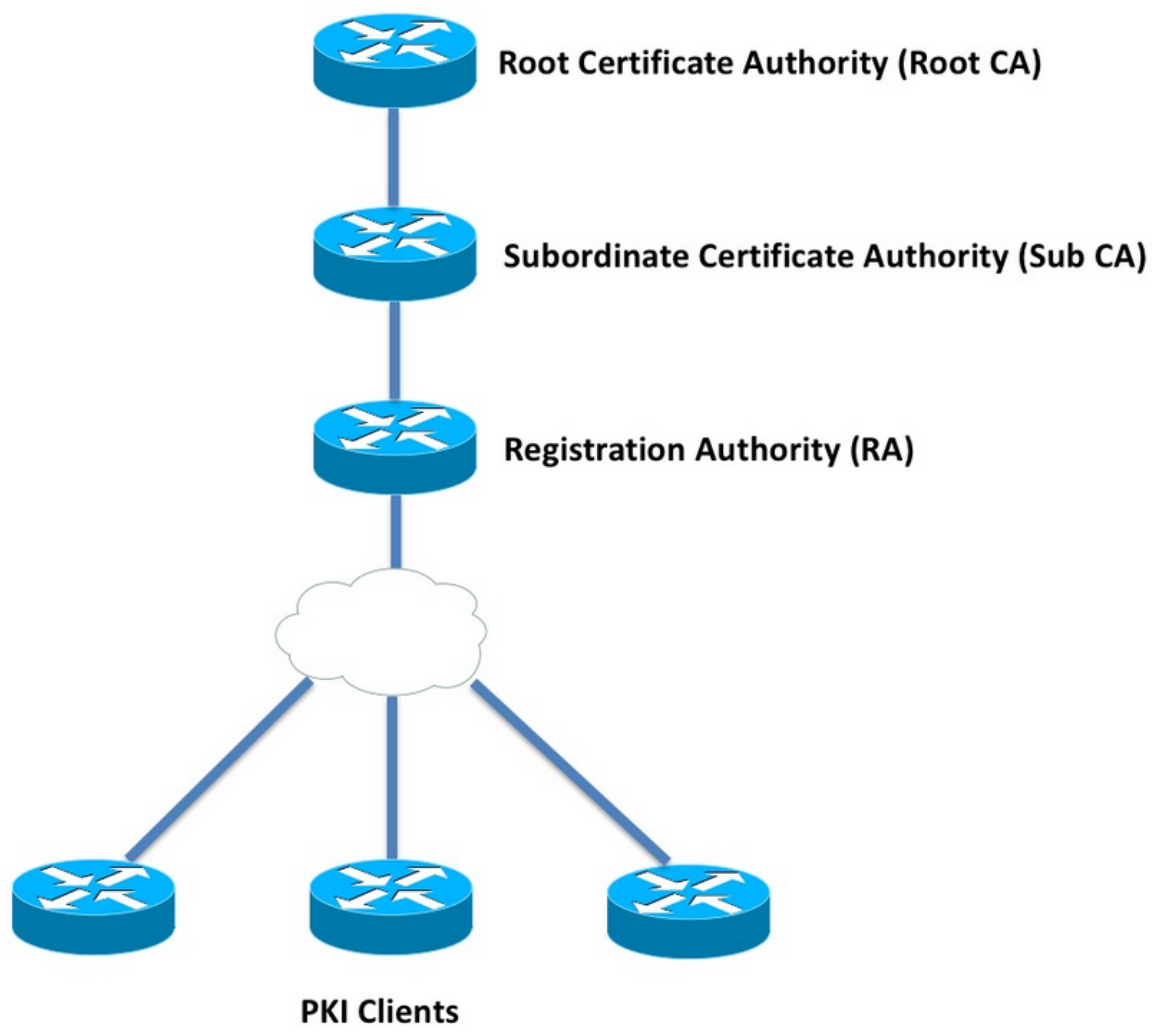
دليل نشر IOS PKI: التصميم الأولي والنشر

المحتويات

المقدمة

يصف هذا المستند وظائف خادم IOS PKI والعميل بالتفصيل. وهو يتناول اعتبارات التصميم والنشر الأولية لبطاقة IOS PKI.

البنية الأساسية ل PKI



جهة منح الشهادة

المرجع المصدق (CA)، ويشار إليه أيضا بخادم PKI في كل الوثيقة، هو كيان موثوق به يصدر تراخيص. تعتمد PKI على الثقة وتبدأ هرمية الثقة من المرجع المصدق الجذر (CA الجذر). لأن المرجع المصدق الجذر يقع في أعلى التسلسل الهرمي، فإنه يحتوي على شهادة

موقعة ذاتيا.

هيئة الشهادات الفرعية

تعرف كافة مراجع الشهادات الموجودة أسفل الجذر في التسلسل الهيكلي لثقة PKI باسم "مراجع الشهادات التابعة" (المرجع المصدق الفرعي). ومن الواضح أن شهادة من الدرجة الأولى تصدر عن هيئة المنافسة النزيهة، وهي درجة أعلى من ذلك.

لا تفرض PKI أي حد على عدد CAs الفرعية في تسلسل هرمي معين. ومع ذلك، قد يصبح من الصعب إدارة عملية نشر في المؤسسة تضم أكثر من ثلاثة مستويات من مراجع الشهادات.

هيئة التسجيل

تعرف PKI سلطة شهادة خاصة تعرف باسم سلطة التسجيل (RA)، وهي مسؤولة عن السماح لعملاء PKI بالتسجيل في مرجع مصدق فرعي أو المرجع المصدق الجذر. لا يصدر RA شهادات لعملاء PKI، بل يقرر أي عميل PKI يمكن أو لا يمكن إصداره بشهادة من قبل CA الفرعي أو المرجع المصدق الجذر.

يتمثل الدور الرئيسي للترخيص المتطور في إلغاء تحميل التحقق من صحة طلب شهادة العميل الأساسية من المرجع المصدق، وحماية المرجع المصدق من التعرض المباشر للعملاء. وبهذه الطريقة، تقف RA بين عملاء PKI و CA، مما يحمي CA من أي نوع من هجمات الحرمان من الخدمة.

