證書請求。

IOS可以充當第三方CA（例如Microsoft CA）的註冊授權，而且為了保持相容，必須在PKI伺服器配置部分下使用以下命令啟用IOS RA（在啟用PKI伺服器之前）：

```
mode ra transparent
```
在預設RA模式下，IOS使用RA證書對客戶端請求[PKCS#10]進行簽名。此操作指示IOS PKI伺服器證書請求已由RA授權。

在透明RA模式下，IOS以原始格式轉發客戶端請求而不引入RA證書，並且這與Microsoft CA相容（如已知示例）。

## IOS PKI使用者端

IOS PKI客戶端中最重要的配置實體之一是信任點。本節詳細介紹信任點配置引數。

## 權威時間來源

如前所述，權威的時間來源也是PKI客戶端的要求。可以使用以下配置將IOS PKI客戶端配置為NTP客戶端：

```
configure terminal
ntp server <NTP Server IP address>
ntp source <source interface name>
ntp update-calendar

!! optional, if the NTP Server requires the clients to authenticate themselves
ntp authenticate
ntp authentication-key 1 md5 <key>

!! Optionally an access-list can be configured to restrict time-updates from a specifc NTP
server
access-list 1 permit <NTP Server IP address>
ntp access-group peer 1
```

## 主機名和域名

一般建議在路由器上配置主機名和域名：

```
configure terminal
hostname <string>
ip domain name <domain>
```

## RSA金鑰對

在IOS PKI客戶端中，指定信任點註冊的RSA金鑰對可以自動生成或手動生成。

自動生成RSA金鑰的過程涉及以下內容：

- 預設情況下，IOS會建立512位RSA金鑰對
- 自動生成的金鑰對名稱為hostname.domain-name ，它是裝置主機名與裝置域名組合
- 自動生成的金鑰對未標籤為可匯出。

自動生成RSA金鑰的過程涉及以下內容：

- 或者，可以使用以下方法手動生成具有適當強度的通用RSA金鑰對：
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]`
  此處，LABEL - RSA金鑰對名稱
  MOD - RSA金鑰模數或強度在360至4096之間，傳統上為512、1024、2048或4096。
  手動生成RSA金鑰對的優點是可以將金鑰對標籤為可匯出，這反過來允許完全匯出身份證書，然後在另一台裝置上恢復。但是，人們應該理解這一行動的安全影響。
- 使用此命令註冊之前，RSA金鑰對連結到信任點

```
crypto pki trustpoint MGMT
rsakeypair <LABEL> [<MOD> <MOD>]
```

此處，如果名為<LABEL>的RSA金鑰對已存在，則會在信任點註冊期間拾取該金鑰對。
如果名為<LABEL>的RSA金鑰對不存在，則在註冊期間執行以下操作之一：
— 如果未傳遞<MOD>引數，則會生成名為<LABEL>的512位金鑰對。
— 如果傳遞了一個<MOD>引數，則會生成名為<LABEL>的<MOD>位通用金鑰對
— 如果傳遞兩個<MOD>引數，則生成一個<MOD>位簽名金鑰對和一個<MOD>位加密金鑰對，兩個都命名為<LABEL>

## 信任點

信任點是在IOS中儲存證書的抽象容器。單個信任點能夠在任何給定時間儲存兩個活動證書：

- CA證書 — 將CA證書載入到給定信任點稱為信任點身份驗證過程。
- 由CA頒發的ID證書 — 將ID證書載入或匯入到給定信任點稱為信任點註冊過程。

信任點配置稱為信任策略，它定義了：

- 信任點中載入了哪個CA證書？
- 信任點註冊到哪個CA?
- IOS如何註冊信任點？
- 如何驗證由給定CA [載入到信任點中]頒發的證書？

此處介紹信任點的主要組成部分。

### 註冊模式

信任點註冊模式（也定義信任點身份驗證模式）可以通過3種主要方法執行：

1. 終端註冊 — 在CLI終端中使用複製貼上手動執行信任點身份驗證和證書註冊的方法。
2. SCEP註冊 — 使用SCEP over HTTP的信任點身份驗證和註冊。
3. 註冊配置檔案 — 在這裡，身份驗證和註冊方法單獨定義。與終端和SCEP註冊方法一樣，註冊配置檔案也提供了指定HTTP/TFTP命令的選項，以便從伺服器執行檔案檢索（使用配置檔案下的身份驗證或註冊URL定義）。

### 來源介面和VRF

通過HTTP(SCEP)或TFTP（註冊配置檔案）的信任點身份驗證和註冊使用IOS檔案系統執行檔案i/o操作。這些資料包交換可以來自特定的源介面和VRF。

在傳統信任點配置的情況下，在信任點下使用**source interface**和**vrf**子命令啟用此功能。

在註冊配置檔案的情況下，**源接**口和註**冊 | authentication url <http/tftp://Server-location> vrf <vrf-name>命令提**供相同的功能。

