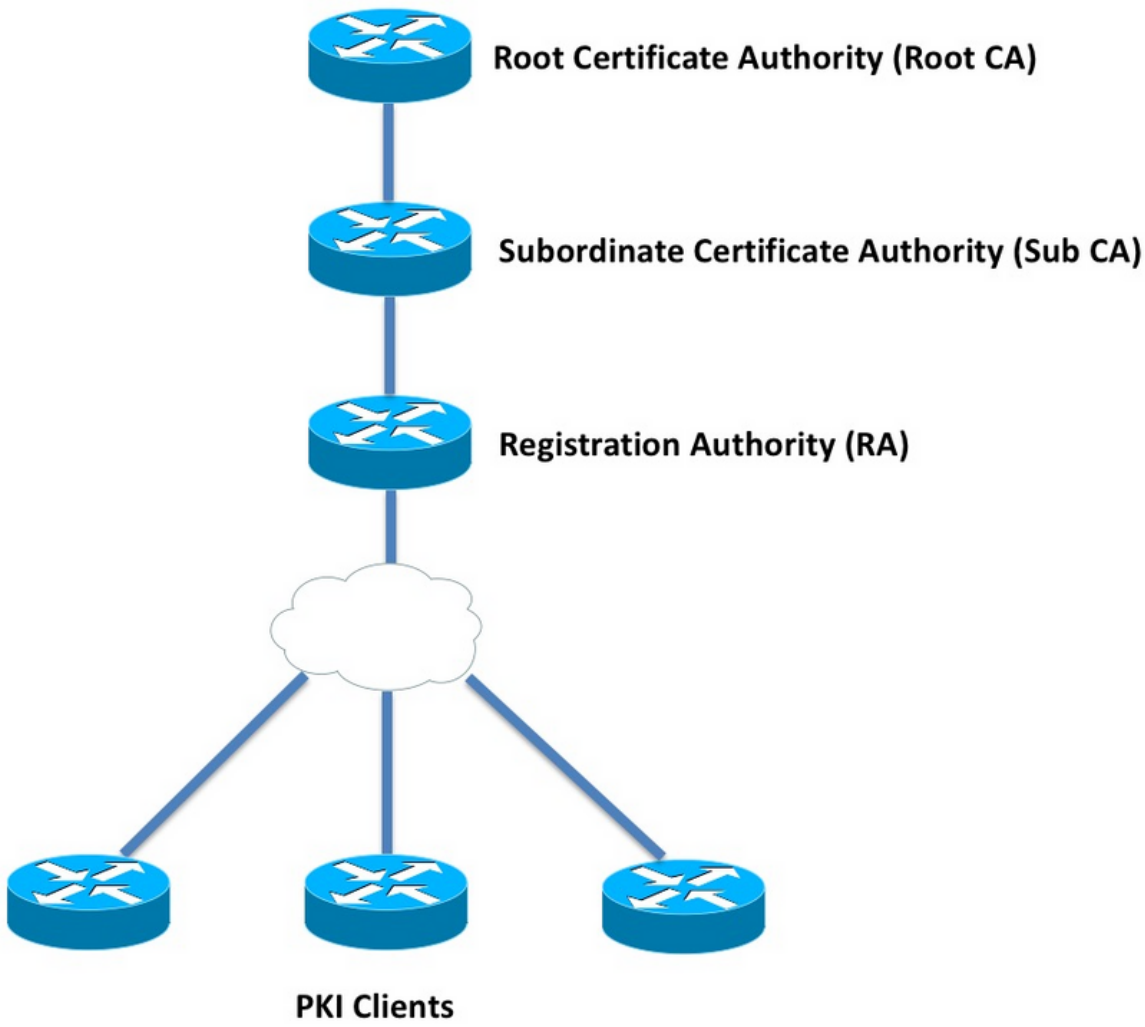
IOS PKI部署指南：初始设计和配置

Contents

Introduction

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



认证机关

Certificate Authority (CA)，也指PKI服务器在本文中，是该的可信的实体问题证书。PKI根据信任，并且信任层次结构开始在根认证机关(根CA)。由于根CA是在层次结构顶部，有一自签证书。

辅助认证机关

在PKI信任层次结构所有认证权限下面的根叫作辅助认证权限(SUB CA)。明显， CA发行SUB CA认证，是上面一个级别。

PKI不实施限制给SUB CAS的编号在一个特定层次结构的。然而，在与超过认证的3个级别的一个企业配置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)的特殊认证机关，对核准从登记的PKI客户端负责到一个特定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发出认证。

RA的主要角色是卸载从CA的基本的客户端证书请求验证，并且保护CA免受对客户端的直接暴露。这样， RA突出在PKI客户端和CA之间，因而保护CA免受任何拒绝服务攻击。

