IOS PKI部署指南：初始设计和部署

目录

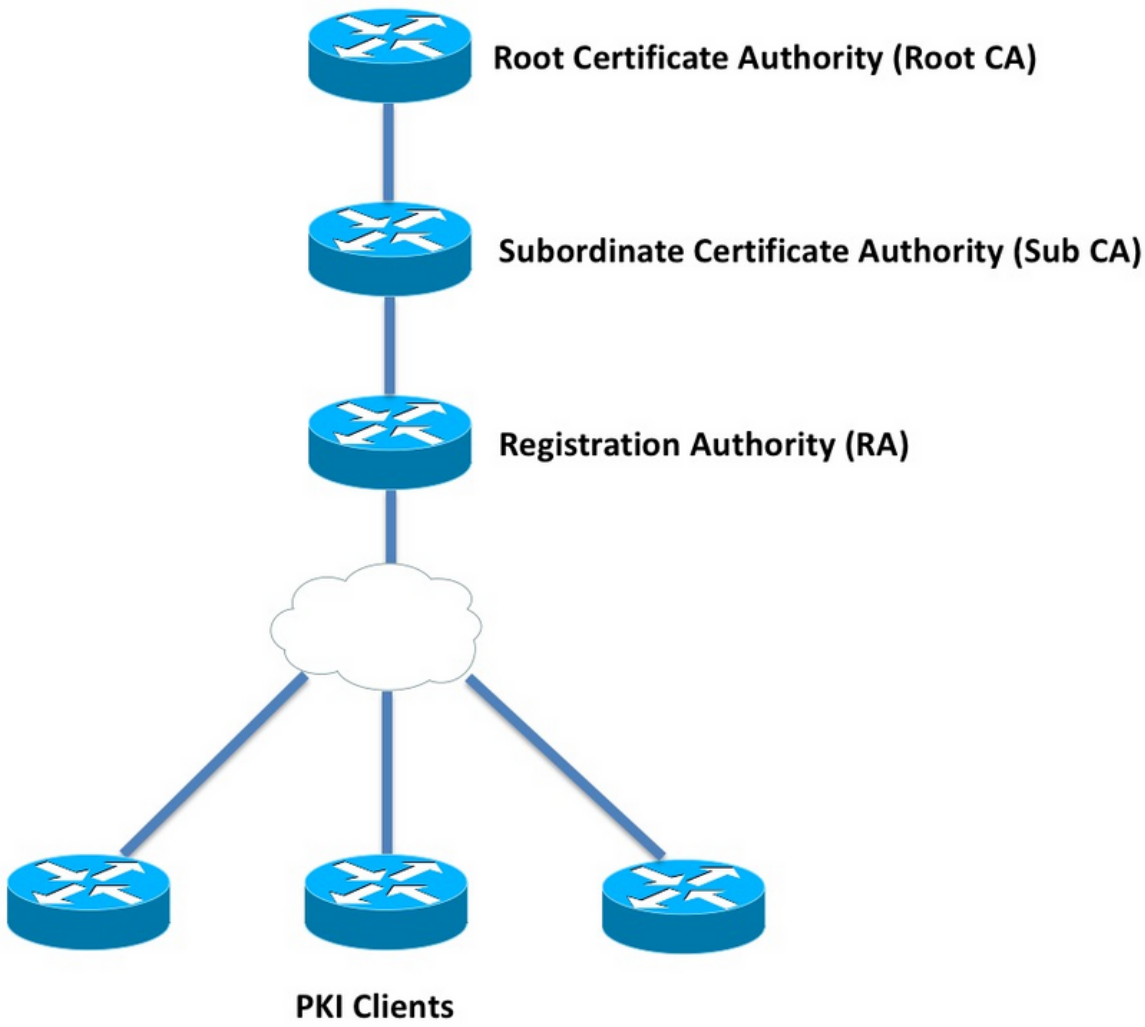
简介
PKI基础设施
认证中心
辅助认证机关
注册审批机构
PKI客户端
IOS PKI服务器
时间授权来源
主机名和域名
HTTP 服务器
RSA密钥对
自动反转计时器考虑事项
CRL考虑事项
发布CRL对HTTP服务器
SCEP GetCRL方法
寿命CRL
数据库考虑事项
Database archive
IOS作为SUB CA
IOS作为RA
IOS PKI客户端
时间授权来源
主机名和域名
RSA密钥对
信任点
登记模式
源接口和VRF
自动证书登记和续订
证书撤销检查
CRL缓存
推荐 的 配置
根CA -配置
没有RA的SUBCA -配置
与RA的SUBCA -配置
SUBCA的RA -配置
证书登记
手动注册
PKI客户端
PKI服务器
登记使用SCEP