عميل PKI

ويعرف أي جهاز يطلب شهادة تستند إلى زوج مفاتيح عام-خاص مقيم لإثبات هويته إلى أجهزة أخرى باسم عميل PKI.

يجب أن يكون عميل PKI قادرا على إنشاء زوج مفاتيح عام-خاص وأن تخزينه مثل RSA أو DSA أو ECDSA.

تعد الشهادة دليلا على هوية مفتاح عام معين وصلاحيته، شريطة وجود المفتاح الخاص المطابق على الجهاز.

خادم IOS PKI

الجدول 1. تطور ميزة خادم IOS PKI

الميزة	IOS [ISR-G1، ISR-G2]	IOS-XE [ASR1K، ISR4K]
خادم IOS CA/PKI	12.3(4)T	XE 3.14.0 / 15.5(1)S
إعادة تمرير شهادة خادم IOS PKI	12,4(1)T	XE 3.14.0 / 15.5(1)S
IOS PKI HA	15,0(1)M	NA [يتوفر تكرار ضمني بين RP]
IOS RA ل CA الخاص بالطرف الثالث	15,1(3)T	XE 3.14.0 / 15.5(1)S

قبل الوصول إلى تكوين خادم PKI، يجب على المسؤول فهم هذه المفاهيم الأساسية.

مصدر موثوق للزمن

والزمن هو أحد أسس البنية الأساسية للمرافق العمومية. تحدد ساعة النظام ما إذا كانت الشهادة صالحة أم لا. لذلك، في نظام التشغيل IOS، يجب جعل الساعة موثوقة وأي جديرة بالثقة. دون مصدر وقت موثوق به، قد يعمل خادم PKI كما هو متوقع، ومن المستحسن بشدة جعل الساعة على IOS موثوقة باستخدام هذه الأساليب:

NTP (بروتوكول وقت الشبكة)

تعد مزامنة ساعة النظام مع خادم الوقت هي الطريقة الصحيحة الوحيدة لجعل ساعة النظام جديرة بالثقة. يمكن تكوين موجه IOS كعميل NTP إلى خادم NTP معروف ومستقر في الشبكة:

```
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
```

كما يمكن تكوين IOS كخادم NTP، والذي سيضع علامة على ساعة النظام المحلية كموثوق بها. في نشر PKI على نطاق صغير، يمكن تكوين خادم PKI كخادم NTP لعملاء PKI:

```
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
```

وضع علامة "موثوق" على ساعة الجهاز

في IOS، يمكن وضع علامة على ساعة الجهاز كموثوق بها باستخدام:

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

ةعاس ةيقوثوم ىلع ظافحلا وه كلذب مايقلل يسيئرلا ببسلاو ،NTP عم اذه نيوكت نكمي
ةيناكمإ مدعو ،ةقاطلا عاطقنا ببسب لاثملا ليبس ىلع ،هجوملا ليمحت ةداعإ دنع ماظنلا
يدؤي امم ،لمعلا نع PKI تيقوت تادحو فقوتتس ،ةلحرملا هذه يف .NTP مداوخ ىلإ لوصولا
نامضك **ةعاسلا ميوقت** لمعيو .تاداهشلا هيجوت ةداعإ/ديدجت يف لشف تالاح ثودح ىلإ هرودب
.تالاحلا هذه لثم يف

ةيراطب تتام اذإ نمازتلا نع فقوتتس ماظنلا ةعاس نأ مهف مهملا نم ،اذه نيوكت ءانثأ
رثكأ رملأا اذه نيوكت نإف ،كلذ عمو .ةنمازتم ريغ ةعاس يف ةقثلا يف PKI أدبتسو ،ماظنلا
.قلاطلإا ىلع تقولل قوثوم ردصم دوجو مدع نم ايبسن انامأ

> **ةظحلام**: تمت ةفاضإ رملأا **CLOCK** حلاصلا **ميوقتلل** يف رادصلإا IOS-XE 3.10.0 / 15.3(3)S يلاتلا.

مسا فيضملا مساو لاجملا

يصوي نيوكتب مسا فيضم مساو لاجم ىلع Cisco IOS ةدحاوك نم تاوطخلا ىلولأا لبق
نيوكت يأ تامدخ تاذ ةلص ب PKI. متي مادختسإ مسا فيضم هجوملا مساو لاجملا يف
:ةيلاتلا تاهويرانيسلا

- متي قاقتشا مسا جوز حيتافم RSA يضارتفالا نع قيرط عيمجت مسا فيضملا مساو لاجملا
- دنع ليجستلا ةداهشل ،ام نوكتي مسا عوضوملا يضارتفالا نم ةمس مسا فيضملا مساو ريغ لهؤم ،وهو مسا فيضملا مساو لاجملا ماع.

:لاجملا مساو فيضملا مسا مادختسإ متي ال ،PKI مداخل ةبسنلاب

- نوكي مسا جوز حيتافملا يضارتفالا وه هسفن مسا مداخ PKI
- .PKI نوكتي مسا عوضوملا يضارتفالا نم CN، وهو سفن مسا مداخ

.لاجم مساو مسا فيضم نيوكت يف ةماعلا ةيصوتلا لثمتت

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

مداخ HTTP

متي نيكمت مداخ IOS PKI طقف يف ةلاح نيكمت مداخ HTTP. نم مهملا ةظحالم نأ هنإ اذإ مت
ليطعت مداخ PKI ببسب ليطعت مداخ HTTP، هنكميف ةعباتم حنم تابلطلا ريغ ةلصتملا
.[ربع terminal]. يلزم رفوت مداخ HTTP ةجلاعمل تابلط SCEP، لاسرإو تاباجتسا SCEP

:مادختساب HTTP مداخ IOS نيكمت متي

```
ip http server
```

ويمكن تغيير منفذ خادم HTTP الافتراضي من 80 إلى أي رقم منفذ صالح باستخدام:

```
ip http port 8080
```

اتصال HTTP Max

إحدى نقاط الاختناق، أثناء نشر IOS كخادم PKI باستخدام SCEP وه الحد الأقصى لاتصالات HTTP المتزامنة ومتوسط إتصالات HTTP في الدقيقة.
حاليا، يقتصر الحد الأقصى للاتصالات المتزامنة على خادم IOS HTTP على 5 بشكل افتراضي ويمكن زيادته إلى 16، وهو ما يوصى به بشدة في عملية نشر متوسطة النطاق:

