IOS PKI Deployment Guide: Certificate Rollover - Configuration and Operation Overview

Contents

Introduction

This document describes the certificate rollover on Cisco IOS Public Key Infrastructure (PKI) Servers and Clients in detail.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these hardware and software versions:

Hardware

- ISR-G1 [8xx, 18xx, 28xx, 38xx]
- ISR-G2 [19xx, 29xx, 39xx]
- ISR-4K [43xx, 44xx]
- ASR1k
- CSR1k

Software

- IOS
  - For ISR-G1 – Latest 15.1(4)M*
  - For ISR-G2 – Latest 15.4(3)M
- IOS-XE
  - XE 3.15 or 15.5(2)S

  Note: General software maintenance for ISR devices is no longer active, any future bug-fixes or feature-enhancements would require a hardware upgrade to ISR-2 or ISR-4xxx series Routers.

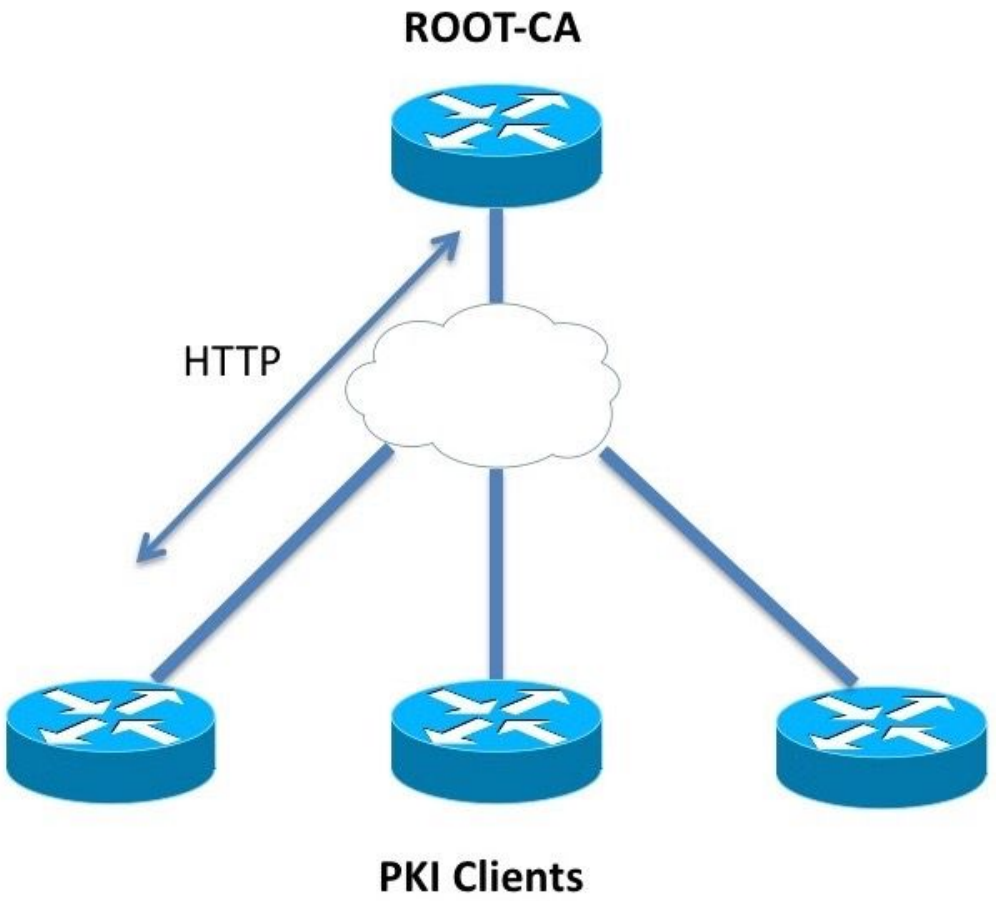
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

Certificate rollover also known as renewal operation ensures that when a certificate expires, a new certificate is ready to take over. From a PKI Server's point of view, this operation involves generating the new server rollover certificate well in-advance to make sure that all the PKI clients have received a new client rollover certificate signed by the new server rollover certificate before

the current certificate expires. From a PKI client's point of view, if the client certificate is expiring but the Certificate Authority (CA) Server's certificate is not, the client requests for a new certificate and replaces the old certificate as soon as the new certificate is received, and if the client certificate is expiring at the same time as the CA server's certificate, the client makes sure to receive the CA server's rollover certificate first, and then it requests for a rollover certificate signed by the new CA server rollover certificate, and both will be activated when old certificates expire.

Setup



PKI and Simple Ceritificate Enrollment Protocol (SCEP) Prerequisite

Authoritative Time Source

In IOS, by default the clock source is considered to be non-authoritative since the hardware clock is not the best source of time. PKI being time sensitive, it is important to configure a valid source of time using NTP. In a PKI deployment, it is recommended to have all the clients and the Server synchronize their clock to a single NTP server, through multiple NTP servers if required. More on this is explained in [IOS PKI Deployment Guide: Initial Design and Deployment]

IOS does not initialize PKI timers without an authoritative clock. Although NTP is highly recommended, as a temporary measure, the administrator can mark the hardware clock as authoritative using:

```
Router(config)# clock calendar-valid
```

HTTP Communication

A requirement for an active IOS PKI Server is HTTP server, which can be enabled using this config-level command:

```
ip http server <1024-65535>
```

This command enables HTTP server on port 80 by default, which can be changed as shown above.

PKI clients should be able to communicate with the PKI server over HTTP to the configured port.