[手工的授予](#)
[无条件的自动格兰特](#)
[已授权自动格兰特](#)
[登记使用SCEP通过RA](#)
[自动格兰特RA授权的请求](#)
[自动格兰特SUB CA/RA反转证书](#)

简介

本文详细描述IOS PKI服务器和客户端功能。 它关注IOS PKI初始设计和部署注意事项。

PKI基础设施



认证中心

Certificate Authority (CA)，也指PKI服务器在本文中，是发行证书的可信的实体。PKI根据信任，并且托拉斯层级开始在根认证机关(根CA)。由于根CA是在层级顶部，有一自签名证书。

辅助认证机关

在PKI托拉斯层级所有证书权限在根之下叫作辅助证书权限(SUB CA)。明显， SUB CA证书由CA发出，是上面一个级别。

PKI不实施限制给SUB CAS编号在一给的层级的。然而，在与超过3个级别的一个企业部署证书权限可能变得难管理。

注册审批机构

PKI定义了叫作注册机关(RA)认证机关的特殊，对授权从登记的PKI客户端负责到一个给的SUB CA或根CA。RA不发行证书给PKI客户端，反而决定哪个Pki客户端或不可能由SUB CA或根CA发出证书。

RA的主要角色是卸载从CA的基本客户端证书请求验证，并且保护从直接暴露的CA对客户端。这样， RA突出在PKI客户端和CA之间，因而保护从任何的CA拒绝服务攻击。

PKI客户端

请求为证书的所有设备根据一常驻公用专用密钥对证明其标识到其它设备是公认的PKI客户端。

PKI客户端一定能够生成或存储一公用专用密钥对例如RSA或DSA或者ECDSA。

假设对应的专用密钥在设备，存在证书是给的公共密钥的身份证明和正确性。

IOS PKI服务器

表1. IOS PKI服务功能演变

功能	IOS [ISR-G1， ISR-G2]	IOS-XE [ASR1K，ISR4K]
IOS CA/PKI服务器	12.3(4)T	XE 3.14.0/15.5(1)S
IOS PKI服务器证书反转	12.4(1)T	XE 3.14.0/15.5(1)S
IOS PKI HA	15.0(1)M	NA [Implicit Inter-RP Redundancy is available]
第三方CA的IOS RA	15.1(3)T	XE 3.14.0/15.5(1)S

在进入PKI服务器配置前，管理员必须了解这些核心概念。

时间授权来源

其中一个PKI基础设施的基础是时间。系统时钟定义了是否证书有效。因此，在IOS，必须使时钟授权或值得信任。没有时间一授权来源， PKI服务器可能不作用正如所料，并且是高度推荐的使时钟在IOS授权使用这些方法：

NTP (网络时钟协议)

同步系统时钟与时间服务器是唯一的真的方式进行值得信任的系统时钟。IOS路由器可以配置作为对一著名的和稳定的Ntp server的一NTP客户机在网络：

```
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可能也配置作为Ntp server，将指示本地系统时钟如授权。在小规模PKI部署， PKI服务器可以配置作为其PKI客户端的一Ntp server ：