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

تتيح عمليات تثبيت IOS هذه الحد الأقصى من إتصالات HTTP المتزامنة حتى 1000:

- نظام التشغيل UniversalK9 IOS مع مجموعة ترخيص UCK9

يتم تغيير واجهة سطر الأوامر (CLI) للقبول وسيطة رقمية بين 1 و 1000

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

يسمح خادم IOS HTTP بإجراء 80 اتصال في الدقيقة [580 في حالة إصدارات IOS حيث يمكن زيادة الحد الأقصى لجلسات العمل المتزامنة ل HTTP إلى 1000] وعندما يتم الوصول إلى هذا الحد في غضون دقيقة، يبدأ موزع رسائل IOS HTTP في إتصالات HTTP الواردة من الخلال إيقاف تشغيل موزع الرسائل لمدة 15 ثانية. يؤدي هذا إلى إسقاط طلبات اتصال العميل بسبب **بلوغ حد قائمة انتظار اتصال** TCP. يمكن العثور على مزيد من المعلومات حول هذا [الموضوع هنا](#)

زوج مفاتيح RSA

يمكن إنشاء زوج مفاتيح RSA لوظائف خادم PKI على IOS تلقائيا أو إنشاؤه يدويا.
أثناء تكوين خادم PKI، يقوم IOS تلقائيا بإنشاء TrustPoint بنفس اسم خادم PKI لتخزين شهادة خادم PKI.

زوج مفاتيح RSA الخاص بخادم PKI الذي يتم إنشاؤه يدويا:

الخطوة 1. قم بإنشاء زوج مفاتيح RSA بنفس اسم خادم PKI:

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

الخطوة 2. قبل تمكين خادم PKI، قم بتعديل TrustPoint الخاص بخادم PKI:

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

ملاحظة: ال يتم أخذ قيمة معدل زوج مفاتيح RSA المشار إليها في TrustPoint الخاصة

بخادم PKI في الاعتبار حتى الإصدار 15.4(3)M4 من IOS، وهذا تحذير معروف. مع دل المفاتح الافتراضي وه 1024 بت.

زوج مفاتيح RSA الخاص بخادم PKI الذي يتم إنشاؤه تلقائيا:

عند تمكين خادم PKI، يقوم IOS تلقائيا بإنشاء زوج مفاتيح RSA بنفس اسم خادم PKI، وحجم المعامل الرئيسي وه 1024 بت.

بدءا من IOS الإصدار 15.4(3)M5، يعمل هذا التكوين على إنشاء زوج مفاتيح RSA مع <LABEL> حيث سيكون الاسم وقوة المفاتح كما وه الحال وفقا لمحول <mod> المحدد.

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

مفسد

يجب أن يسمح خادم CSCuu73408 IOS PKI بحجم المفاتح غير الافتراضي لذاكرة المرور.
يجب أن يسمح خادم CSCuu73408 IOS PKI بحجم المفاتح غير الافتراضي لقاعدة المرور.

معيار الصناعة الحالي وه إستخدام زوج مفاتيح RSA إصدار 2048 بت كحد أدنى.

إعتبار مؤقت المرور التلقائي

حاليا، ال يقوم خادم IOS PKI بإنشاء شهادة مرور بشكل افتراضي، ويجب تمكينها بشكل صريح تحت خادم PKI باستخدام أمر **التمرير التلقائي <days-قبل انتهاء الصلاحية>**. يتم شرح المزيد حول إعادة توجيه الشهادة في

يحدد هذا الأمر عدد الأيام قبل انتهاء صلاحية شهادة PKI Server/CA في حالة قيام IOS بإنشاء شهادة مرجع مصدق تمرير (CA). الحظ أن شهادة مرجع مصدق المرور يتم تنشيطها بمجرد انتهاء صلاحية شهادة المرجع المصدق النشطة الحالية. القيمة الافتراضية حاليا هي 30 يوما. يجب تعيين هذه القيمة إلى قيمة معقولة وفقا لفترة وجود شهادة CA، وهذا بدوره يؤثر على تكوين مؤقت التسجيل التلقائي على عميل PKI.

> **ملاحظة**: يجب تشغيل مؤقت إعادة التوجيه التلقائي دائما قبل مؤقت التسجيل التلقائي على العميل أثناء تمرير شهادة CA والعميل [المعروف باسم]

إعتبارات CRL

تدعم البنية الأساسية ل IOS PKI طريقتين لتوزيع CRL:

نشر CRL على خادم HTTP

يمكن تكوين خادم IOS PKI لنشر ملف CRL إلى موقع محدد على خادم HTTP باستخدام هذا الأمر تحت خادم PKI:

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

ويمكن تكوين خادم PKI لدمج وقوع CRL هذا في جميع شهادات عميل PKI باستخدام هذا الأمر تحت خادم PKI:

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

بروتوكول SCEP GetCRL

يقوم خادم IOS PKI تلقائيا بتخزين ملف CRL في موقع قاعدة البيانات المحدد، والذي يعد مخادم مضيف بشكل افتراضي، ومن المستحسن بشدة الاحتفاظ بنسخة على خادم NVRAM باستخدام هذا الأمر تحت خادم PKI:

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

بشكل افتراضي، الخادم IOS PKI دمج موقع CDP في شهادات عميل PKI. إذا تم تكوين الخادم IOS PKI لإجراء التحقق من الإبطال، ولكن الشهادة التي تم التحقق من صحتها لعميل IOS PKI تحتوي على CDP مضمن فيها، ويتم تكوين التحقق من صحة CA TrustPoint باستخدام موقع CA (باستخدام http://<CA-Server-IP أو FQDN>)، فإن IOS يرجع إلى بروتوكول GetCRL المستند إلى SCEP بشكل افتراضي.
يقوم SCEP GetCRL بتنفيذ استرداد CRL من خلال تنفيذ HTTP GET على عنوان URL هذا:

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

> **ملاحظة**: في واجهة سطر الأوامر (CLI) عبر نظام IOS، قبل الإدخال؟، اضغط على تسلسل المفاتيح **Ctrl+V**.