PKI Configuration

Server - Rollover

PKI Server automatic rollover configuration looks like:

```
crypto pki server ROOTCA
 database level complete
 database archive pkcs12 password 7 01100F175804575D72
 issuer-name CN=RootCA,OU=TAC,O=Cisco
 grant auto
 lifetime certificate 365
 lifetime ca-certificate 730
 database url ftp://10.1.1.1/DB/ROOTCA/
 auto-rollover 90
```

The auto-rollover parameter is defined in days. At a more granular level, the command looks like:

```
auto-rollover <days> <hours> <minutes>
```

An auto-rollover value of 90 indicates that the IOS creates a rollover server certificate 90 days before the expiry of the current Server certificate, and the validity of this new rollover certificate starts at the same time as the expiry time of the current active certificate.

Auto-rollover should be configured with such a value that makes sure that the rollover CA certificate is generated on the PKI server well in advance before any PKI client in the network performs GetNextCACert operation as described in the **SHADOW operation overview** section below.

Client - Renewal

PKI Client automatic certificate renewal configuration looks like:

```
crypto pki trustpoint Root-CA
  enrollment url http://172.16.1.1:80
  serial-number
  ip-address none
  password 0 Rev0cati0n$Passw0rd
  subject-name CN=spoke-1.cisco.com,OU=CVO
  revocation-check crl
  rsakeypair spoke-1-RSA
   auto-enroll 80
```

Here, **auto-enroll <percentage> [regenerate]** command states that IOS should perform certificate renewal at exactly 80% of the current certificate's lifetime.

The keyword **regenerate** states that IOS should regenerate the RSA key-pair known as shadow key-pair during every certificate renewal operation.

Care should be taken while configuring auto-enroll percentage. On any given PKI client in the deployment, if a condition arises where the identity certificate expires at the same time as the issuing CA certificate, then the auto-enroll value should always trigger the [shadow] renewal operation after the CA has created the rollover certificate. *Refer to* **PKI Timer dependencies** section under the Deployment examples.

PKI Renewal/Rollover Prerequisites

This document addresses certificate rollover and renewal operations in detail, and hence these events are considered to be completed successfully:

- PKI server initialization with a valid CA certificate.
- PKI clients have been enrolled successfully with the PKI server. i.e. Each PKI client has the CA certificate and an identity certificate aka router certificate.

Enrolling a client involves these events. Without getting too much into detail:

- Trustpoint authentication
- Trustpoint Enrollment

In IOS, a trustpoint is a container for certificates. Any given trustpoint can contain one active Identity certificate and/or one active CA certificate. A trustpoint is considered authenticated if it contains an active CA ceryificate. And it is considered enrolled if it contains an identity certificate. A trustpoint must be authenticated before an enrollment. PKI Server and client configuration, along with trustpoint authentication and enrollment are covered in detail in IOS PKI Deployment Guide: Initial Design and Deployment

Following the CA certificate retrieval/installation, the PKI client retrieves the PKI server capabilities before performing an enrollment. CA capabilities retrieval is explained in this section.