```
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
```

指示的硬件时钟作为委托

在IOS中，硬件时钟可以被标记作为授权使用：

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

这可以与NTP一起配置，并且执行的此关键原因是保持系统时钟授权，当路由器重启，例如由于断电和NTP服务器不可及的时。在此阶段， PKI计时器将停止作用，反过来导致证书续订/反转失败。在这些情况下**clock calendar-valid**作为保障。

当配置此，它是关键了解时系统时钟将出去同步，如果系统电池中断，并且PKI将开始委托一个失调的时钟。然而，配置此，比有时间一授权来源是相对安全。

> **注意**：**clock calendar-valid命令**在IOS-XE版本XE 3.10.0/15.3(3)S向前被添加了。

主机名和域名

推荐配置主机名和a domain-name在Cisco IOS作为其中一第一步在配置任何PKI相关服务前。路由器主机名和domain-name用于以下方案：

- 默认RSA密钥对名称通过结合主机名派生和domain-name
- 当登记为证书时，请默认subject-name包括主机名属性和无特定结构的NAME，是主机名和domain-name汇集。

关于PKI服务器，和domain-name没有使用主机名：

- 默认密钥对名称将是相同的象那PKI服务器名
- 默认subject-name包括CN，是相同的象那PKI服务器名。

一般建议是配置适当的主机名和a domain-name。

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

HTTP 服务器

只有当HTTP服务器启用， IOS PKI服务器启用。请注意，如果PKI服务器禁用的归结于禁用的HTTP服务器，它能继续授权脱机请求[via terminal]。HTTP服务器功能要求处理SCEP请求，并且派出SCEP答复。

IOS HTTP服务器启用使用：

```
ip http server
```

并且默认HTTP服务器端口可以从80更改到任何有效端口号使用：

```
ip http port 8080
```

HTTP麦斯连接

其中一个瓶颈，当部署IOS作为PKI服务器使用SCEP时是最大并发HTTP连接和平均的HTTP连接每分钟。
默认情况下目前，在IOS HTTP服务器的最大并发连接被限制到5并且可以增加到16，是高度推荐的在一中比例尺部署：

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

此IOS安装允许最大并发HTTP连接至1000 ：

- UniversalK9与许可证设置的uck9的IOS

CLI自动地更改接受在1到1000之间的一个数字参数

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

IOS HTTP服务器允许每分钟[580 80连接一旦最大值HTTP并发会话可以增加到1000]的IOS版本，并且当此限制在一分钟内时达到， IOS HTTP监听程序开始通过关闭监听程序限制传入的HTTP连接15秒。这导致客户端连接的请求丢弃的归结于TCP**达到的连接队列限制**。可以找到关于此的更多信息[此处]

RSA密钥对

PKI服务器功能的RSA密钥对在IOS可以主动生成或手工生成。
当配置PKI服务器时， IOS由名称自动地创建信任点和PKI服务器一样为了存储PKI服务器证书。

手工生成PKI服务器RSA密钥对：

步骤1.创建与名称的一RSA密钥对和那PKI服务器一样：

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

第二步：在启用PKI服务器前，请修改PKI服务器信任点：

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

> **注意**：RSA密钥对模数值被提及在PKI服务器信任点下没有被考虑到直到IOS Ver 15.4(3)M4，并且这是已知问题说明。DEFAULT键模数是1024个位。

主动生成PKI服务器RSA密钥对：

当启用PKI服务器， IOS自动地生成与名称的一RSA密钥对和一样时那PKI服务器和关键模数大小是1024个位。

因为名称和密钥强度将是根据定义<MOD>模数，开始IOS Ver 15.4(3)M5，此配置创建与<LABEL>的一RSA密钥对。

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

掠夺者
CSCuu73408 IOS PKI服务器应该允许反转的cert非默认密钥大小。
CSCuu73408 IOS PKI服务器应该允许反转的cert非默认密钥大小。

当前工业标准是使用至少2048个位RSA密钥对。

自动反转计时器考虑事项

默认情况下目前， IOS PKI服务器不生成反转证书，使用自动**反转<days-before-expiry>**命令，并且必须明确地启用在PKI服务器下。更多在证书反转解释

此命令指定多少个天，在PKI Server/CA证书终止前如果IOS创建反转CA证书。注意反转CA证书一次激活当前活动CA证书超时。默认值当前是30天。应该设置此值为一个合理的值根据CA证书寿命，并且这反过来影响在PKI客户端的自动注册计时器配置。

> **注意**：在CA和客户端证书反转[known as]期间，自动反转计时器应该在自动注册之前总是触发在客户端的计时器

CRL考虑事项

IOS PKI基础设施支持分配CRL两种方式：

发布CRL对HTTP服务器

IOS PKI服务器可以配置发布CRL文件到HTTP服务器的一个特定位置使用此命令在PKI服务器下：

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

并且PKI服务器可以配置嵌入此CRL位置到所有PKI客户端证书使用此命令在PKI服务器下：

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

SCEP GetCRL方法

IOS PKI服务器在特定数据库位置自动地存储CRL文件，默认情况下是nvram，并且是高度推荐的保留在SCP/FTP/TFTP服务器的复制使用此命令在PKI服务器下：

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

默认情况下， IOS PKI服务器不嵌入CDP位置到PKI客户端证书。如果IOS PKI客户端配置执行撤销检查，但是验证的证书没有在它嵌入的CDP，并且验证的CA信任点配置与CA位置(使用http://<CA服务器IP或FQDN>)，默认情况下IOS下跌回到SCEP基于GetCRL方法。
SCEP GetCRL通过执行在此URL的HTTP GET进行CRL检索：

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

> **注意**：在IOS CLI，在输入之前 **?** 请按**Ctrl+V**键序列。

IOS PKI服务器能也嵌入此URL作为CDP位置。优点执行此是二倍的：

- 它保证所有非IOS SCEP基于PKI客户端可进行CRL检索。
- 没有嵌入式CDP， IOS SCEP GetCRL请求消息签字(使用一临时自签名证书)如对SCEP草稿定义。然而， CRL检索请求不需要签字，并且通过嵌入GetCRL方法的CDP URL，签署CRL请求可以避免。

寿命CRL

使用在PKI服务器下的此命令IOS PKI服务器的CRL寿命可以被控制：