يمكن لخادم IOS PKI أيضا دمج عنوان URL هذا كموقع CDP. إن الميزة من القيام بذلك تتألف من شقين:

- وهو يضمن إمكانية قيام جميع عملاء PKI المستندة إلى SCEP بخلاف IOS باسترداد CRL.
- بدون بروتوكول CDP مدمج، يتم توقيع رسائل طلب GetCRL الخاصة ب IOS (باستخدام شهادة موقتة ذاتية التوقيع) كما هو محدد في مسودة بروتوكول SCEP. ومع ذلك، لا يزال بروتوكول CDP طريقة لتضمين عنوان URL، ومن خلال استرداد CRL، يمكن تجنب توقيع طلبات GetCRL.

وضع CRL

يمكن التحكم في الوضع الافتراضي لخادم IOS PKI باستخدام هذا الأمر تحت خادم PKI:

```
crypto pki server <PKI-SERVER-Name>
 lifetime crl <0 - 360>
```
القيمة بالساعات. بشكل افتراضي، يتم تعيين مدة البقاء ل CRL على 6 ساعات. طبقا لمدى تكرار إبطال الشهادات، فإن ضبط عمر CRL إلى قيمة مثلى يزيد من أداء إسترجاع CRL في الشبكة.

اعتبارات قاعدة البيانات

يستخدم خادم IOS PKI NVRAM كموقع قاعدة البيانات الافتراضي، ومن المستحسن بشدة إستخدام خادم FTP أو TFTP أو SCP كموقع قاعدة البيانات. بشكل افتراضي، يقوم خادم IOS PKI بإنشاء ملفين:

- <server-name>.ser - يحتوي هذا على آخر رقم تسلسلي تم إصداره من قبل CA بالنظام السداسي العشري. الملف بتنسيق نص عادي، ويحتوي على هذه المعلومات:
  db_version = 1
  last_serial = 0x4

- <server-name>.crl - هذا هو ملف CRL المرمز من قبل CA

يقوم خادم IOS PKI بتخزين المعلومات في قاعدة البيانات على 3 مستويات قابلة للتكوين:

- الأدنى - هذا هو المستوى الافتراضي، وعلى هذا المستوى ال يتم إنشاء أي ملف في قاعدة البيانات، وبالتالي ال تتوفر أي معلومات على خادم CA فيما يتعلق بشهادات العميل الممنوحة في الماضي.
- الأسمى - على هذا المستوى، يقوم خادم IOS PKI بإنشاء ملف باسم <serial-number>.cnm إلى الرقم التسلسلي لكل شهادة عميل صادرة، حيث يشير الاسم <serial-number> اسم الموضوع وتاريخ انتهاء CNM هذا على اسم ملف يحتوي و الصادرة العميل لشهادة الصالحية شهادة العميل.
- كامل - عند هذا المستوى، يقوم خادم IOS PKI بإنشاء ملفين لكل شهادة عميل يتم إصدارها:
- <serial-number>.cnm
- <serial-number>.crt

هنا، ملف CRT هو ملف شهادة العميل، وهو DER المشفر.

وهذه النقاط مهمة:

- قبل إصدار شهادة عميل، يشير خادم IOS PKI إلى <server-name>.ser لتحديد الرقم التسلسلي للشهادة واستخراجه.
- مع تعيين مستوى قاعدة البيانات إلى "أسمى" أو "الكامل"، يجب كتابة <Serial-Number>.cnm و<Serial-Number>.crt إلى قاعدة البيانات قبل إرسال الشهادة المنحة/الصادرة إلى العميل
- مع تعيين عنوان URL لقاعدة البيانات إلى "أسمى" أو "الكامل"، يجب أن يحتوي عنوان URL لقاعدة البيانات على مساحة كافية لحفظ الملفات. وبالتالي فإن التوصية هي تكوين خادم ملف خارجي [FTP أو TFTP أو SCP] كعنوان URL لقاعدة البيانات.
- مع تكوين عنوان URL لقاعدة البيانات الخارجية، من الضروري للغاية التأكد من إمكانية

الوصول إلى خادم الملفات أثناء عملية منح الشهادة، وهو ما قد يؤدي بالفعل إلى ذلك إلى وضع علامة على خادم CA على أنه معطل. والتدخل اليدوي مطلوب لإعادة خادم CA إلى الإنترنت.

أرشيف قاعدة البيانات

أثناء نشر خادم PKI، من المهم مراعاة سيناريوهات الفشل والاستعداد، في حالة حدوث فشل في برنامج تشغيل الجهاز. وهناك طريقتان لتحقيق هذه الغاية:

1. نُسخ الاحتياطية/معلومات مكررة
   في هذه الحالة، يعمل جهازان أو وحدات معالجة كأجهزة إحتياطية نشطة توفير التكرار. يمكن تحقيق إمكانية الوصول إلى خادم IOS PKI باستخدام موجهات ISR (ISR G1 و ISR G2 اللذين تم تمكينهما من HSRP كما هو موضح في

   لا تتوفر للأنظمة المستندة إلى IOS XE [ISR4K و ASR1k] خيار تكرار الجهاز. ومع ذلك، في ASR1k يتوفر تكرار Inter-RP بشكل افتراضي.
2. أرشفة زوج مفاتيح وملفات CA Server
   يوفر IOS منشأة لأرشفة زوج مفاتيح PKI Server والشهادة. يمكن عمل الأرشفة باستخدام نوعين من الملفات:
   PEM - IOS يقوم بإنشاء ملفات بتنسيق PEM لتخزين مفتاح RSA العام، ومفتاح RSA الخاص المشفر، وشهادة خادم CA. تتم أرشفة زوج المفاتيح المرور والشهادات تلقائيPKCS12 - يقوم IOS بإنشاء ملف PKCS12 واحد يحتوي على شهادة خادم CA ومفتاح RSA الخاص المتوافق المشفر باستخدام كلمة مرور.