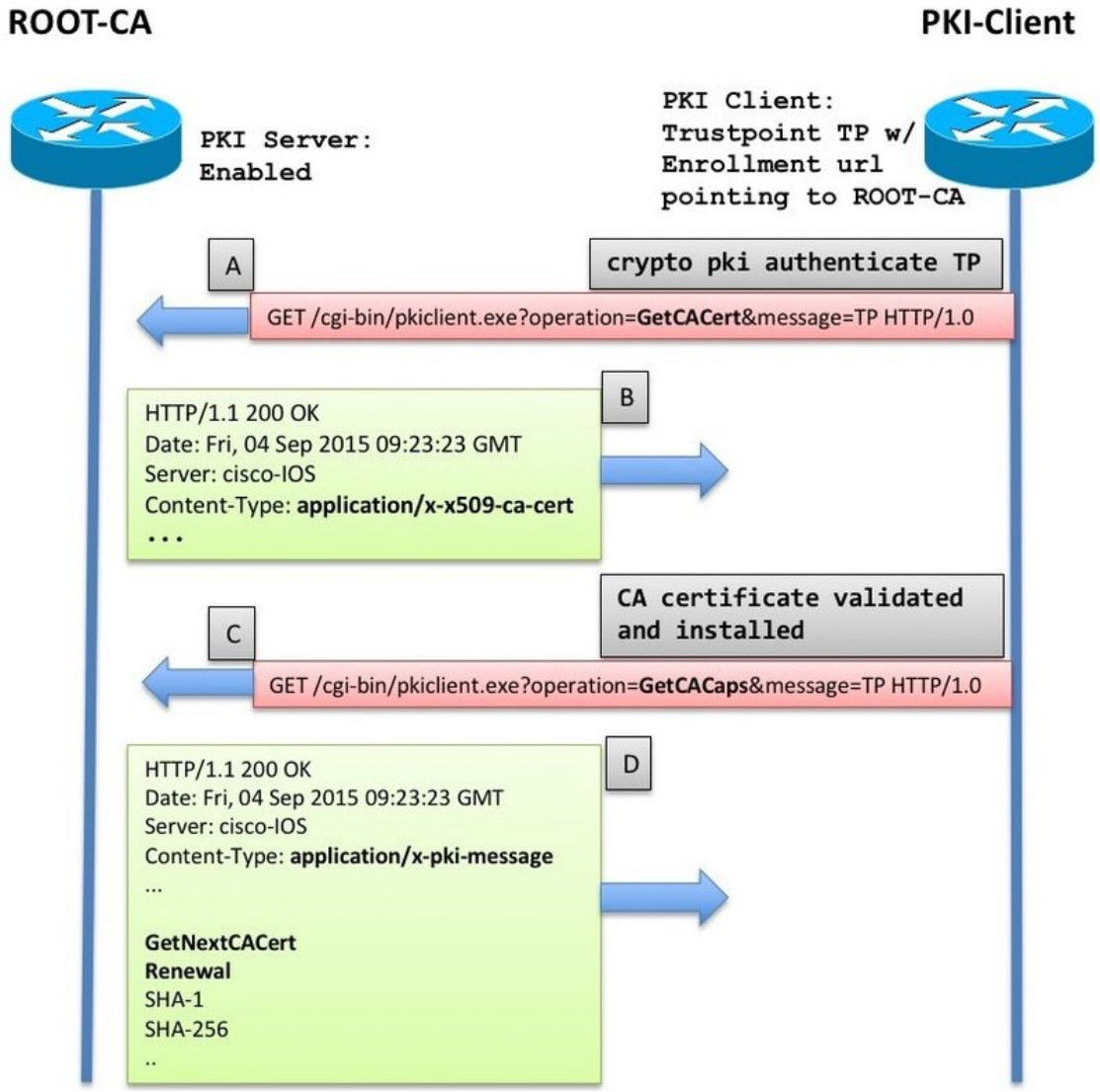
CA Capabilities

In IOS, when a PKI client authenticates a CA, in other words, when an administrator creates a trustpoint on an IOS router, and executes the command **crypto pki authenticate <trustpoint-name>**, these events take place on the router:

- IOS sends a SCEP request containing GetCACert operation type.
- The expected response here is an HTTP message with a content-type of **application/x-x509-**

ca-cert in case of a CA deployment, or **application/x-x509-ca-ra-cert** in case of an RA and a CA deployment. And the HTTP body contains the CA certificate. [and an RA certificate in the latter case].

- Following the CA/RA certificate retrieval and installation, client initiates an automatic SCEP request containing GetCACaps operation.
- Expected response here is an HTTP message with a content-type of **application/x-pki-message**, which could also be **text/plain** and the HTTP body contains a series of capabilities supported by the CA, separated by a line-feed character. A typical IOS PKI Server response is as shown in the diagram below.



The response is interpreted as this by the IOS PKI Client:

```
CA_CAP_GET_NEXT_CA_CERT
CA_CAP_RENEWAL
CA_CAP_SHA_1
CA_CAP_SHA_256
```

Of these Capabilities, this document focuses on these two.

GetNextCACert

When this capability is returned by the CA, IOS understands that the CA supports CA-Certificate

Rollover. With this capability returned, if **auto-enroll** command is not configured under the trustpoint, IOS initializes a SHADOW timer set to 90% of the CA certificate's validity period.

When the SHADOW timer expires, IOS performs GetNextCACert SCEP operation to fetch the Rollover CA certificate.

> **Note**: If **auto-enroll** command has been configured under the trustpoint along with an **enrollment url**, a RENEW timer is initialized even before authenticating the trustpoint, and it constantly tries to enroll with the CA located at the **enrollment url**, although no actual enrolment message [CSR] is sent until the trustpoint is authenticated.

> **Note**: GetNextCACert is sent as a capability by the IOS PKI server even if **auto-rollover** is not configured on the serv

Renewal

With this capability, the PKI server informs the PKI client that it can use an active ID certificate to sign a certificate signing request to renew the existing certificate.

More on this in the **PKI Client Auto-Renewal** section.

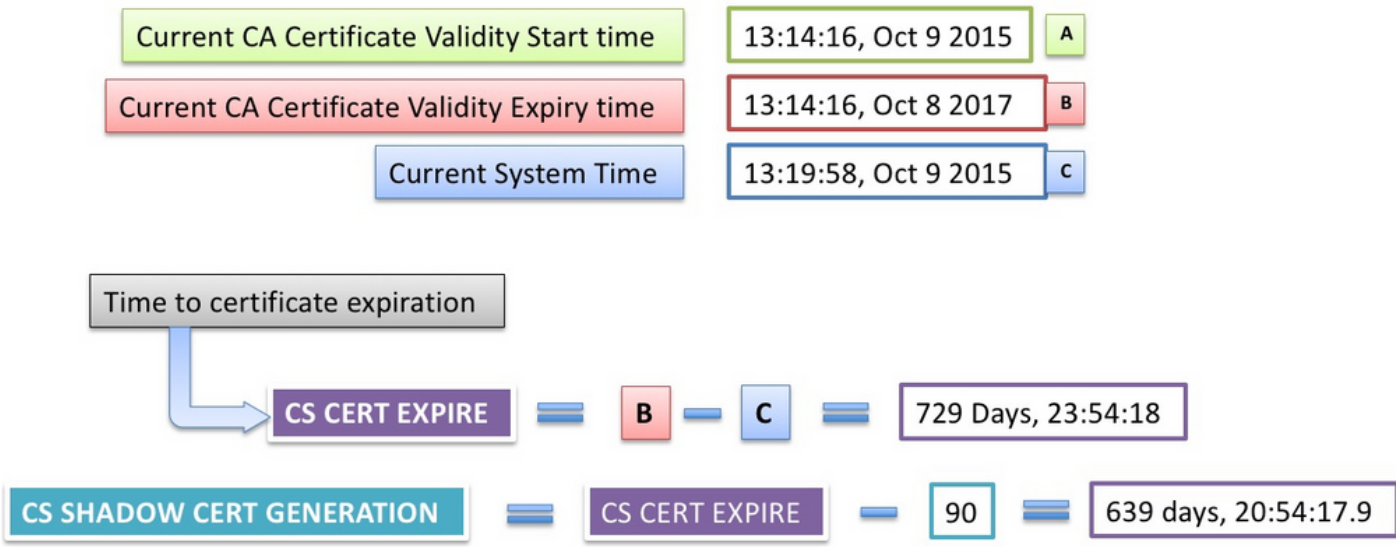
PKI Server Auto-Rollover

With the above configuration on the CA Server, you see:

```
Root-CA#show crypto pki certificates
CA Certificate
 Status: Available
 Certificate Serial Number (hex): 01
 Certificate Usage: Signature
 Issuer:
   cn=RootCA
   ou=TAC
   o=Cisco
 Subject:
   cn=RootCA
   ou=TAC
   o=Cisco
 Validity Date:
   start date: 13:14:16 CET Oct 9 2015
   end   date: 13:14:16 CET Oct 8 2017
 Associated Trustpoints: ROOTCA

Root-CA#terminal exec prompt timestamp

Root-CA#show crypto pki timers
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 13:19:58.946 CET Fri Oct 9 2015
PKI Timers
|        7:49.003
 |        7:49.003  SESSION CLEANUP
 |  3d 7:05:24.003  TRUSTPOOL
CS Timers
```

```
|     5:54:17.977
 |     5:54:17.977  CS CRL UPDATE
 |639d23:54:17.977  CS SHADOW CERT GENERATION
 |729d23:54:17.971  CS CERT EXPIRE
```