```
crypto pki server <PKI-SERVER-Name>
```

```
lifetime crl <0 - 360>
```
值是以几小时。默认情况下CRL的寿命设置为6个小时。根据证书如何频繁地取消，调整CRL寿命对最佳值增加在网络的CRL检索性能。

数据库考虑事项

IOS PKI服务器使用nvram作为默认数据库位置，并且是高度推荐的使用FTP或TFTP或者SCP服务器作为数据库位置。默认情况下， IOS PKI服务器创建两个文件：

- <Server-Name>.ser –这包含在十六进制的CA发出的最后序列号。文件在纯文本格式，并且包含此信息：
  db_version = 1
  last_serial = 0x4

- <Server-Name>.crl –这是CA发布的DER编码的CRL文件

IOS PKI服务器存储在数据库的信息在3个可配置级别：

- 最低–这是默认级别，并且文件在数据库在这个阶层没有创建，并且信息不是可用的在关于以前授权的客户端证书的CA服务器。
- 名称– IOS PKI服务器在这个阶层创建名叫发出的每个客户端证书的<Serial-Number>.cnm的文件，其中命名<Serial-Number>是指发出的客户端证书的序列号，并且此cnm文件包含subject-name和客户端证书的到期日。
- 完整–在这个阶层， IOS PKI服务器创建发出的每个客户端证书的两个文件：
- <Serial-Number>.cnm
- <Serial-Number>.crt

此处， crt文件是客户端证书文件，是编码的DER。

这些点是重要：

- 在发出客户端证书前， IOS PKI服务器是指<Server-Name>.ser确定和派生证书的序列号。
- 使用数据库对名称的级别集或请完成， <Serial-Number>.cnm和<Serial-Number>.crt需要写入到数据库在发送授权的前/已签发证书对客户端
- database url设置为名称或请完成， database url必须有保存足够的空间文件。因此建议是配置外部文件服务器[FTP or TFTP or SCP]作为database url。
- 使用配置的外部数据库URL，确保是绝对必要的，文件服务器在证书授予进程中是可及的，将否则指示CA服务器如禁用。并且人工干预要求带来联机CA服务器的上一步。

Database archive

当部署PKI服务器时，考虑故障情景是重要的，并且准备，应该有hadrware失败。有两种方式达到此：

1. 冗余
   在这种情况下，两个设备或处理器作为激活待机提供冗余。
   高性能IOS PKI的服务器可以达到使用两HSRP启用的ISR路由器[ISR G1和ISR G2]按照说明

IOS XE根据系统[ISR4K和ASR1k]没有设备冗余选项联机。默认情况下然而，在ASR1k RP之间冗余是可用的。

2. 归档CA服务器密钥对和文件
   IOS提供一个设备归档PKI服务器密钥对和证书。使用文件的两种类型存档可以执行：
   PEM - IOS创建PEM格式文件存储RSA公共密钥，已加密RSA专用密钥， CA服务器证书。自动地归档反转密钥对和证书PKCS12 - IOS创建包含CA服务器证书和对应的RSA专用密钥的单个PKCS12文件加密使用密码。

使用在PKI服务器下的此命令数据库存档可以启用：

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

存储归档文件到独立服务器，可能使用安全协议(SCP)也是可能的使用以下命令在PKI服务器下：

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

在数据库的所有文件除了归档文件和。Ser文件，其他文件在明文并且不造成实时威胁，如果丢失，并且可以存储在独立服务器，无需导致开销，当写入文件，例如TFTP server时。

IOS作为SUB CA

默认情况下IOS PKI服务器占去根CA的角色。要配置辅助PKI服务器(SUB CA)，首先请启用此命令在PKI服务器配置部分下(在启用PKI服务器前) ：

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

使用此请配置根CA的URL在PKI服务器的信任点下：

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
 enrollment url <Root-CA URL>
```