يمكن تمكين أرشفة قاعدة البيانات باستخدام هذا الأمر ضمن خادم PKI:

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

من الممكن أيضا تخزين الملفات التي تمت أرشفتها إلى خادم منفصل، ربما باستخدام بروتوكول آمن (SCP) باستخدام الأمر التالي تحت خادم PKI:

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

من بين جميع الملفات الموجودة في قاعدة البيانات باستثناء الملفات التي تمت أرشفتها والملف .ser، تكون جميع الملفات الأخرى في نص واضح الوحدات للكشف أي تهديد حقيقي في حالة فقدانها، وبالتالي يمكن تخزينها على خادم منفصل دون تكبد نفقات إضافية كبيرة أثناء كتابة الملفات، على سبيل المثال خادم TFTP.

IOS AS-CA الفرعي

يوفر خادم IOS PKI بشكل افتراضي أداء دور مرجع مصدق جذري. لتكوين خادم PKI عبات (CA فرعي)، قم أولا بتمكين هذا الأمر تحت قسم تكوين خادم PKI (قبل تمكين خادم PKI):

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

باستخدام هذا التكوين ل URL CA الجذر تحت TrustPoint الخاص بخادم PKI:

```
crypto pki trustpoint <Sub-PKI-SERVER-Name>
```

```
 enrollment url <Root-CA URL>
```
يؤدي تمكين خادم PKI هذا الآن إلى تشغيل هذه الأحداث:

- تتم مصادقة نقطة ثقة خادم PKI من أجل تثبيت شهادة المرجع المصدق الجذر.
- بعد مصادقة المرجع المصدق الجذر، يقوم IOS بإنشاء CSR للمرجع المصدق الفرعي وبأرسله إلى المرجع المصدق الجذر [x509 قيد أساسي يحتوي على CA:TRUE flag]

يقوم IOS بصرف النظر عن وضع المنحة الذي تم تكوينه على المرجع المصدق الجذر، وبدلا من ذلك يضع طلب شهادة المرجع المصدق (أو RA) في قائمة انتظار معلقة. يجب على المسؤول منح شهادات CA يدويا.
لعرض طلب الشهادة المعلق ومعرف الطلب:

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

للموافقة على الطلب:

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

- باستخدام هذا، تقوم عملية إستطلاع SCEP (GetCertInitial) التالية بتنزيل شهادة CA الفرعية وتثبيتها على الموجه، مما يمكن خادم PKI التابع

IOS RA

يمكن تكوين خادم IOs PKI كمرجع تسجيل إلى مرجع مصدق مرؤوس أو جذر محدد. لتكوين خادم PKI كمرجع تسجيل، قم بإصدار الأمر ضمن قسم تكوين خادم PKI (قبل تمكين خادم PKI):

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

بعد ذلك، قم بتكوين عنوان URL ل CA تحت TrustPoint الخاص بخادم PKI. وهذا يشير إلى المرجع المصدق الذي تتم حمايته بواسطة RA:

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

تحت RA، والتي تصدر هيئة التسجيل شهادات، ومن ثم يشترط تكوين **اسم الموضوع** ضمن RA لكي تكون التكوين فعالا حتى إذا تم تكوينه. يتم تكوين اسم الموضوع ل RA باستخدام الأمر **subject-name**. من المهم تكوين **OU = ioscs RA** كجزء من اسم TrustPoint الموضوع لتحديد IOS CA ل IOS RA، أي لتحديد طلبات الشهادات المصرح بها من قبل IOS RA.

يمكن أن يعمل IOS كمرجع تسجيل لجهات خارجية مثل Microsoft CA، وللحفاظ على توافقه، يجب تمكين IOS RA باستخدام هذا الأمر ضمن قسم تكوين خادم PKI (قبل تمكين خادم PKI):

```
mode ra transparent
```

في وضع RA الافتراضي، يقوم IOS بتوقيع طلبات العميل [PKCS#10] باستخدام شهادة RA. تشير هذه العملية إلى خادم IOS PKI أن طلب الشهادة تم توجيهه بواسطة RA.

مع وضع RA الشفاف، يقوم IOS بإعادة توجيه طلبات العميل في تنسيقها الأصلي دون تقديم شهادة RA، وهذا متوافق مع Microsoft CA المعروف.

عميل IOS PKI

يعد أحد أهم كيانات التكوين في عميل IOS PKI وهو TrustPoint. يتم شرح معلمات تكوين TrustPoint بالتفصيل في هذا القسم.

مصدر موثوق للزمن

وكما أشير من قبل، فإن مصدر الوقت الرسمي وهو مطلب على عميل شركة PKI أيضا. يمكن تكوين عميل IOS PKI كعميل NTP باستخدام التكوين التالي:

```
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
```

اسم المضيف واسم المجال

توصية عامة هي تكوين اسم مضيف واسم مجال على الموجه:

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

زوج مفاتيح RSA

في عميل IOS PKI، يمكن إنشاء زوج مفاتيح RSA لتسجيل نقطة ثقة معينة تلقائيا أو إنشاؤه يدويا.

تتضمن عملية إنشاء مفاتيح RSA التلقائية ما يلي:

- يعمل IOS بشكل افتراضي على إنشاء زوج مفاتيح RSA 512 بت
- اسم زوج المفاتيح الذي تم إنشاؤه تلقائيا هو hostname.domain-name، وهو اسم مضيف الجهاز المدمج مع اسم مجال الجهاز
- لم يتم تمييز زوج المفاتيح الذي تم إنشاؤه تلقائيا على أنه قابل للتصدير.

تتضمن عملية إنشاء مفاتيح RSA التلقائية ما يلي:

- وبشكل إختياري، يمكن إنشاء زوج مفاتيح RSA للأغراض العامة بقوة منا سبة يدويا باستخدام:
- `crypto key generate rsa general-keys label <LABEL> modulus < MOD> [exportable]`
  هنا، التسمية - اسم زوج مفاتيح RSA
  MOD - معدل مفاتيح RSA وأ القوة في وحدات بت بين 360 حتى 4096، والتي هي 4096 وأ 2048 وأ 1024 وأ 512 تقليديا.

تتمثل ميزة إنشاء زوج مفاتيح RSA يدويا في القدرة على تعيين زوج المفاتيح على أنه قابل للتصدير، مما يسمح بالتالي بتصدير شهادة الهوية بالكامل، والتي يمكن استعادتها بعد ذلك على جهاز آخر. بيد أنه ينبغي للمرء أن يفهم الآثار الأمنية المترتبة على هذا الإجراء.