Notice this:



Rollover Operation

When the **CS SHADOW CERT GENERATION** timer expires:

- IOS generates a rollover key-pair first – currently it has the same name as the active key-pair with a # hash appended to it.

```
Jul 10 13:14:16.510: CRYPTO_CS: shadow generation timer fired.
Jul 10 13:14:16.510: CRYPTO_CS: key 'ROOTCA#' does not exist; generated automatically

Root-CA# show crypto key mypubkey rsa
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 13:19:19.652 CET Mon Jul 10 2017

% Key pair was generated at: 13:14:16 CET Oct 9 2015
Key name: ROOTCA
Key type: RSA KEYS
 Storage Device: private-config
 Usage: General Purpose Key
 Key is not exportable.
 Key Data&colon;
 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00B07127
 360CF006 13B259CE 7BB8158D E6BC8AA4 8A763F73 50CE64B0 71AC5D93 ED59C936
 F751D810 70CEA8C8 B0023B4B 0FB9A538 A1C118D3 5530D46D C4B4DC14 3BD1D231
 48B0C053 A781D0C7 86DEE9DE CCA58C18 B5804B29 911D1D57 76B3EC3F 42D38C3A
 1E0F8DD9 1DE228B9 95AC3C10 87C132FC 75956338 258727F6 1A1F0818 83020301 0001
```

```
% Key pair was generated at: 13:14:18 CET Jul 10 2017
Key name: ROOTCA#
Key type: RSA KEYS
 Storage Device: not specified
 Usage: General Purpose Key
 Key is not exportable.
 Key Data&colon;
 30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00BF2A52
 687F112B C9263541 BB402939 9C66D270 8D3EACED 4F63AA50 9FB340E8 38C8AC38
 1818EA43 93C17CA1 C4917F43 C9199C9E F9F9C059 FDE11DA9 C7991826 43736FCE
 A80D0CEE 2378F23B 6AC5FC3B 4A7A0120 D391BE8F A9AFD212 E05A2864 6610233C
 E0E58D93 23AA0ED2 A5B1C140 122E6E3D 98A7D974 E2363902 70A89CE3 BF020301 0001
```