启用此PKI服务器当前触发这些事件：

- PKI服务器信任点验证为了安装根CA证书。
- 在根CA验证后， IOS生成包含CA:的辅助CA [x509基本限制条件的CSR真标志]和发送它对根CA

不考虑在根CA配置的授予模式， IOS放CA (或RA)证书请求到待定队列。管理员必须手工授权CA证书。
查看待定证书请求和请求id ：

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

同意请求：

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

- 使用此，随后的SCEP POLL (GetCertInitial)操作在路由器下载SUB CA证书并且安装它，启用辅助PKI服务器

IOS作为RA

IOs PKI服务器可以配置作为一个给的辅助或根CA的一个注册审批机构。要配置PKI服务器作为注册审批机构，首先请启用此命令在PKI服务器配置部分下(在启用PKI服务器前) ：

```
crypto pki server <RA-SERVER-Name>
```

```
 mode ra
```
在此之后，请配置CA的URL在PKI服务器的信任点下。这指示哪个CA由RA保护：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
注册审批机构不发行证书，因此在RA下的**签发方名称**配置没有要求，并且没有效，即使配置。使用**subject-name命令，** subject-name RA配置在RA信任点下。配置**OU= ioscs RA**作为一部分subject-name为了IOS CA能识别IOS RA识别IOS RA授权的证书请求即是重要的。

IOS能作为注册审批机构对第三方CA例如Microsoft CA，并且为了坚持兼容IOS RA必须启用使用在PKI服务器配置部分下的此命令(在启用PKI服务器前)：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
使用RA证书，在默认RA模式， IOS签署客户端的要求[PKCS#10]。此操作指示IOS PKI服务器证书请求由RA授权。

使用透明RA模式， IOS转发在他们的原始格式的客户端的要求，无需介绍RA证书，并且这是与Microsoft CA兼容作为一著名的示例。

IOS PKI客户端

一个在IOS PKI客户端的多数必需的配置实体是信任点。信任点配置参数在此部分详细解释。

时间授权来源

作为时间被指出的前，授权来源是在PKI客户端的一个需求。IOS PKI客户端可以配置作为NTP客户机使用这些配置：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
主机名和域名