PKI客户端

所有设备请求为根据一个常驻公用专用密钥对的认证证明其身份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个公用专用密钥对例如RSA或DSA或者ECDSA。

假设对应的专用KEY在设备，存在认证是身份证明特定公共密钥的和正确性。

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服务器证明反转	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。

时间的授权来源

其中一个PKI基础设施的基础是时间。系统时钟定义了是否认证是有效的。因此，在IOS，必须使时钟授权或信得过。没有时间的一个授权来源， 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 (网络时间协议)

同步系统时钟与时间服务器是使系统时钟唯一的真的方式信得过。IOS路由器可以被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可能也被配置作为Ntp server，将指示本地系统时钟如授权。在小规模PKI配置， PKI服务器可以被配置作为其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
```

指示的硬件时钟如委托

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一起被配置，并且执行此的关键原因是保持系统时钟授权，当路由器重新载入，例如由于停电和NTP服务器不可及的时。在此阶段， PKI计时器将停止作用，反过来导致认证续订/反转故障。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，比有时间的一个授权来源是相对安全。

> **Note**:**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-和在Cisco IOS的一domain-name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
- 当登记为认证时，请默认subject-name包括被汇集的主机名-请归因于和无特定结构的NAME，是主机名-和domain-name。

关于PKI服务器，主机名-，并且没有使用domain-name ：

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

一般推荐是配置一个适当的主机名-和domain-name。

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

HTTP服务器

只有当HTTP服务器是启用的， IOS PKI服务器被启用。请注意，如果PKI服务器失效归结于是的HTTP服务器失效的，它能继续授予脱机请求[via terminal]。要求HTTP服务器功能处理SCEP请求，并且派出SCEP回应。

IOS HTTP服务器是启用使用：

```
ip http server
```

并且默认HTTP服务器端口可以从80被更换到所有有效端口号使用：

```
ip http port 8080
```

HTTP最大连接

其中一个瓶颈，当配置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，是高度推荐的在中比例尺配置：

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

此IOS安装允许最大并发HTTP连接至1000 ：

- UniversalK9与uck9许可证SET的IOS

自动地更改CLI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监听程序启动节流传入的HTTP连接通过关闭监听程序15秒。这导致客户端连接的请求下降的归结于TCP**达到的连接队列限制**。可以找到关于此的更多信息这里

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由名字自动地创建一信任点和PKI服务器一样为了存储PKI服务器证明。

手工生成PKI服务器RSA密钥对：

步骤1.用名字创建一个RSA密钥对和那PKI服务器一样：

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

Step 2.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>
```

> **Note**: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，并且这是一个已知警告。DEFAULT键模数是1024位。

主动生成PKI服务器RSA密钥对：

当启用PKI服务器， IOS自动地生成与名字的一个RSA密钥对和一样时那PKI服务器和关键模数大小是1024位。

因为名字和KEY力量将是根据被定义的<MOD>模数，开始IOS Ver 15.4(3)M5，此配置用<LABEL>创建一个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位RSA密钥对。

自动反转计时器考虑

默认情况下目前， IOS PKI服务器不生成反转认证，使用自动**反转<days-before-expiry>**命令，并且必须明确地被启用在PKI服务器下。更多在认证反转解释

此命令指定多少日，在PKI Server/CA认证终止前如果IOS创建反转CA证书。注意一次激活反转CA证书当前活动CA证书到期。DEFAULT值当前是30天。应该设置此值为合理的值根据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。

> **Note**: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

IOS PKI基础设施支持分配CRL两种方式：

发布CRL到HTTP服务器

可以配置IOS PKI服务器发布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>
```

并且可以配置PKI服务器嵌入此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。如果配置IOS PKI客户端执行撤销检查，但是被验证的认证没有在它嵌入的CDP，并且验证的CA信任点配置有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
```