配置示例：

```
vrf definition MGMT
 rd 1:1
 address-family ipv4
 exit-address-family

crypto pki trustpoint MGMT
```

```
 source interface Ethernet0/0
 vrf MGMT
```
或

```
crypto pki profile enrollment MGMT-Prof
 enrollment url http://10.1.1.1:80 vrf MGMT
 source-interface Ethernet0/0
crypto pki trustpoint MGMT
 enrollment profile MGMT-Prof
```

### 自動證書註冊和續訂

IOS PKI客戶端可以在PKI trustpoint部分下使用以下命令執行自動註冊和續訂：

```
crypto pki trustpoint MGMT
auto-enroll <percentage> <regenerate>
```
此處，**auto-enroll <percentage> [regenerate]**命令說明IOS應在當前證書生存期的80%準確執行證書續訂。

關鍵字**regenerate**表明，IOS應該在每次證書續訂操作期間重新生成名為shadow key-pair的RSA金鑰對。

以下是自動註冊行為：

- 配置**自動註冊**時，如果信任點經過身份驗證，IOS將在PKI信任點部分或註冊配置檔案下，對位於作為**enrollment url**命令一部分提及的URL處的伺服器執行自動註冊。
- 當信任點註冊到PKI伺服器或CA時，在PKI客戶端上根據信任點下安裝的當前身份證書的**自動註冊百分**比，初始化RENEW或SHADOW計時器。此計時器在**show crypto pki timer**命令下可見。有關計時器功能的詳細*資訊，請參閱*
- 更新功能支援來自PKI伺服器。有關此問題的詳細資訊，請參閱
  IOS PKI客戶端執行兩種型別的續訂：
  隱式續訂：如果PKI伺服器不將「續訂」作為支援的功能傳送，則IOS會以定義的自動註冊百分比執行初始註冊。即IOS使用自簽名證書來簽署續訂請求。顯式續訂：當PKI伺服器支援PKI客戶端證書續訂功能時，它將「續訂」通告為受支援的功能。IOS在證書續訂期間會考慮此功能，即IOS使用當前活動的身份證書來簽署續訂證書請求。

設定自動註冊百分比時請務必小心。在部署中的任何特定PKI客戶端上，如果出現身份證書與頒發CA證書同時到期的情況，則自動註冊值應始終在CA建立滾動證書後觸發[shadow]續訂操作。請參閱中的**PKI計時器依賴性***部分*

### 證書撤銷檢查

經過身份驗證的PKI信任點（即包含CA證書的PKI信任點）能夠在IKE或SSL協商期間執行證書驗證，其中，對等證書經過徹底的證書驗證。驗證方法之一是使用以下兩種方法之一檢查對等證書吊銷狀態：

- 證書吊銷清單(CRL) — 此檔案包含由給定CA吊銷的證書的序列號。此檔案使用頒發的CA證書簽名。CRL方法涉及使用HTTP或LDAP下載CRL檔案。

- 線上證書狀態協定(OCSP)- IOS與名為OCSP響應器的實體建立通訊通道，該實體是頒發CA指

定的伺服器。客戶端（例如IOS）傳送包含正在驗證的證書的序列號的請求。OCSP響應方以給定序列號的吊銷狀態進行響應。可以使用任何受支援的應用/傳輸協定（通常是HTTP）建立通訊通道。

可以在PKI信任點部分下使用以下命令定義撤銷檢查：

```
crypto pki trustpoint MGMT
 revocation-check crl ocsp none
```

預設情況下，信任點配置為使用crl執行撤銷檢查。

可以對這些方法進行重新排序，並且按定義的順序執行撤銷狀態檢查。方法「none」繞過撤銷檢查。

### CRL快取

使用基於CRL的撤銷檢查，每個證書驗證都可以觸發新的CRL檔案下載。而且，由於CRL檔案變大或者如果CRL分發點(CDP)更遠，在每次驗證過程中下載檔案都會影響依賴於證書驗證的協定的效能。因此，執行CRL快取以提高效能，並且快取CRL考慮到CRL的有效性。

CRL有效性使用兩個時間引數定義：**LastUpdate**（發出CA上次發佈CRL的時間）和**NextUpdate**（發出CA將來發佈新版本的CRL檔案）。

只要CRL有效，IOS就會快取每個下載的CRL。但是，在某些情況下（例如CDP暫時無法訪問），可能需要在快取中保留CRL較長時間。在IOS中，快取的CRL可在CRL有效期到期後24小時內保留，並且可以在PKI信任點部分下使用以下命令進行配置：

```
crypto pki trustpoint MGMT
 crl cache extend <0 - 1440>
!! here the value is in minutes
```