一一般建议是配置主机名和a domain-name在路由器：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```
RSA密钥对

在IOS PKI客户端，一个给的信任点登记的RSA密钥对可能自动地生成或手工生成。

自动RSA密钥生成过程介入以下：

- 默认情况下IOS创建512个位RSA密钥对
- 自动地生成的密钥对名称是hostname.domain NAME，是与设备一起的设备主机名domain-name
- 自动生成的密钥对没有被标记作为可导出。

自动RSA密钥生成过程介入以下：

- 随意地，一适当的优点的一般用途RSA密钥对可以手工生成使用：
- crypto pki trustpoint <RA-SERVER-Name>
  enrollment url <CA URL>
  subject-name CN=<Common Name>, **OU=ioscs RA**, OU=TAC, O=Cisco这里，标签- RSA密钥对名称
  MOD - RSA密钥模数或优点在360之间的位耕种4096，传统上是512， 1024， 2048或者4096。
  手工生成RSA密钥对优点是能力标记密钥对如可导出，反过来允许完全导出的身份证书，在另一个设备可能然后恢复。然而，一个人应该了解此操作安全影响。
- 使用此命令， RSA密钥对与在登记前的一信任点连接

  crypto pki trustpoint <RA-SERVER-Name>
  enrollment url <CA URL>
  subject-name CN=<Common Name>, **OU=ioscs RA**, OU=TAC, O=Cisco这里，如果名为<LABEL>的RSA密钥对已经存在，在信任点登记期间，然后它被拾起。
  如果名为<LABEL>的RSA密钥对不存在，在登记期间，则一个以下操作被执行：
  -，如果<MOD>参数没有通过，然后512个位密钥对名为<LABEL>生成。
  -，如果一个<MOD>参数通过，然后名为<LABEL>的<MOD>位通用密钥对生成
  -，如果两个<MOD>参数通过，然后一<MOD>位签名密钥对和一<MOD>位加密密钥对，两个已命名<LABEL>生成

信任点

信任点是有在IOS的一证书的一个抽象容器。单个信任点能够在指定时候存储两活动证书：

- 一个CA证书-装载CA证书到一给的信任点叫作信任点认证过程。
- CA发出的ID证书-加载或导入ID证书到一给的信任点叫作信任点登记进程。

信任点配置是公认的信任策略，并且这定义了那：

- 哪个CA证书装载在信任点？
- 哪个CA信任点是否登记？
- IOS如何登记信任点？
- 给的CA [loaded in the trustpoint]发出的证书如何验证？

信任点的主要组件解释此处。

登记模式

信任点登记模式，也定义了信任点认证模式，可以通过3主要手段执行：

1. 终端的登记-执行的信任点验证和证书登记手工方法使用复制-粘贴在CLI终端。
2. SCEP登记-信任点验证和登记使用SCEP在HTTP。
3. 登记配置文件-这里，验证和登记方法分开定义。与终端和SCEP登记方法一起，登记配置文件提供一个选项指定HTTP/TFTP命令进行从服务器的文件检索，使用验证或登记URL在配置文件下，定义。

源接口和VRF

信任点验证和登记在HTTP (SCEP)或TFTP (登记配置文件)使用IOS文件系统执行文件I/O操作。这

些信息包交换可以从一个特定源接口和VRF来源。

使用在信任点下的**源接口**和**VRF**子命令在经典信任点配置的情况下，此功能启用。

在登记配置文件、**源接口**和**登记的情况下|验证URL <http/tftp://Server-location > VRF <vrf-name>**命令提供同一个功能。

配置示例：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

或

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

自动证书登记和续订

IOS PKI客户端可以配置执行自动注册和续订使用此命令在Pki trustpoint部分下：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