> **Note**:在进入之前的IOS CLI中， **？**请按**Ctrl+ v KEY**顺序。

IOS PKI服务器能也嵌入此URL作为CDP位置。执行此的优点是二倍的：

- 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

IOS PKI服务器的CRL寿命可以是受控的使用此命令在PKI服务器下：

```
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小时。根据证书如何频繁地被废除，调整对一个最佳值的CRL寿命增加在网络的CRL检索性能。

数据库考虑

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 –这在十六进制包含CA发出的最后序列号。文件以明文格式，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服务器在数据库存储信息在3个可配置级别：

- 最小数量–这是默认级别，并且文件在数据库在这个阶层没有被创建，并且不是可用的在关于客户端证书的CA服务器以前被授予的信息。
- 名字– IOS PKI服务器在这个阶层创建名叫被发行的每个客户端证书的<Serial-Number>.cnm的一个文件，其中命名<Serial-Number>是指被发行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。
- 完成–在这个阶层， IOS PKI服务器创建被发行的每个客户端证书的两个文件：
- <Serial-Number>.cnm
- <Serial-Number>.crt

这里， crt文件是客户端证书文件，是编码的DER。

这些点是重要的：

- 在发行客户端证书前， IOS PKI服务器是指<Server-Name>.ser确定和派生认证的序列号。
- 使用数据库级别设置为名字或完成， <Serial-Number>.cnm和<Serial-Number>.crt需要给数据库被写在发送被授予的/发出的认证前到客户端
- database url设置为名字或请完成， database url必须有保存足够的空间文件。因此推荐是配置一外部文件服务器[FTP or TFTP or SCP]作为database url。
- 当外部Database url被配置，确信是绝对必要的，文件服务器在认证授予进程中是可及的，将否则指示CA服务器作为禁用。并且要求人工干预带来CA服务器返回在线。

Database archive

当配置PKI服务器时，考虑故障情景是重要的，并且准备，应该有hadrware故障。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活动暂挂提供冗余。
   高性能IOS PKI的服务器可以达到使用两HSRP被启用的ISR路由器[ISR G1和ISR G2]按照说明

   IOS XE根据系统[ISR4K和ASR1k]没有可用设备冗余的选项。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服务器密钥对和认证。使用文件的两种类型归档可以完成：
   PEM - IOS创建PEM格式文件存储RSA公共密钥，被加密的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文件被加密使用密码。

数据库归档可以是启用的使用此命令在PKI服务器下：

```
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际威胁，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。配置辅助PKI服务器(SUB CA)，第一enable (event)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
```

使用此请配置根加州的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基本的约束的CSR ：TRUE标志位]和发送它到根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。管理员必须手工授予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)操作下载SUB CA认证并且在路由器上安装它，enable (event)辅助PKI服务器

IOS作为RA

IOs PKI服务器可以被配置作为对一个特定辅助或根CA的一个注册审批机构。配置PKI服务器作为注册审批机构，第一enable (event)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受RA的保护：

```
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，因此没有需要在RA下的**签发方名称**配置，并且不是有效的，即使配置。使用**subject-name命令，** subject-name RA被配置在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核准的证书请求即是重要的。

IOS能作为注册审批机构对第三方CAs例如Microsoft CA，并且为了坚持兼容IOS RA必须是启用的使用此命令在PKI服务器配置部分下(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认证，在默认RA模式下， 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核准了。

使用透明RA模式， IOS转发客户端的要求以他们的原始格式，无需介绍RA认证，并且这是与Microsoft CA兼容作为一个著名的示例。

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。

时间的授权来源

作为时间的被指出的前，授权来源是在PKI客户端的一个需求。IOS PKI客户端可以被配置作为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

一般推荐是配置主机名-和在路由器的一domain-name ：

```
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，一个特定信任点登记的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位RSA密钥对
- 自动地生成的密钥对名称是hostname.domain NAME，是与设备domain-name -一起的设备主机名
- 主动生成的密钥对没有被标记作为可输出。

自动RSA密钥生成过程介入以下：

- 随意地，适当的力量的一般用途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密钥模数或力量在位在360之间耕种4096，传统上是512， 1024， 2048或者4096。
  手工生成RSA密钥对的优点是能力标记密钥对如可输出，反过来允许身份认证完全地被导出，在另一个设备可能然后恢复。然而，一个人应该了解此动作安全影响。
- 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，然后一个<MOD>位签名密钥对和一个<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的一个认证的一个抽象容器。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证书到一特定信任点叫作信任点认证过程。
- CA发行的ID认证-加载或导入ID认证到一特定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证书被装载在信任点？
- 哪个CA信任点是否登记？
- IOS如何登记信任点？
- 特定CA [loaded in the trustpoint]发行的认证如何被验证？

信任点的主要组件解释得这里。

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-进行信任点认证和认证登记手工方法使用复制-粘贴在CLI终端。
2. SCEP登记-信任点认证和登记使用SCEP在HTTP。
3. 登记配置文件-这里，认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，使用在配置文件下的认证或登记URL被定义。

源接口和VRF

信任点认证和登记在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。

在经典信任点配置的情况下，此功能使用**源接口**和**VRF**子命令是启用的在信任点下。

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

示例配置：

```
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

可以配置IOS PKI客户端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应该执行认证续订在正确地80%当前认证的寿命。