在某些情況下，例如在CRL有效期之內發出CA撤銷證書，IOS可以配置為更頻繁地刪除快取。通過過早刪除CRL，IOS被迫更頻繁地下載CRL以使CRL快取保持最新。此配置選項在PKI信任點部分下可用：

```
crypto pki trustpoint MGMT
 crl cache delete-after <1-43200>
!! here the value is in minutes
```

最後，在PKI信任點部分下使用此命令可以將IOS配置為不快取CRL檔案：

```
crypto pki trustpoint MGMT
 crl cache none
```

# 建議的配置

以下是具有根CA和子CA配置的典型CA部署。示例還包括受RA保護的子CA配置。

對於2048位的RSA金鑰對，此示例建議進行以下設定：
Root-CA的使用壽命為8年
Sub-CA的生命週期為3年
客戶端證書頒發一年，配置為自動請求證書續訂。

## 根CA — 配置

```
crypto pki server ROOTCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=RootCA,OU=TAC,O=Cisco
lifetime crl 120
lifetime certificate 1095
lifetime ca-certificate 2920
grant auto rollover ca-cert
auto-rollover 85
database url ftp:://10.1.1.1/CA/ROOT/
database url crl ftp:://10.1.1.1/CA/ROOT/
database url crl publish ftp:://10.1.1.1/WWW/CRL/ROOT/
cdp-url http://10.1.1.1/WWW/CRL/ROOT/ROOTCA.crl
```

## 不帶RA的SUBCA — 配置

```
crypto pki server SUBCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=SubCA,OU=TAC,O=Cisco
lifetime crl 12
lifetime certificate 365
grant auto SUBCA
auto-rollover 85
database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
cdp-url http://10.1.1.1/WWW/CRL/SUB/SUBCA.crl
mode sub-cs
```

```
crypto pki trustpoint SUBCA
revocation-check crl
rsakeypair SUBCA 2048
enrollment url http://172.16.1.1
```

## 含RA的SUBCA — 組態

```
crypto pki server SUBCA
database level complete
database archive pkcs12 password p12password
issuer-name CN=SubCA,OU=TAC,O=Cisco
lifetime crl 12
lifetime certificate 365
grant ra-auto
grant auto rollover ra-cert
auto-rollover 85
 database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
cdp-url http://10.1.1.1/WWW/CRL/SUB/SUBCA.crl
mode sub-cs

crypto pki trustpoint SUBCA
revocation-check crl
rsakeypair SUBCA 2048
enrollment url http://172.16.1.1
```

## SUBCA的RA — 配置

```
crypto pki server RA-FOR-SUBCA
database level complete
database archive pkcs12 password p12password
mode ra
grant auto RA-FOR-SUBCA
auto-rollover 85
database url ftp:://10.1.1.1/CA/RA4SUB/


crypto pki trustpoint RA-FOR-SUBCA
enrollment url http://172.16.1.2:80
password ChallengePW123
subject-name CN=RA,OU=ioscs RA,OU=TAC,O=Cisco
revocation-check crl
rsakeypair RA 2048
```

# 證書註冊

## 手動註冊

手動註冊涉及在PKI客戶端上生成離線CSR，並手動將其複製到CA。管理員手動簽署請求，然後將其匯入客戶端。

### PKI客戶端

PKI客戶端配置：

```
crypto pki trustpoint MGMT
enrollment terminal
serial-number
ip-address none
password ChallengePW123
subject-name CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
revocation-check crl
rsakeypair PKI-Key
```