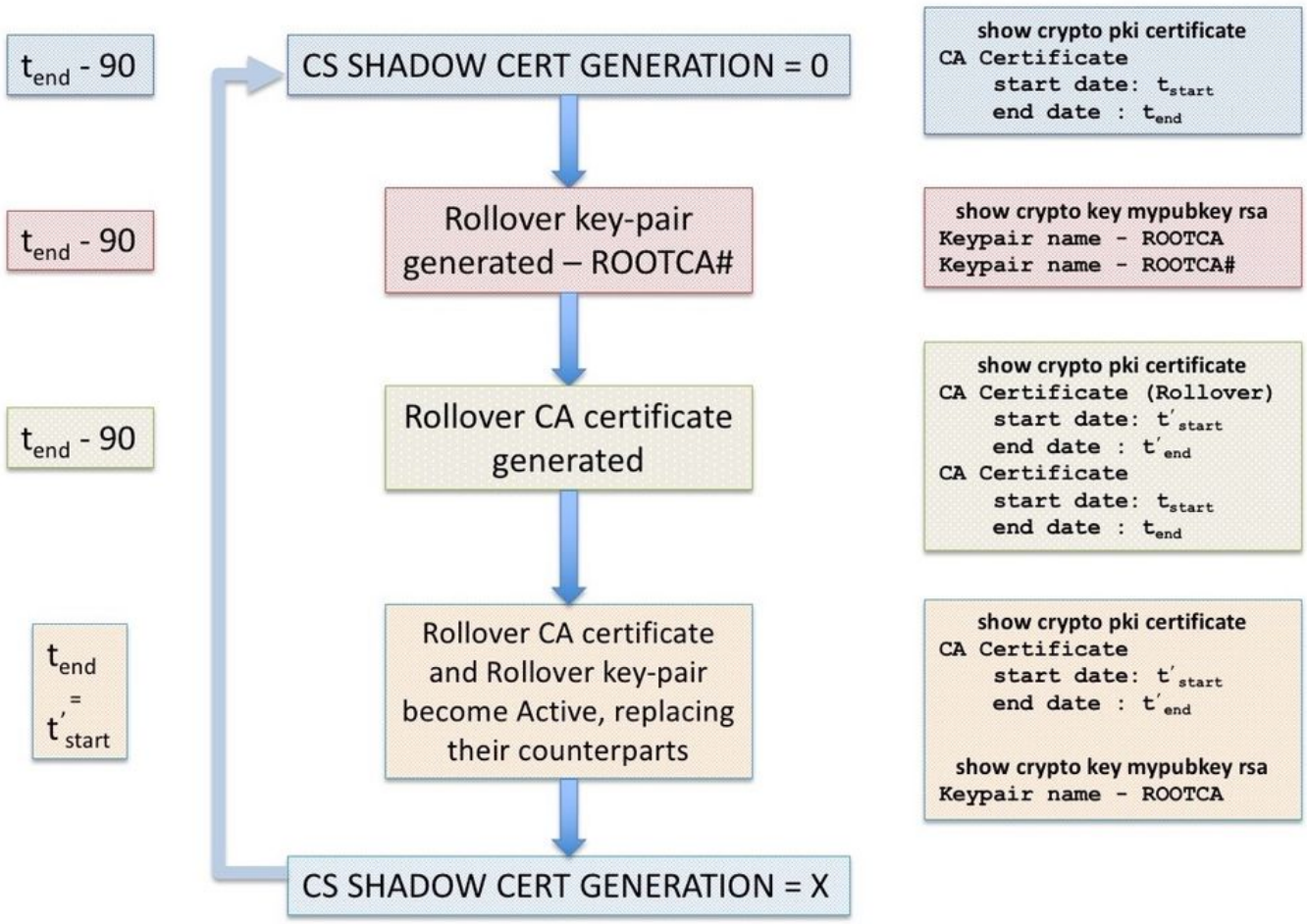
- IOS then generates the rollover CA certificate, where the validity start date is the same as the validity end date of the current active CA certificate.

```
Jul 10 13:14:18.326: CRYPTO_CS: shadow CA successfully created.
Jul 10 13:14:18.326: CRYPTO_CS: exporting shadow CA key and cert
Jul 10 13:14:18.327: CRYPTO_CS: file opened: ftp://10.1.1.1/DB/ROOTCA/ROOTCA_00001.p12
```

```
Root-CA# show crypto pki certificates
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 13:14:46.820 CET Mon Jul 10 2017

CA Certificate (Rollover)
 Status: Available
 Certificate Serial Number (hex): 03
 Certificate Usage: Signature
 Issuer:
   cn=RootCA
   ou=TAC
   o=Cisco
 Subject:
   Name: RootCA
   cn=RootCA
   ou=TAC
   o=Cisco
 Validity Date:
   start date: 13:14:16 CET Oct 8 2017
   end   date: 13:14:16 CET Oct 8 2019
 Associated Trustpoints: ROOTCA

CA Certificate
 Status: Available
 Certificate Serial Number (hex): 01
 Certificate Usage: Signature
 Issuer:
   cn=RootCA
   ou=TAC
   o=Cisco
 Subject:
   cn=RootCA
   ou=TAC
   o=Cisco
 Validity Date:
   start date: 13:14:16 CET Oct 9 2015
```

```
    end   date: 13:14:16 CET Oct 8 2017
 Associated Trustpoints: ROOTCA
 Storage: nvram:RootCA#1CA.cer

Root-CA# show crypto pki server
Certificate Server ROOTCA:
Status: enabled
State: enabled
Server's configuration is locked (enter "shut" to unlock it)
Issuer name: CN=RootCA,OU=TAC,O=Cisco
CA cert fingerprint: CC748544 A0AB7832 935D8CD0 214A152E
Granting mode is: manual
Last certificate issued serial number (hex): 6
CA certificate expiration timer: 13:14:16 CET Oct 8 2017
CRL NextUpdate timer: 19:11:54 CET Jul 10 2017
Current primary storage dir: unix:/iosca-root/
Database Level: Complete - all issued certs written as <serialnum>.cer
 Rollover status: available for rollover
Rollover CA certificate fingerprint: 031904DC F4FAD1FD 8A866373 C63CE20F
 Rollover CA certificate expiration time: 13:14:16 CET Oct 8 2019
Auto-Rollover configured, overlap period 90 days

Root-CA# show run | section chain ROOTCA
crypto pki certificate chain ROOTCA
 certificate ca rollover 03
 30820237 308201A0 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
 2F310E30 0C060355 040A1305 43697363 6F310C30 0A060355 040B1303 54414331
 0F300D06 03550403 1306526F 6F744341 301E170D 31373130 30383132 31343136
 5A170D31 39313030 38313231 3431365A 302F310E 300C0603 55040A13 05436973
 636F310C 300A0603 55040B13 03544143 310F300D 06035504 03130652 6F6F7443
 4130819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281 8100BF2A
 52687F11 2BC92635 41BB4029 399C66D2 708D3EAC ED4F63AA 509FB340 E838C8AC
 381818EA 4393C17C A1C4917F 43C9199C 9EF9F9C0 59FDE11D A9C79918 2643736F
 CEA80D0C EE2378F2 3B6AC5FC 3B4A7A01 20D391BE 8FA9AFD2 12E05A28 64661023
 3CE0E58D 9323AA0E D2A5B1C1 40122E6E 3D98A7D9 74E23639 0270A89C E3BF0203
 010001A3 63306130 0F060355 1D130101 FF040530 030101FF 300E0603 551D0F01
 01FF0404 03020186 301F0603 551D2304 18301680 1419FCA4 DDE84233 F79C066F
 93CCF6B3 E14F8355 31301D06 03551D0E 04160414 19FCA4DD E84233F7 9C066F93
 CCF6B3E1 4F835531 300D0609 2A864886 F70D0101 04050003 81810065 AC780BB4
 2398D765 BE4C4C0A 0D0F16C0 82530D85 99933BDC 8388C46D 926145D8 B0BA275A
 93AAB497 FC876F6A E951C138 F5D652AE C0C25E2A FDD80BAA C6BD5A78 E439158F
 5544F30F 33C59E22 1994A8D3 AADC1287 BD15A104 55CB5DC3 49A9401A 8DB3940A
 5054EA21 99CCE4F3 40B471FE DEB4BB38 AC3ACD48 4CDDCBC9 9829D3
     quit
certificate ca 01
 30820237 308201A0 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
 2F310E30 0C060355 040A1305 43697363 6F310C30 0A060355 040B1303 54414331
 0F300D06 03550403 1306526F 6F744341 301E170D 31353130 30393132 31343136
 5A170D31 37313030 38313231 3431365A 302F310E 300C0603 55040A13 05436973
 636F310C 300A0603 55040B13 03544143 310F300D 06035504 03130652 6F6F7443
 4130819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281 8100B071
 27360CF0 0613B259 CE7BB815 8DE6BC8A A48A763F 7350CE64 B071AC5D 93ED59C9
 36F751D8 1070CEA8 C8B0023B 4B0FB9A5 38A1C118 D35530D4 6DC4B4DC 143BD1D2
 3148B0C0 53A781D0 C786DEE9 DECCA58C 18B5804B 29911D1D 5776B3EC 3F42D38C
 3A1E0F8D D91DE228 B995AC3C 1087C132 FC759563 38258727 F61A1F08 18830203
 010001A3 63306130 0F060355 1D130101 FF040530 030101FF 300E0603 551D0F01
 01FF0404 03020186 301F0603 551D2304 18301680 148D421A BED6DCAD B8CFE4B4
 1B2C7E41 C73428AC 9A301D06 03551D0E 04160414 8D421ABE D6DCADB8 CFE4B41B
 2C7E41C7 3428AC9A 300D0609 2A864886 F70D0101 04050003 8181008C 3495278E
 DA6C14B0 533E746D 8DA743AF 06BE4088 913BF9BC A94576FA BC86EFD1 1DFE6B9F
 0D244144 473C67AD 24414A20 84E9B083 D1720766 0A698C29 115482C6 2FB57E86
 95CDECF2 29662362 866CDC91 730ADBB3 BDBBDC3C EA5301B0 150658E7 AF722BD7
 6B5C2D6A 661A4FED CDA32DE5 D6C2CE7A 544086DC F957A87C 2C07FF
```

```
quit
```