- يتم ربط زوج مفاتيح RSA بنقطة اتصال قبل التسجيل باستخدام هذا الأمر

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

هنا، إذا كان زوج مفاتيح RSA المسمى <LABEL> موجودا بالفعل، فسيتم التقاطه أثناء تسجيل TrustPoint.

إذا لم يكن زوج مفاتيح RSA المسمى <LABEL> موجودا، فسيتم تنفيذ أحد الإجراءات التالية أثناء التسجيل:

- في حالة عدم تمرير وسيطة <mod>، يتم إنشاء زوج مفاتيح 512 بت باسم <LABEL>.
- في حالة تمرير وسيطة <mod> واحدة، يتم إنشاء زوج مفاتيح الغرض العام ل <mod> بت باسم <LABEL>
- في حالة تمرير وسيطتين <mod>، يتم إنشاء زوج مفاتيح توقيع <mod> بت واحد وزوج مفاتيح تشفير <mod> بت واحد، وكلاهما باسم <LABEL>

TrustPoint

نقطة الثقة هي حاوية مجردة لحمل شهادة في IOS. يمكن لنقطة ثقة واحدة تخزين شهادتين فعالتين في أي وقت:

- شهادة المرجع المصدق - يعرف تحميل شهادة المرجع المصدق في نقطة موثوق بها بعملية مصادقة TrustPoint.
- تعريف شهادة المعرف التي تم إصدارها من قبل المرجع المصدق - تحميل أو إستيراد شهادة معرف إلى نقطة ثقة معينة بعملية تسجيل نقطة الثقة.

يعرف تكوين TrustPoint باسم نهج الثقة، وهذا يحدد ما يلي:

- ما هي شهادة المرجع المصدق التي تم تحميلها في TrustPoint؟
- ما هو المرجع المصدق الذي يتم الانضمام إليه في TrustPoint؟
- كيف يقوم IOS بتسجيل نقطة الثقة؟
- كيف يتم التحقق من صحة الشهادة التي تم إصدارها من المرجع المصدق (CA) المحدد [المحملة في TrustPoint]؟

يتم شرح المكونات الرئيسية لنقطة الثقة هنا.

وضع التسجيل

يمكن تنفيذ وضع تسجيل TrustPoint، والذي يحدد أيضا وضع مصادقة TrustPoint، عبر 3 طرق رئيسية:

1. التسجيل الطرفي - الطريقة اليدوية لإجراء مصادقة TrustPoint وتسجيل الشهادة باستخدام Copy Paste في وحدة CLI الطرفية.
2. تسجيل SCEP - مصادقة TrustPoint والتسجيل باستخدام SCEP عبر HTTP.
3. ملف تعريف التسجيل - هنا، يتم تحديد طرق المصادقة والتسجيل بشكل منفصل. بجانب طرق تسجيل SCEP والطرفية، توفر ملفات تعريف التسجيل خيارا لتحديد

أوامر HTTP/TFTP لإجراء استرداد الملف من الخادم، والذي تم تعريفه باستخدام عنوان URL للمصادقة أو التسجيل ضمن ملف التعريف.

الواجهة المصدر و VRF

تستخدم مصادقة TrustPoint والتسجيل عبر HTTP (SCEP) أو TFTP (ملف تعريف التسجيل) إدخال/إخراج الملف. يمكن الحصول على عمليات تبادل الحزم لتنفيذ عمليات IOS في نظام ملف VRF و واجهة مصدر معينة من هذه.

في حالة تكوين TrustPoint الكلاسيكي، يتم تمكين هذه الوظيفة باستخدام **واجهة المصدر** والأوامر الفرعية **vrf** تحت TrustPoint.

في حالة ملفات تعريف التسجيل، **واجهة المصدر** و**التسجيل |** توفر أوامر **<vrf-name>url** نفس الوظائف. **<http/tftp://Server-location> vrf للمصادقة**

مثال تشكيل:

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

crypto pki trustpoint MGMT
 source interface Ethernet0/0
 vrf MGMT
```

أو

```
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
```

التسجيل والتجديد التلقائي للشهادات

يمكن تكوين عميل IOS PKI لإجراء التسجيل والتجديد التلقائي باستخدام هذا الأمر ضمن قسم TrustPoint لـ PKI:

```
crypto pki trustpoint MGMT
auto-enroll <percentage> <regenerate>
```

هنا، يصف **الأمر <percentage> [regenerate]** للتسجيل التلقائي على أن يقوم IOS بتجديد الشهادة بنسبة 80٪ تمامًا من العمر الافتراضي للشهادة الحالية.

تذكر الكلمة الأساسية **regenerate** أنه يجب على IOS إعادة إنشاء زوج مفاتيح RSA المعروف باسم زوج مفاتيح الظل أثناء كل عملية تجديد الشهادة.

هذا هو سلوك التسجيل التلقائي:

- يتم تكوين اللحظة **التسجيل التلقائي**، في حالة مصادقة TrustPoint، سيقوم IOS بإجراء تسجيل تلقائي إلى الخادم الموجود على عنوان URL المذكور في أمر **url** ضمن قسم TrustPoint لـ PKI أو ضمن ملف تعريف التسجيل. **التسجيل**

- في اللحظة التي يتم فيها تسجيل نقطة الثقة بخادم PKI أو مرجع مصدق، تتم تهيئة تجديد أو مؤقت لظ على عميل PKI استنادا إلى النسبة المئوية **للتسجيل التلقائي** لشهادة الهوية الحالية المثبتة ضمن TrustPoint. يكون هذا المؤقت مرئيا تحت أمر **show crypto pki timer**. تشير المزيد من المعلومات حول إعدادات المؤقت إلى
- يأتي عدم قدرة التجديد من خادم PKI. المزيد حول هذا في
يجري عميل IOS PKI نوعين من التجديد:
تجديد ضمني: إذا لم يرسل خادم PKI "تجديد" كقدرة مدعومة، يقوم IOS بإجراء تسجيل جديد أولي في النسبة المئوية المحددة للتسجيل التلقائي. على سبيل المثال، يستخدم IOS شهادة موقعة ذاتيا لتوقيع طلب التجديد.تجديد صريح: عندما يعلم خادم PKI ميزة "تجديد" عن إعلانه كقدرة مدعومة، فإن IOS يأخذ هذه القدرة في الاعتبار أثناء تجديد الشهادة، أي أن IOS يستخدم شهادة الهوية النشطة الحالية لتوقيع طلب شهادة التجديد.
يجب توخي الحذر أثناء تكوين النسبة المئوية للتسجيل التلقائي. في أي عميل PKI محدد في النشر، إذا تأشن حالة تنتهي فيها شهادة الهوية في نفس الوقت الذي تنتهي فيه صلاحية شهادة CA المصدرة، فيجب أن تقوم مهمة التسجيل التلقائي بتشغيل عملية تجديد [الظل] بعد أن يقوم CA بإنشاء شهادة إعادة توجيه. ارجع إلى قسم **تعيينات تقوم PKI في**

إلغاء الشهادة- التحقق من صحة الشهادة

ويمكن لنقطة الثقة PKI المصدق عليها، أي نقطة الثقة PKI التي تحتوي على شهادة CA، إجراء التحقق من صحة الشهادة أثناء تفاوضات IKE أو SSL، حيث تخضع شهادة النظير للتحقق من الشامل من الشهادة. تتمثل إحدى طرق التحقق من الصحة في التحقق من حالة إبطال شهادة النظير باستخدام إحدى الطريقتين التاليتين:

- قائمة إلغاء الشهادة (CRL) - هذا ملف يحتوي على الأرقام التسلسلية للشهادات التي تم إبطالها بواسطة مرجع مصدق معين. يتم توقيع هذا الملف باستخدام شهادة CA الإصدار. يتضمن أسلوب CRL تنزيل ملف CRL باستخدام HTTP أو LDAP.

- بروتوكول حالة الشهادة عبر الإنترنت (OCSP) - ينشئ IOS قناة اتصال مع كيان يسمى OCSP Responder، وهو خادم معين بواسطة المرجع المصدق (CA) الإصدار. يرسل عميل IOS طلبا يحتوي على الرقم التسلسلي للشهادة التي يتم التحقق منها. يستجيب المستجيب OCSP بحالة إبطال الرقم التسلسلي المحدد. يمكن إنشاء قناة الاتصال باستخدام أي بروتوكول نقل/تطبيق مدعوم، والذي يكون عادة HTTP.

يمكن تحديد التحقق من الإبطال باستخدام هذا الأمر ضمن قسم PKI TrustPoint:

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

بشكل افتراضي، يتم تكوين TrustPoint لإجراء فحص الإبطال باستخدام CRL.

يمكن إعادة ترتيب الطرق، ويتم إجراء التحقق من حالة الإبطال في الأمر المحدد. يتجاوز الأسلوب "none" التحقق من الإبطال.

ذاكرة التخزين المؤقت ل CRL

لتنزيل أي فحص إبطال مستند إلى CRL، قد يؤدي كل تدقيق للشهادة إلى تشغيل تنزيل ملف CRL جديد. ونظرا لأن ملف CRL يكبر أو إذا كانت نقطة توزيع CRL (CDP) بعيدة، فإن

تنزيل الملف أثناء كل عملية تحقق تحقق أداء البروتوكول الذي يعتمد على التحقق من صحة الشهادة. وبالتالي، يتم إجراء التخزين المؤقت ل CRL لتحسين الأداء، ويأخذ التخزين المؤقت ل CRL في الاعتبار صحة CRL.

يتم تحديد صلاحية CRL باستخدام معاملات الوقت: **LastUpdate**، وهو آخر مرة تم فيها نشر CRL بواسطة CA الإصدار، و**NextUpdate**، وهو الوقت الذي يتم فيه نشر إصدار جديد من ملف CRL بواسطة CA الإصدار.

يقوم IOS بتخزين كل CRL تم تنزيله طالما أن CRL صالح. ومع ذلك، في ظل ظروف معينة مثل عدم إمكانية الوصول إلى بروتوكول CDP بشكل مؤقت، قد يكون من الضروري الإبقاء على الوصول في الحكم الدائمة على (CRL) في ذاكرة التخزين المؤقت لفترة زمنية طويلة. في IOS، يمكن الاحتفاظ بقائمة التحكم في الوصول (CRL) المخزنة مؤقتا لمدة 24 ساعة بعد انتهاء صلاحية CRL، ويمكن تكوين هذا باستخدام الأمر هذا ضمن قسم PKI TrustPoint:

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

في ظروف معينة مثل إصدار شهادات إبطال CA ضمن فترة صلاحية CRL، يمكن أن يقوم IOS بتكوين حذف ذاكرة التخزين المؤقت بشكل أكثر تواترا. فحذف CRL قبل الأوان، يتم فرض IOS لتنزيل CRL بشكل أكثر تواترا لإبقاء ذاكرة التخزين المؤقت CRL محدثة. يتوفر خيار التكوين هذا ضمن قسم PKI TrustPoint:

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

وأخيرا، يمكن تكوين IOS لعدم تخزين ملف CRL مؤقتا باستخدام هذا الأمر تحت قسم PKI TrustPoint:

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

التكوين الموصى به

يكون نشر المرجع المصدق النموذجي مع تهيئة المرجع المصدق الجذر والتكوين الفرعي كما هو موضح أدناه. ويتضمن المثال أيضا تكوين CA الفرعي المحمي بواسطة RA.

مع زوج مفاتيح RSA 2048 بت بعرض اللوحة، يوصي هذا المثال بإعداد حيث:
عمر Root-CA 8 سنوات
عمر CA الفرعي 3 سنوات
يتم إصدار شهادات العميل لمدة عام، يتم تكوينها لطلب تجديد الشهادة تلقائيا.

CA الجذر - التكوين

```
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
```

SUBCA بدون RA - التكوين

```
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
```

SUBCA مع RA - التكوين

```
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
```

RA ل Subca - التكوين

```
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
```

تسجيل الشهادة

التسجيل اليدوي

يتضمن التسجيل اليدوي إنشاء CSR دون اتصال على عميل PKI، والذي يتم نسخه يدويا إلى CA. يقوم المسؤول بتوقيع الطلب يدويا، والذي يتم إستيراده بعد ذلك إلى العميل.

عميل PKI

تكوين عميل PKI:

```
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
```

الخطوة 1. قم بالمصادقة على TrustPoint (يمكن أيضا تنفيذ ذلك بعد الخطوة 2).