步驟1.首先驗證信任點（這也可以在步驟2之後執行）。

```
crypto pki authenticate MGMT
!! paste the CA, in this case the SUBCA, certificate in pem format and enter “quit” at the end
in a line by itself]

PKI-Client-1(config)# crypto pki authenticate MGMT

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjA0MjI3
WhcNMTgxMDE3MjA0MjI3WjAuMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
```

```
MQ4wDAYDVQQDEwVTdWJDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJ7hKmBfDo/GOQAEYY/1ptpg28DejUE0ZlDorDkADP2vKfRI0kalSnOs2PIe01ip
7pHFurFVUx/p8teMCkmvnbrSBfyUrWo9YfQeGOELb4d3dSW4jGakm6M8lNRkO7HP
s+IVVTuJSeUZxov6DPa92Y/6HLayX15Iq8ZL+KwmA9oS5NeTi1tBbrcc3Hq8W2Ay
879nDDOqDOsQMQqKtc7E/IA7SBjowImra6FUxzgJ5ye5MymRfRYAH+c4qZJxwHTc
/tSmjiOJlM7X5dtehU/XPEEEbs78peXO9FyzAbhOtCRBVTnhc8WWijq84xu8Oej7
LbXGBKIHSP0uDe32CV0noEUCAwEAAaNgMF4wDwYDVR0TAQH/BAUwAwEB/zALBgNV
HQ8EBAMCAYYwHwYDVR0jBBgwFoAU+oNBdIj9mjpieQ2Z7v79JhKnL68wHQYDVR0O
BBYEFFOv8xtHRojMdJ65oQ2PFBeD5oHiMA0GCSqGSIb3DQEBBQUAA4IBAQAZ/W3P
Wqs4vuQ2jCnVE0v1PVQe/VNS54P/fprQRelceawiBCHA3D0SRgHqUWJUIqBLv4sD
QBegmyTmS76C8YC/jN7VbI30hf6R4qP7CWu8Ef9sWPRC/+Oy6e8AinrK+sVd2dp/
LLDMVoBhS2bQFLWiyRvC9FgyczXRdF+rhKTKeEVXGs7C/Yk/9z+/00rVmSGZAS+v
aPpZWjoC3459t51t8Y3iE6GtjBvmyxBwWt01/5gCu6Mszi7X/kXdmqgNfT5bBBnv
yjWE2ZS8NsH4hwDZpmDJqx4qhrH6bw3iUm+pK9fceZ/HTYasxtcr4NUvvxwXc60y
Wrtlpq3g2XfG+qFB
-----END CERTIFICATE-----
quit
Trustpoint 'MGMT' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
     Fingerprint SHA1: EAD41B32 BB37BC11 6E0FBC13 41701BFE 200DC46E

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

步驟2.產生憑證簽署請求，將CSR傳送到CA並取得授予的憑證：

```
PKI-Client-1(config)# crypto pki enroll MGMT
% Start certificate enrollment ..

% The subject name in the certificate will include: CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
% The subject name in the certificate will include: PKI-Client-1.cisco.com
% The serial number in the certificate will be: 104Certificate Request follows:

MIIC2zCCAcMCAQAwdTEOMAwGA1UEChMFQ2lzY28xDDAKBgNVBAsTA1RBQzENMAsG
A1UECxMETUdNVDETMBEGA1UEAxMKUEtJLUNsaWVudDExMAoGA1UEBRMDMTA0MCMG
CSqGSIb3DQEJAhYWUEtJLUNsaWVudC0xLmNpc2NvLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANwa7g+DJxG57sMg020w1Fdv9+mIZ6R4livbt7vo
AbW8jpzQlMv4lV3r6ulTJumhBvV7xI+1ZijXP0EqqQZLNboYv37UTJgm83DGO57I
8RTn9DfDQpHiqvhtNuC5S3SCC/hvCxFXnfNXqC3dkfuVkVWoJiLZY87R6j44jUq0
tTL5d8t6lz2L0BeekzKJlOs73gONx0VgQyI/WjDiEwL0xF4DNHURaYyOxBWJc7/B
psDCf7376mb7XXz0LB++E8SvvM/Li6+yQzYv1Lagr0b8C4uE+tCDxG5OniNDiS82
JXsVd43vKRFW85W2ssrElgkuWAvS017XlwK+UDX21dtFdfUCAwEAAaAhMB8GCSqG
SIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUAA4IBAQA+
UqkqUZZar9TdmB8I7AHku5m79l42o8cuhwOccehxE6jmzh9P+Ttb9Me7l7L8Y2iR
yYyJHsL7m6tjK2+Gllg7RJdoxG8l8aMZS1ruXOBqFBrmo7OSz1nfXpiTyh88jyca
Hw/8G8uaYuQbZIj53BWmQGRpm7J//ktn0D4W3Euh9HttMuYYX7BOct05BLqqiCCw
n+kKHZxzGXy7JSZpUlDtvPPnnuqWK7iVoy3vtV6GoFOrxRoo05QVFehS0/m4NFQI
mXA0eTEgujSaQi4iWte/UxruO/3p/eHr67MtZXLRl0YDFgaQd7vD7fCsDx5pquKV
jNEUT6FNHdsnqrAKqodO

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]: no
```

步驟3.現在通過終端匯入已授予的證書：