PKI Server Manual-Rollover

IOS PKI Server supports manual rollover of the CA certificate, i.e. an administrator can trigger the generation of a rollover CA certificate in advance without needing to configure **auto-rollover** under the PKI server configuration. It is highly recommended to configure **auto-rollover** whether or not one plans to extend the lifetime of an initially deployed CA server to be on the safer side. **PKI clients can overload the CA without a rollover CA certificate**. *Refer to* Dependency of Client SHADOW operation on PKI Server Rollover.

A manual rollover can be triggered using the configuration level command:

```
crypto pki server <Server-name> rollover
```

And also, a rollover CA certificate can be cancelled to generate a fresh one manually, however something an admin should not do in a production environment, using:

```
crypto pki server <Server-name> rollover cancel
```

This deletes the rollover rsa key-pair and the rollover CA certificate. This is advised against because:

- Once the CA generates the rollover certificate, multiple clients may download the rollover CA certificate as well as a rollover client-certificate signed by the rollover CA certificate.
- At this stage if the rollover is cancelled, the client may have to be re-enrolled.

PKI Client Auto-Renewal

Types of Client Certificate Renewal - RENEW and SHADOW

IOS on the PKI server always makes sure that the expiry time of the ID certificate issued to the client never goes beyond the expiry time of the CA certificate.

On a PKI client, IOS always takes the following timers into consideration before scheduling the renewal operation:

- Expiry time of the Identity certificate being renewed
- Expiry time of the issuer’s (CA) certificate

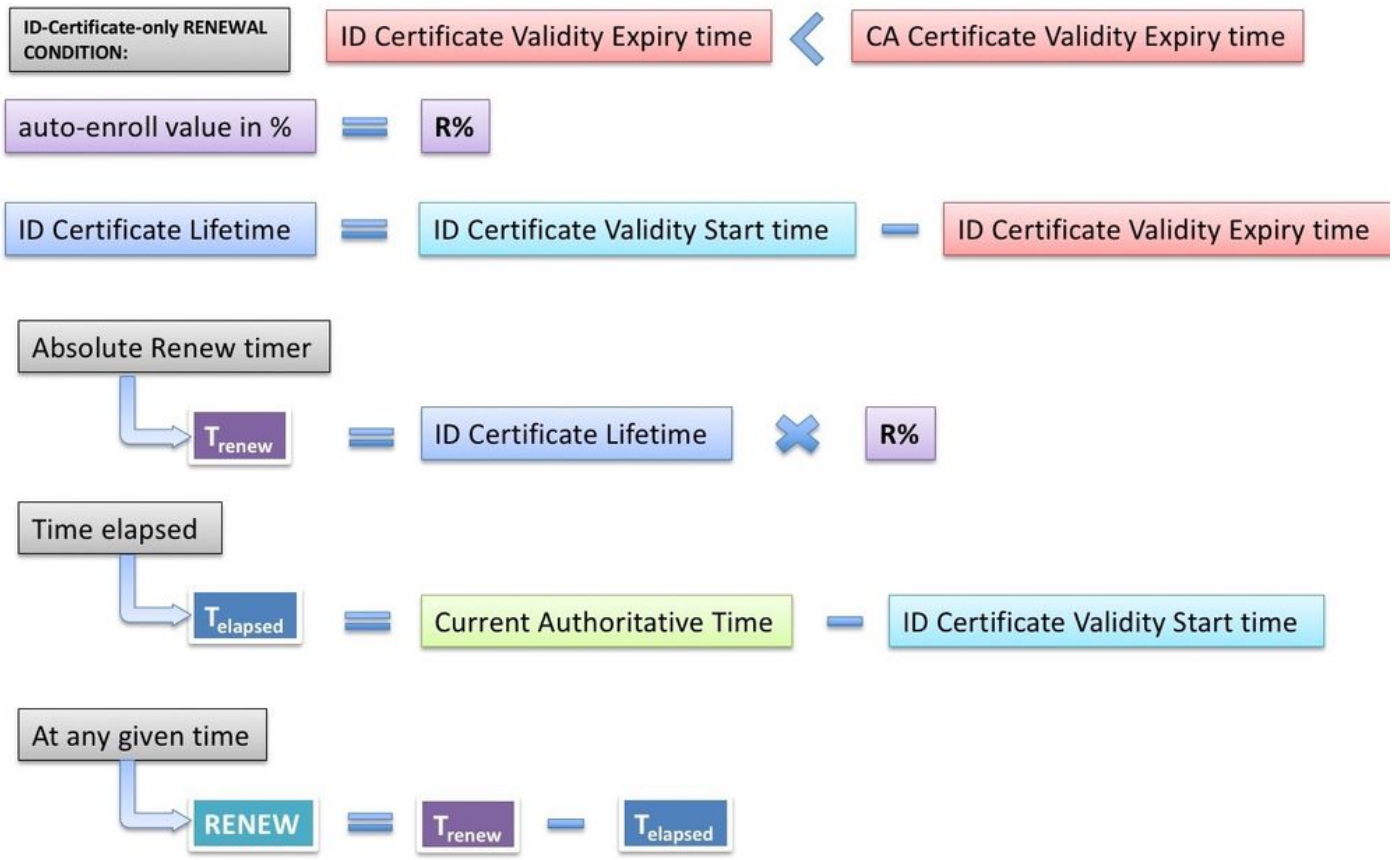
If the Expiry time of the identity certificate is not the same as the expiry time of the CA certificate, IOS performs a simple renewal operation.