```
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
```

الخطوة 2. إنشاء طلب توقيع الشهادة وأخذ CSR إلى CA والحصل على الشهادة المميزة:

```
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
```

الخطوة 3. قم الآن باستيراد الشهادة المميزة عبر وحدة طرفية:

```
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
```

خادم PKI

الخطوة 1. قم أولا بتصدير شهادة المرجع المصدق المرجع المصدق من المرجع المصدق، والتي هي شهادة المرجع المصدق الفرعي SUBCA. يتم إستيراد هذا أثناء الخطوة 1 أعلاه على عميل PKI، أي في هذه الحالة مصادقة TrustPoint.

```
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-----
```

الخطوة 2. بعد الخطوة 2-على عميل PKI، قم بأخذ CSR من العميل ووفره للتوقيع على Subca باستخدام هذا الأمر:

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

يقترح هذا الأمر أن تقبل SUBCA طلب توقيع شهادة من المحطة الطرفية، وبمجرد منحه، تطبع بيانات الشهادة بتنسيق PEM.

```
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
```

إذا كان المرجع المصدق في وضع المنحة التلقائية، فإن الشهادة المموقعة يتم عرضها بتنسيق PEM أعلاه. عندما يكون المرجع المصدق في وضع منح يدوي، يتم وضع علامة على الانتظار، ويتم تعيين قيمة معرف له، ويتم وضعه في قائمة انتظار **معلق** من أجل طلب الشهادة على طلبات التسجيل.

```
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
```

الخطوة 3. منح هذا الطلب يدويا باستخدام هذا الأمر:

```
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-----
```

ملاحظة: لا يمكن التسجيل اليدوي للمرجع المصدق الفرعي إلى المرجع المصدق الجذر.

ملاحظة: يمكن ل CA في حالة معطلة بسبب تعطيل خادم HTTP منح طلبات الشهادة يدويا.

التسجيل باستخدام SCEP

تكوين عميل PKI وه:

```
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
```

تكوين خادم PKI وه:

```
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/
```

الوضع الافتراضي لمنح طلب الشهادة هو اليدوي:

```
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
```

منحة يدوية

الخطوة 1. عميل PKI: كخطوة أولى، تكون إلزامية، مصادقة نقطة الثقة على عميل PKI:

```
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.
```

الخطوة 2. PKI-Client: بعد مصادقة TrustPoint، يمكن تسجيل عميل PKI للحصول على شهادة.

> **ملاحظة**: في حالة تكوين التسجيل التلقائي، سيقوم العميل تلقائيا بإجراء التسجيل.

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

وراء الكواليس تحدث هذه الأحداث:

- يبحث IOS عن زوج مفاتيح RSA باسم PKI-Key. إذا كان موجودا، يتم انتقاؤه لطلب شهادة هوية. وإذا لم تكن هناك مساحة، يقوم IOS بإنشاء زوج مفاتيح 2048 بت باسم PKI-Key، ثم إستخدامه لطلب شهادة هوية.
- يقوم IOS بإنشاء طلب توقيع شهادة بتنسيق PKCS10.
- ثم يقوم IOS بتشفير CSR هذا باستخدام مفتاح متماثل عشوائي. يتم تشفير المفتاح المتماثل العشوائي باستخدام المفتاح العام الخاص بالمستلم، وهو مفتاح

SUBCA (يتوفر المفتاح العام ل SUBCA's بسبب مصادقة TrustPoint). يتم وضع CSR المشفر والمفتاح العشوائي المتماثل المشفر ومعلومات المستلم على هيئة بيانات PKCS#7 المضمنة.

- يتم توقيع هذه البيانات المغلفة ل PKCS#7 باستخدام شهادة ذاتية التوقيع مؤقتة أثناء التسجيل الأولي. يتم تجميع بيانات PKCS#7 المغلفة وشهادة التوقيع المستخدمة من قبل العميل وتوقيع العميل على هيئة حزمة بيانات PKCS#7 الموقعة. هذا هو base64 مشفر، مثل URL مشفر. يتم إرسال النقطة الكبيرة للبيانات الناتجة كوسيطة "رسالة" في URI ل HTTP إلى CA:

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

الخطوة 3. خادم PKI:

عندما يستقبل خادم IOS PKI الطلب، فإنه يتحقق مما يلي:

1. التحقق مما إذا كانت قاعدة بيانات طلب التسجيل تحتوي على طلب بنفس شهادة معرف الحركة المقترن بالطلب الجديد.

ملاحظة: معرف المعاملة هو تجزئة MD5 للمفتاح العام، والذي يتم طلب شهادة هوية له من قبل العميل.

2. التحقق مما إذا كانت قاعدة بيانات طلب التسجيل تحتوي على طلب بنفس شهادة كلمة مرور التحدي التي يرسلها العميل.

ملاحظة: إذا (1) إرجاع true أو إرجاع كال (1) و(2) صحيح معا، فسيكون بإمكان خادم CA رفض الطلب على أساس طلب الهوية المكررة. ومع ذلك، في مثل هذه الحالة، يقوم خادم IOS PKI باستبدال الطلب الأقدم بالطلب الأحدث.

الخطوة 4. خادم PKI:

منح الطلبات يدويا على خادم PKI:

عرض الطلب:

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

للموافقة على طلب محدد أو على جميع الطلبات:

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

الخطوة 5. عميل PKI:

في هذه الأثناء، يبدأ عميل PKI عملية ضبط مؤقت للاستطلاع. هنا، يقوم IOS بتنفيذ GetCertInitial على فترات منتظمة حتى يتم استلام شهادة SCEP CertRep = "منح" مع الشهادة الممنوحة من قبل العميل.

وبمجرد إستلام الشهادة الممنوحة، يقوم IOS بتثبيتها تلقائيا.

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).