```
PKI-Client-1(config)# crypto pki import MGMT certificate

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBAzANBgkqhkiG9w0BAQQFADAuMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ4wDAYDVQQDEwVTdWJDQTAeFw0xNTEwMTkyMDM1MDZa
Fw0xNjEwMTgyMDM1MDZaMHUxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNUQUMx
DTALBgNVBAsTBE1HTVQxEzARBgNVBAMTClBLSS1DbGllbnQxMTAKBgNVBAUTAzEw
NDAjBgkqhkiG9w0BCQIWFlBLSS1DbGllbnQtMS5jaXNjby5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDcGu4PgycRue7DINNtMNRXb/fpiGekeJYr
27e76AG1vI6c0JTL+JVd6+rpUybpoQb1e8SPtWYo1z9BKqkGSzW6GL9+1EyYJvNw
xjueyPEU5/Q3w0KR4qr4bTbguUt0ggv4bwsRV53zV6gt3ZH7lZFVqCYi2WPO0eo+
OI1KtLUy+XfLepc9i9AXnpMyiZTrO94DjcdFYEMiP1ow4hMC9MReAzR1EWmMjsQV
iXO/wabAwn+9++pm+1189CwfvhPEr7zPy4uvskM2L9S2oK9G/AuLhPrQg8RuTp4j
Q4kvNiV7FXeN7ykRVvOVtrLKxJYJLlgL0tNe15cCvlA19tXbRXX1AgMBAAGjUjBQ
MA4GA1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBRTr/MbR0aIzHSeuaENjxQXg+aB
4jAdBgNVHQ4EFgQUK+9/lrlL+TyYxvsgxzPwwrhmS5UwDQYJKoZIhvcNAQEEBQAD
ggEBAIrLrzFLnm9z7ula1uRh03r6dSCFy9XkOk6ZaHfksbENoDmkcgIwKoAsSF9E
rQmA9W5qXVU7PEsqOmcu8zEv7uuiqM4D4nDP69HsyToPjxVcoG7PSyKJYnXRgkVa
IYyMaSaRKWlhb2uWj3XPLzS0/ZBOGAG9rMBVzaqLfLAZgnQUVJvwsNofe+ASojk9
mCRsEHD8WVuAzcnwYKXx3j3x/T7jbB3ibPfbYKQqlS12XFHhJoK+HfSA2fyZBFLF
syN/B2Ow0bvc71YlYOQuYwz3XOMIHD6vARTO4f0ZIQti2dy1kHc+5lIdhLsn/bA5
yUo7WxnAE8LOoYIf9iU9q0mqkMU=
-----END CERTIFICATE-----
quit
% Router Certificate successfully imported
```

## PKI伺服器

步驟1.首先從CA匯出頒發CA證書，在本例中為SUBCA證書。在上面步驟1中，在PKI客戶端（即信任點身份驗證）上匯入此項。

```
SUBCA(config)# crypto pki export SUBCA pem terminal
% CA certificate: !! Root-CA certificate
-----BEGIN CERTIFICATE-----
MIIDPDCCAiSgAwIBAgIBATANBgkqhkiG9w0BAQQFADAvMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjAwOTIx
WhcNMjMxMDE2MjAwOTIxWjAvMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ8wDQYDVQQDEwZSb290Q0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
AQCajfMy8gU3ZXQfKgP/wYKLB0cuywzYcDaSoNVlEvUZOWgUltCGP4CiCXyw0U0U
Zmy0rusibMV7mtkTX5muaPC0XfT98rswPiZV0qvEYpHF2YodPOUoqR3FeKj/tDbI
IikcLrfj87aeMJjCrWD888wfTN9Hw9x2QVDoSxLbzTLticXdXxwS5wxlM16GspmT
WL4fg1JRWgjRqMmOcpf716Or88XJ2N2HeWxxVFIwYQf3thHR6DgTdCgJ1uqjVE6q
1LQ1g8k81mvuCZX0uLZiTMJ69xo+Ot/RpeeE2RShxK5rh56ObQq4MT4lbIPKqIxU
lbKzWdh10NiYwjgTNwTs9GGvAgMBAAGjYzBhMA8GA1UdEwEB/wQFMAMBAf8wDgYD
VR0PAQH/BAQDAgGGMB8GA1UdIwQYMBaAFPqDQXSI/Zo6YnkNme7+/SYSpy+vMB0G
A1UdDgQWBBT6g0F0iP2aOmJ5DZnu/v0mEqcvrzANBgkqhkiG9w0BAQQFAAOCAQEA
VKwqI9vpmoRh9QoOJGtOA3qEgV4eCfXdMuYxmmo0sdaBYBfQm2RhZeQ1X90vVBso
G4Wx6cJVSXCtkqZTm1IoMtya+gdhLbKqZmxc+I5/js88SrbrBIm4zj+sOoySV9kW
THEEmZjdTCWXo2wNCr23gGdnb4RqZ0FTOfoZO/2Xnpcbvhz2/K7wlDRJ5k1wrsRW
RRwsQEh4LYMFIg0aBs4gmRLZ8ytwrvvrhQTVrAA/MeomUEPhcIYESg1AlWxoCYZU
0iqKfDa9+4wEJ+PMGDhM2UV0fuP0rWitKWxecSVbo54z3VHYwwCbz2jCs8XGE61S
+XlxCZKFVdlVaMmuaZTdFg==
-----END CERTIFICATE-----

% General Purpose Certificate: !! SUBCA certificate
-----BEGIN CERTIFICATE-----
MIIDODCCAiCgAwIBAgIBAjANBgkqhkiG9w0BAQUFADAvMQ4wDAYDVQQKEwVDaXNj
```