If the Expiry time of the identity certificate is the same as the expiry time of the CA certificate, IOS performs a shadow renewal operation.

RENEW - Router Identity Certificate Renewal

As mentioned before, IOS PKI client performs a simple renewal operation if the expiry time of the identity certificate is not the same as the expiry time of the CA certificate, in other words the identity certificate expiring before the issuer’s certificate triggers a simple renewal of the identity certificate.

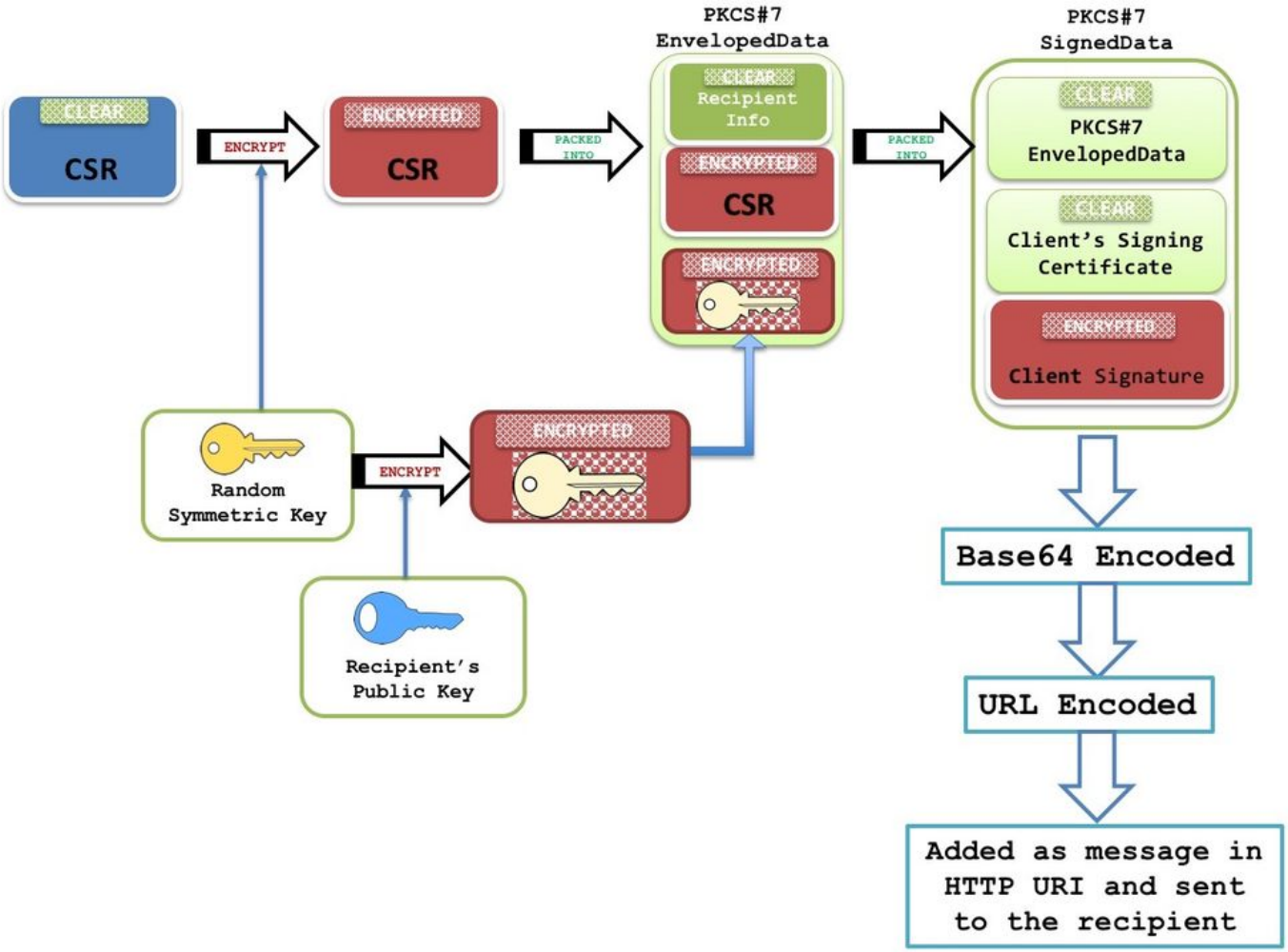
As soon as an identity certificate is installed, IOS calculates the RENEW timer for the specific trust-point as shown below:



Current-Authoritative-Time means that the system clock has to be an authoritative source of time as described here. (link to authoritative Time source section) PKI timers will not be initialized without an authoritative source of time. And as a consequence, renewal operation will not take place.