这里，**自动注册<percentage> [regenerate]**命令状态IOS应该执行证书续订在正确地80%当前证书的寿命。

关键字**重新生成**阐明， IOS应该重新生成叫作Shadow密钥对的RSA密钥对在每证书续订操作时。

这是自动注册行为：

- 瞬间**自动注册**配置，如果信任点验证， IOS将执行自动注册到服务器查找在URL被提及作为**enrollment url命令一部分**在Pki trustpoint部分下或在登记配置文件下。
- 信任点用PKI服务器登记的瞬间或CA、更新或者SHADOW计时器在PKI客户端初始化根据当前身份证书的**自动注册**百分比安装在信任点下。此计时器可视下**显示crypto pki计时器**命令。更多在*参考的*计时器fuctions
- 续订功能支持来自PKI服务器。更多在此
  IOS PKI客户端执行续订的两种类型：
  隐式续订：如果PKI服务器不发送“续订”作为一个支持的功能， IOS进行一个最初的登记在定义自动注册百分比。即IOS使用一自签名证书签署续订请求。明确续订：当PKI服务器支持PKI客户端证书续订功能时，通告“续订”作为一个支持的功能。即IOS考虑到此功能在证书续订IOS期间使用当前活动身份证书签署续订证书请求。

应该保重，当配置自动注册百分比时。在部署的所有给的PKI客户端，如果情况出现身份证书超时在发出的CA证书的同时的地方，然后自动注册值应该总是触发[shadow]续订操作，在CA创建反转证书后。*参考的***PKI计时器从属关系**部分

证书撤销检查

即一已验证Pki trustpoint包含CA证书的Pki trustpoint有能力在执行的证书确认上在IKE或SSL协商时，对等项证书对彻底的证书确认被服从。其中一个验证方法是检查对等项证书废止状态使用以下两个方法之一：

- 证书撤销列表(CRL) -这是包含证书的序列号的文件取消由给的CA。使用发出的CA证书，此文件签字。使用HTTP或LDAP， CRL方法介入下载CRL文件。

- 联机证书状态协议(OCSP) - IOS设立有作为OCSP响应方呼叫的实体的通信信道，是指定的服务器由发出的CA。一个客户端例如IOS发送包含证书的序列号的请求验证。OCSP响应方回应给的序列号的废止状态。使用所有支持的应用/传输协议，通信信道可能设立，通常是HTTP。

撤销检查可以定义使用这些在Pki trustpoint部分下的命令：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

默认情况下，使用crl，信任点配置执行撤销检查。

方法可以被重新命令，并且废止状态检查按定义顺序被执行。方法“无”绕过撤销检查。

CRL缓存

对于CRL基于撤销检查，每证书确认可能触发一新CRL文件下载。并且，因为CRL文件变得更大或，如果控制分配点(CDP)离开，下载在每个验证过程中的文件阻碍协议的性能从属于证书确认。因此， CRL高速缓冲存储执行改进性能，并且高速缓冲存储CRL考虑到CRL正确性。

使用两个次参数， CRL正确性定义：**LastUpdate**，是上次CRL由发出的CA和**NextUpdate**发布，是时间是未来，当CRL文件新版本由发出的CA时发布。

只要CRL有效， IOS缓存每个下载的CRL为。然而，在某种状况下例如的CDP可及的临时地，在缓存长时间保留CRL可能是必要的。在IOS中被缓存的CRL能保留为，只要24个小时，在CRL正确性超时后，使用在Pki trustpoint部分下的此命令，并且这可以配置：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

在某种状况下例如取消在CRL有效性周期的发出的CA证书， IOS能confiugured频繁地删除缓存。通过过早删除CRL， IOS被迫频繁地下载CRL保持CRL缓存最新状态。此配置选项是可用的在Pki trustpoint部分下：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

并且终于， IOS可以配置不缓存CRL文件使用此命令在Pki trustpoint部分下：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

推荐 的 配置

与根CA和SUB CA配置的一典型CA部署是作为下面。示例也包括RA保护的SUB CA配置。

使用2048个位全面的RSA密钥对，此示例推荐设置where:
根CA有寿命8年
SUB CA有寿命3年
客户端证书自动地发出一年，配置为证书续订请求。

根CA -配置

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

没有RA的SUBCA -配置

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

与RA的SUBCA -配置

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

SUBCA的RA -配置

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

证书登记

手动注册

手动注册介入在PKI客户端的脱机CSR生成，手工复制给CA.管理员手册签署请求，然后导入到客户端。

PKI客户端

PKI客户端配置：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

步骤1:首先请验证信任点(这可能在步骤2)以后也执行。

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Ciscocrypto pki trustpoint <RA-SERVER-
Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