```
bzEMMAoGA1UECxMDVEFDMQ8wDQYDVQQDEwZSb290Q0EwHhcNMTUxMDE4MjA0MjI3
WhcNMTgxMDE3MjA0MjI3WjAuMQ4wDAYDVQQKEwVDaXNjbzEMMAoGA1UECxMDVEFD
MQ4wDAYDVQQDEwVTdWJDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
AJ7hKmBfDo/GOQAEYY/1ptpg28DejUE0ZlDorDkADP2vKfRI0kalSnOs2PIe01ip
7pHFurFVUx/p8teMCkmvnbrSBfyUrWo9YfQeGOELb4d3dSW4jGakm6M8lNRkO7HP
s+IVVTuJSeUZxov6DPa92Y/6HLayX15Iq8ZL+KwmA9oS5NeTi1tBbrcc3Hq8W2Ay
879nDDOqDOsQMQqKtc7E/IA7SBjowImra6FUxzgJ5ye5MymRfRYAH+c4qZJxwHTc
/tSmjiOJlM7X5dtehU/XPEEEbs78peXO9FyzAbhOtCRBVTnhc8WWijq84xu8Oej7
LbXGBKIHSP0uDe32CV0noEUCAwEAAaNgMF4wDwYDVR0TAQH/BAUwAwEB/zALBgNV
HQ8EBAMCAYYwHwYDVR0jBBgwFoAU+oNBdIj9mjpieQ2Z7v79JhKnL68wHQYDVR0O
BBYEFFOv8xtHRojMdJ65oQ2PFBeD5oHiMA0GCSqGSIb3DQEBBQUAA4IBAQAZ/W3P
Wqs4vuQ2jCnVE0v1PVQe/VNS54P/fprQRelceawiBCHA3D0SRgHqUWJUIqBLv4sD
QBegmyTmS76C8YC/jN7VbI30hf6R4qP7CWu8Ef9sWPRC/+Oy6e8AinrK+sVd2dp/
LLDMVoBhS2bQFLWiyRvC9FgyczXRdF+rhKTKeEVXGs7C/Yk/9z+/00rVmSGZAS+v
aPpZWjoC3459t51t8Y3iE6GtjBvmyxBwWt01/5gCu6Mszi7X/kXdmqgNfT5bBBnv
yjWE2ZS8NsH4hwDZpmDJqx4qhrH6bw3iUm+pK9fceZ/HTYasxtcr4NUvvxwXc60y
Wrtlpq3g2XfG+qFB
-----END CERTIFICATE-----
```

步驟2.在PKI-Client上執行步驟2後，從客戶端獲取CSR，並使用以下命令提供它以在SUBCA上簽名：

```
crypto pki server SUBCA request pkcs10 terminal pem
```

此命令建議SUBCA接受來自終端的證書簽名請求，一旦獲得授權，證書資料將以PEM格式列印。

```
SUBCA# crypto pki server SUBCA request pkcs10 terminal pem
PKCS10 request in base64 or pem

% Enter Base64 encoded or PEM formatted PKCS10 enrollment request.
% End with a blank line or "quit" on a line by itself.
MIIC2zCCAcMCAQAwdTEOMAwGA1UEChMFQ2lzY28xDDAKBgNVBAsTA1RBQzENMAsG
A1UECxMETUdNVDETMBEGA1UEAxMKUEtJLUNsaWVudDExMAoGA1UEBRMDMTA0MCMG
CSqGSIb3DQEJAhYWUEtJLUNsaWVudC0xLmNpc2NvLmNvbTCCASIwDQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBANwa7g+DJxG57sMg020w1Fdv9+mIZ6R4livbt7vo
AbW8jpzQlMv4lV3r6ulTJumhBvV7xI+1ZijXP0EqqQZLNboYv37UTJgm83DGO57I
8RTn9DfDQpHiqvhtNuC5S3SCC/hvCxFXnfNXqC3dkfuVkVWoJiLZY87R6j44jUq0
tTL5d8t6lz2L0BeekzKJlOs73gONx0VgQyI/WjDiEwL0xF4DNHURaYyOxBWJc7/B
psDCf7376mb7XXz0LB++E8SvvM/Li6+yQzYv1Lagr0b8C4uE+tCDxG5OniNDiS82
JXsVd43vKRFW85W2ssrElgkuWAvS017XlwK+UDX21dtFdfUCAwEAAaAhMB8GCSqG
SIb3DQEJDjESMBAwDgYDVR0PAQH/BAQDAgWgMA0GCSqGSIb3DQEBBQUAA4IBAQA+
UqkqUZZar9TdmB8I7AHku5m79l42o8cuhwOccehxE6jmzh9P+Ttb9Me7l7L8Y2iR
yYyJHsL7m6tjK2+Gllg7RJdoxG8l8aMZS1ruXOBqFBrmo7OSz1nfXpiTyh88jyca
Hw/8G8uaYuQbZIj53BWmQGRpm7J//ktn0D4W3Euh9HttMuYYX7BOct05BLqqiCCw
n+kKHZxzGXy7JSZpUlDtvPPnnuqWK7iVoy3vtV6GoFOrxRoo05QVFehS0/m4NFQI
mXA0eTEgujSaQi4iWte/UxruO/3p/eHr67MtZXLRl0YDFgaQd7vD7fCsDx5pquKV
jNEUT6FNHdsnqrAKqodO
quit
% Enrollment request pending, reqId=1
```

如果CA處於自動授予模式，則已授予的證書以上述PEM格式顯示。當CA處於手動授予模式時，證書請求被標籤為**pending**，分配了一個id值並排入註冊請求隊列中。