关键字**重新生成**阐明， IOS应该重新生成叫作Shadow密钥对的RSA密钥对在每次认证续订操作时。

这是自动注册工作情况：

- 配置瞬间**自动注册**，如果信任点验证， IOS将执行自动注册到服务器位于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认证的**自动注册**百分比安装了在信任点下。此计时器是可视的下**显示crypto pki计时器**命令。更多在计时器fuctions*是指*
- 续订功能技术支持来自PKI服务器。更多在此
  IOS PKI客户端执行续订的两种类型：
  含蓄续订：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被定义的自动注册百分比。即IOS使用一自签证书签署更新请求。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考虑到此功能在认证续订IOS期间使用当前活动身份认证签署续订证书请求。

应该保重，当配置自动注册百分比时。在配置的所有被测量的PKI客户端，如果情况出现身份认证到期在发出的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创建了反转认证后。*参考***PKI计时器依靠**部分

认证撤销检查

即一验证的Pki trustpoint包含CA证书的Pki trustpoint能够执行证书确认在IKE或SSL协商时，对等体认证对彻底的证书确认被服从。其中一个验证方法是检查对等体认证吊销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特定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回应者被呼叫的实体的通信信道，是一个指定的服务器由发出的CA。一个客户端例如IOS发送包含认证的序列号的一个请求被验证。OCSP回应者回应特定序列号的废止状态。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。

撤销检查可以被定义使用这些发出命令在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
```
默认情况下，配置使用crl，信任点执行撤销检查。

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

CRL高速缓冲存储器

对于CRL基于撤销检查，每证书确认可能触发新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)是去的，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缓存执行改进性能，并且缓存CRL考虑到CRL正确性。

使用两个次参数， CRL正确性被定义：**LastUpdate**，是上次CRL由发出的CA和**NextUpdate**发布，是时间是未来，当CRL文件的一个新版本由发出的CA时发布。

只要CRL是有效的， 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，长时间保留CRL在高速缓冲存储器可能是必要的。在IOS中被缓存的CRL能保留为，只要24小时，在CRL正确性到期后，使用在Pki trustpoint部分下的此命令，并且这可以被配置：

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

在某种状况下例如废除在CRL有效性周期的发出的CA证书， IOS能confiugured频繁地删除高速缓冲存储器。通过过早删除CRL， IOS被迫频繁地下载CRL保持CRL高速缓冲存储器最新状态。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，可以配置IOS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
```

建议的配置

与根CA和SUB CA配置的典型的CA配置是作为下面。示例也包括RA的保护的一种SUB CA配置。

使用2048位全面的RSA密钥对，此示例推荐的设置：
根CA有寿命8年
SUB CA有寿命3年
客户端证书自动地被发行一年，配置为认证续订请求。

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
```

证书注册

手动注册

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

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
```

第 1 步：首先请验证信任点(这可能在第2)步以后也执行。

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

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

步骤2.生成认证署名请求并且采取CSR对CA并且获得被授予的certificat ：

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

第 3 步：现在请通过终端导入被授予的认证：

```
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

第 1 步：首先从CA请导出发出的CA证书，在这种情况下是SUBCA认证。这被导入在上面step1期间在PKI客户端，即信任点认证。

```
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-----
```

Step 2.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接受从终端的一个认证署名请求，并且一次授予，身份验证数据在PEM格式被打印。

```
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在自动授予模式下，被授予的认证显示以上面PEM格式。当CA在手工的授予的模式下时，证书请求被标记作为**待定**，被赋予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-----
```

Note:一个SUB CA的手动注册对根CA的不是可能的。

Note:在于一个的禁用状态的CA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
```

DEFAULT模式证书请求授予手工：

```
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，是必须的，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：在信任点认证之后， PKI客户端可以为认证被登记。

Note:如果配置自动注册，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的一个RSA密钥对。如果它存在，为请求身份认证被拾起。否则， IOS创建2048位密钥对名为Pki KEY，然后使用它请求身份认证。
- IOS创建一个认证署名请求以PKCS10格式。
- 使用随机的对称密钥， IOS然后加密此CSR。使用接收人的公共密钥，随机的对称密钥被加密，是SUBCA (SUBCA的公共密钥可用归结于信任点认证)。 被加密的CSR、被加密的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证书。PKCS-7包围了数据，客户端使用的签署的认证，并且客户端的签名在PKCS-7签名数据信息包被汇集。这是编码的base64编码的，然后URL。数据发生的一滴被发送作为“在HTTP URI的消息”参数被发送到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与新要求产生关联的同样交易ID的证书请求。

Note:交易ID是公共密钥的MD5哈希，身份认证由客户端要求。

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

Note: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复制身份请求。然而， IOS PKI服务器用更新的请求在这种情况下取代请求。

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 =授予与被授予的认证一起由客户端，接受IOS定期执行GetCertInitial。

一旦被授予的认证被接受， IOS自动地安装它。