步骤2.生成证书签名请求并且采取CSR对CA并且获得授权的certificat ：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

第三步：现在请通过终端导入授权的证书：

```
crypto pki trustpoint <RA-SERVER-Name>
 enrollment url <CA URL>
 subject-name CN=<Common Name>, OU=ioscs RA, OU=TAC, O=Cisco
```

PKI服务器

步骤1:首先请导出从CA的发出的CA证书，在这种情况下是SUBCA证书。这导入在上面step1期间在PKI客户端，即信任点验证。

```
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-----
```

第二步：使用此命令，在Pki客户端的Step-2以后，请采取从客户端的CSR并且为签字提供它在SUBCA ：

```
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-----
```

此命令建议SUBCA接受从终端的一证书签名请求，并且一次授权，身份验证数据在PEM格式打印。

```
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-----
```

如果CA在自动格兰特模式，授权的证书在上面PEM格式显示。当CA在手工的授权的模式时，证书请求被标记作为**待定**，分配id值并且排队到注册请求队列。

```
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-----
```

步骤3.使用此命令，请手工同意此请求：

```
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-----
```

注意：一个SUB CA的手动注册对根CA的不是可能的。

注意：CA在于的禁用状态禁用HTTP服务器能手工同意证书请求。

登记使用SCEP

PKI客户端配置是：

```
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服务器配置是：

```
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
```

默认模式证书请求授权手工：

```
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
```

手工的授予

步骤1. PKI客户端：首先，是必须，请验证在PKI客户端的信任点：

```
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
```

步骤2. Pki客户端：在信任点验证之后， PKI客户端可以为证书登记。

注意：如果自动注册配置，客户端将自动地进行登记。

```
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
```

在幕后，这些事件发生：

- IOS寻找名为Pki KEY的一RSA密钥对。如果它存在，为请求身份证书被拾起。否则， IOS创建2048位密钥对名为Pki KEY，然后使用它请求身份证书。
- IOS创建在PKCS10格式的一证书签名请求。
- 使用随机的对称密钥， IOS然后加密此CSR。使用收件人的公共密钥，随机的对称密钥加密，是SUBCA (SUBCA的公共密钥是联机由于信任点验证)。 已加密CSR、已加密随机的对称密钥和接收信息在PKCS-7被包围的数据被汇集。
- 在最初的登记期间，此PKCS-7被包围的数据签字使用一临时自签名证书。PKCS-7包围了数据，客户端使用的签署的证书，并且客户端的签名在PKCS-7签名数据数据包被汇集。这是编码的base64编码的，然后URL。数据发生的blob发送作为“在HTTP URI的消息”参数发送对CA:

```
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
```

步骤3. Pki服务器：

当IOS PKI服务器收到请求时，检查这些：

1. 检查注册请求数据库是否包含与同样交易ID的证书请求关联与新要求。

注意：交易ID是公共密钥的MD5哈希，身份证书由客户端请求。

2. 检查注册请求数据库是否包含与私钥保护密码的证书请求和客户端发送的那个一样。

注意：如果(1)一起返回真或(1)和(2)回归真，则CA服务器能够拒绝请求根据重复的标识请求。然而， IOS PKI服务器在这种情况下取代请求与更新的请求。

步骤4. Pki服务器：

请手工同意在PKI服务器的请求：

查看请求：

```
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
```

同意特定请求或所有请求：

```
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
```

步骤5. Pki客户端：

同时， PKI客户端启动POLL计时器。这里， SCEP CertRep =授权与授权的证书一起由客户端，接收IOS定期执行GetCertInitial。

一旦授权的证书接收， IOS自动地安装它。