```
SUBCA#show crypto pki server SUBCA requests
Enrollment Request Database:

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

```
1      pending    7710276982EA176324393D863C9E350E serialNumber=104+hostname=PKI-Client-
1.cisco.com,cn=PKI-Client,ou=MGMT,ou=TAC,o=Cisco
```

步驟3.使用以下命令手動授予此請求：

```
SUBCA# crypto pki server SUBCA grant 1
% Granted certificate:
-----BEGIN CERTIFICATE-----
MIIDcDCCAligAwIBAgIBAzANBgkqhkiG9w0BAQQFADAuMQ4wDAYDVQQKEwVDaXNj
bzEMMAoGA1UECxMDVEFDMQ4wDAYDVQQDEwVTdWJDQTAeFw0xNTEwMTkyMDM1MDZa
Fw0xNjEwMTgyMDM1MDZaMHUxDjAMBgNVBAoTBUNpc2NvMQwwCgYDVQQLEwNUQUMx
DTALBgNVBAsTBE1HTVQxEzARBgNVBAMTClBLSS1DbGllbnQxMTAKBgNVBAUTAzEw
NDAjBgkqhkiG9w0BCQIWFlBLSS1DbGllbnQtMS5jaXNjby5jb20wggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDcGu4PgycRue7DINNtMNRXb/fpiGekeJYr
27e76AG1vI6c0JTL+JVd6+rpUybpoQb1e8SPtWYo1z9BKqkGSzW6GL9+1EyYJvNw
xjueyPEU5/Q3w0KR4qr4bTbguUt0ggv4bwsRV53zV6gt3ZH7lZFVqCYi2WPO0eo+
OI1KtLUy+XfLepc9i9AXnpMyiZTrO94DjcdFYEMiP1ow4hMC9MReAzR1EWmMjsQV
iXO/wabAwn+9++pm+1189CwfvhPEr7zPy4uvskM2L9S2oK9G/AuLhPrQg8RuTp4j
Q4kvNiV7FXeN7ykRVvOVtrLKxJYJLlgL0tNe15cCvlA19tXbRXX1AgMBAAGjUjBQ
MA4GA1UdDwEB/wQEAwIFoDAfBgNVHSMEGDAWgBRTr/MbR0aIzHSeuaENjxQXg+aB
4jAdBgNVHQ4EFgQUK+9/lrlL+TyYxvsgxzPwwrhmS5UwDQYJKoZIhvcNAQEEBQAD
ggEBAIrLrzFLnm9z7ula1uRh03r6dSCFy9XkOk6ZaHfksbENoDmkcgIwKoAsSF9E
rQmA9W5qXVU7PEsqOmcu8zEv7uuiqM4D4nDP69HsyToPjxVcoG7PSyKJYnXRgkVa
IYyMaSaRKWlhb2uWj3XPLzS0/ZBOGAG9rMBVzaqLfLAZgnQUVJvwsNofe+ASojk9
mCRsEHD8WVuAzcnwYKXx3j3x/T7jbB3ibPfbYKQqlS12XFHhJoK+HfSA2fyZBFLF
syN/B2Ow0bvc71YlYOQuYwz3XOMIHD6vARTO4f0ZIQti2dy1kHc+5lIdhLsn/bA5
yUo7WxnAE8LOoYIf9iU9q0mqkMU=
-----END CERTIFICATE-----
```

**附註**：無法將子CA手動註冊到根CA。

**附註**：由於已禁用HTTP伺服器而處於禁用狀態的CA可以手動授予證書請求。

## 使用SCEP註冊

PKI客戶端配置為：

```
crypto pki trustpoint MGMT
enrollment url http://172.16.1.2:80
 serial-number
 ip-address none
 password 7 110A1016141D5A5E57
 subject-name CN=PKI-Client,OU=MGMT,OU=TAC,O=Cisco
 revocation-check crl
 rsakeypair PKI-Key 2048
```

PKI伺服器配置為：

```
SUBCA# show run all | section pki server
crypto pki server SUBCA
 database level complete
 database archive pkcs12 password 7 01100F175804575D72
 issuer-name CN=SubCA,OU=TAC,O=Cisco
 lifetime crl 12
```

```
 lifetime certificate 365
 lifetime ca-certificate 1095
 lifetime enrollment-request 168
 mode sub-cs
 auto-rollover 85
 database url ftp:://10.1.1.1/CA/SUB/
database url crl ftp:://10.1.1.1/CA/SUB/
database url crl publish ftp:://10.1.1.1/WWW/CRL/SUB/
```

證書請求授予的預設模式為手動：

```
SUBCA# show crypto pki server
Certificate Server SUBCA:
   Status: enabled
   State: enabled
   Server's configuration is locked  (enter "shut" to unlock it)
   Issuer name: CN=SubCA,OU=TAC,O=Cisco
   CA cert fingerprint: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
   Server configured in subordinate server mode
   Upper CA cert fingerprint: CD0DE4C7 955EFD60 296B7204 41FB6EF6
    Granting mode is: manual
   Last certificate issued serial number (hex): 4
   CA certificate expiration timer: 21:42:27 CET Oct 17 2018
   CRL NextUpdate timer: 09:42:37 CET Oct 20 2015
   Current primary storage dir: unix:/SUB/
   Current storage dir for .crl files: unix:/SUB/
   Database Level: Complete - all issued certs written as <serialnum>.cer
   Auto-Rollover configured, overlap period 85 days
   Autorollover timer: 21:42:27 CET Jul 24 2018
```

**手動授予**

步驟1. PKI客戶端：作為第一步（必需），在PKI客戶端上驗證信任點：

```
PKI-Client-1(config)# crypto pki authenticate MGMT
Trustpoint 'MGMT' is a subordinate CA and holds a non self signed cert
Certificate has the following attributes:
      Fingerprint MD5: DBE6AFAC 9E1C3697 01C5466B 78E0DFE3
     Fingerprint SHA1: EAD41B32 BB37BC11 6E0FBC13 41701BFE 200DC46E

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

步驟2. PKI客戶端：在信任點身份驗證之後，可以為PKI客戶端註冊證書。

**附註**：如果配置了自動註冊，客戶端將自動執行註冊。

```
config terminal
crypto pki enroll MGMT
```

這些事件在幕後發生：

- IOS查詢名為PKI-Key的RSA金鑰對。如果存在，則會擷取該封包以要求取得身分憑證。否則，IOS會建立一個名為PKI-Key的2048位金鑰對，然後使用它請求身份證書。
- IOS以PKCS10格式建立證書簽名請求。

- 然後IOS使用隨機對稱金鑰加密此CSR。隨機對稱金鑰使用接收者的公鑰加密，該公鑰是SUBCA（由於信任點身份驗證，SUBCA的公鑰可用）。 加密的CSR、加密的隨機對稱金鑰和收件人資訊被放在PKCS#7封裝資料中。

- 此PKCS#7封裝資料在初始註冊期間使用臨時自簽名證書進行簽名。PKCS#7封裝資料、客戶端使用的簽名證書和客戶端的簽名一起放在PKCS#7簽名資料包中。先進行base64編碼，然後進行URL編碼。產生的blob資料作為「message」引數在HTTP URI中傳送至CA:

```
GET /cgi-bin/pkiclient.exe?operation=PKIOperation&message=MII…. HTTP/1.0
```

步驟3. PKI-Server:

IOS PKI伺服器收到請求時，會檢查以下內容：

1.檢查註冊請求資料庫是否包含具有與新請求關聯的相同事務ID的證書請求。

**附註**：事務ID是公鑰的MD5雜湊，客戶端正在請求其身份證書。

2.檢查註冊請求資料庫是否包含與客戶端傳送的證書請求具有相同質詢密碼的證書請求。

**附註**：如果(1)返回true或(1)和(2)一起返回true，則CA伺服器能夠以重複的身份請求為由拒絕該請求。但是，在這種情況下，IOS PKI伺服器會用較新的請求替換較舊的請求。

步驟4. PKI-Server:

在PKI伺服器上手動授予請求：

要檢視請求，請執行以下操作：

```
show crypto pki server SUBCA requests
```

要批准特定請求或所有請求，請執行以下操作：

```
crypto pki server SUBCA grant <id|all>
```

步驟5. PKI客戶端：

同時，PKI客戶端啟動POLL計時器。這裡，IOS會定期執行GetCertInitial，直到客戶端收到隨已授予的證書一起提供的SCEP CertRep = GRANTED。

收到授予的證書後，IOS會自動安裝該證書。