The following events take place when RENEW timer expires:

- IOS generates a shadow key-pair if **regenerate** is configured [example: auto-enroll 80 regenerate]. Without **regenerate** IOS re-uses the currently active RSA key-pair.
- IOS creates a PKCS-10 formatted certificate request, which is then encrypted into a PKCS-7 envelope. This envelope also contains the RecipientInfo, which is the subject-name and the serial number of the issuing CA. This PKCS7-envelope is in turn packed into a PKCS-7 signed-data. During the initial enrollment, IOS uses a self-signed certificate to sign this message. And during the subsequent enrollments, i.e. re-enrollments, IOS uses the active identity certificate to sign the message. The PKCS7 signed data is also embedded with the signing certificate, i.e. either the self-signed certificate or the identity certificate.



For more information on this packet structure refer to SCEP Overview Document

> **Note**: The key information here is the RecipientInfo which is the subject-name and the serial number of the issuing CA, and the public key of this CA is used to encrypt the symmetric-key. The CSR in the PKCS7 envelope is encrypted using this symmetric-key.

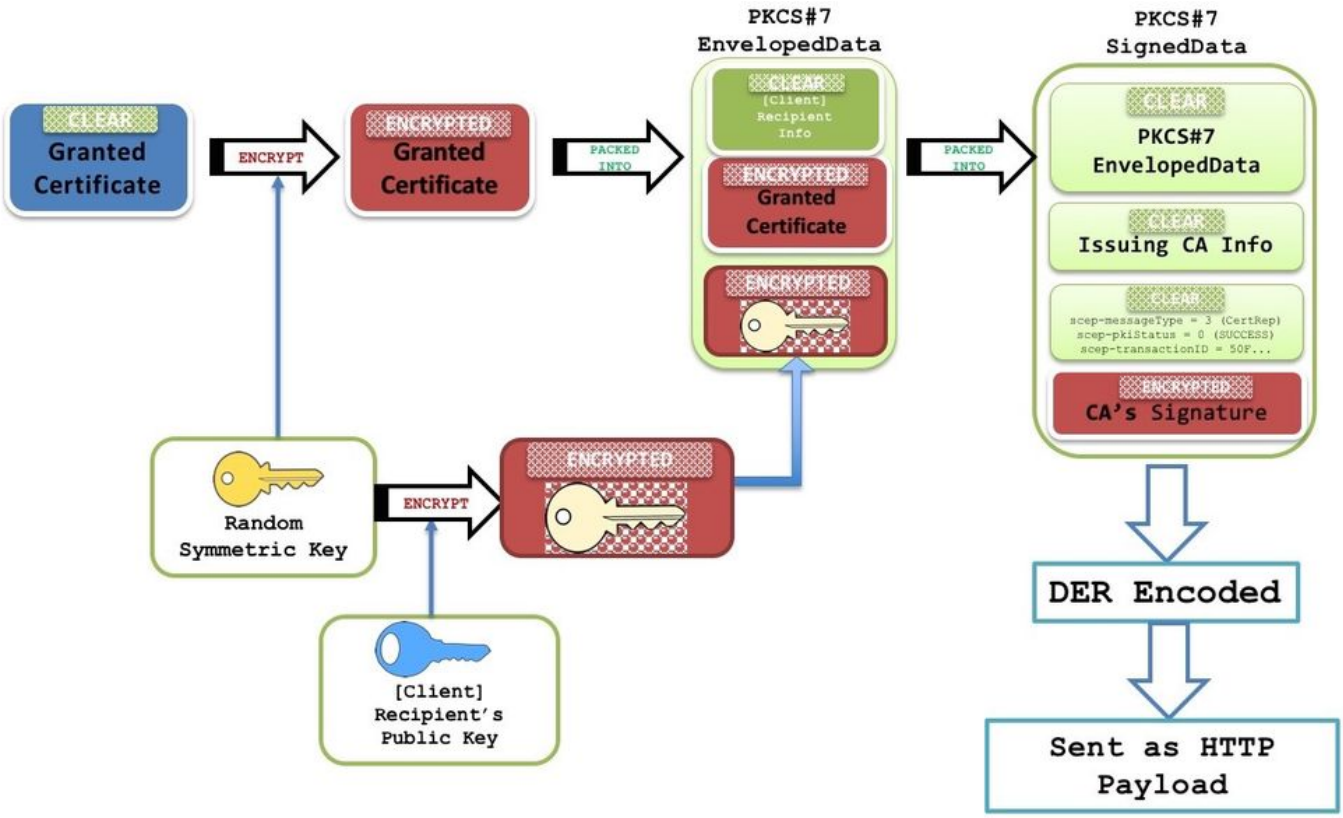
This encrypted symmetric-key is decrypted by the receiving CA using its private key, and this symmetric-key is used to decrypt the PKCS7 envelope revealing the CSR.

- This Certificate Signing Request (CSR) packaged in PKCS7 format is then sent to the CA with a SCEP message-type of PKCSReq and a SCEP operation called PKIOperation.
- If the CA rejects the request, IOS stops the RENEW timer. From this point on, to renew the identity certificate, the administrator must perform a manual renewal (link to **PKI client Manual-Renewal** section)
- If the CA sends a SCEP status as **pending**, IOS on the PKI client starts a POLL timer starting at 60 seconds or 1 minute. Every time a POLL timer expires, IOS sends GetCertInitial SCEP message through a PKIOperation operation.When the first POLL timer expires, if the GetCertInitial message is responded to with a SCEP Pending status, an exponential backoff algorithm sets the first POLL timer retry interval to 1 minute, second POLL timer retry interval to 2 minutes, third POLL timer retry interval to 4 minutes and so on for the next 999 retries by default or till the Issuing CA certificate expires.
  Poll count and first retry period can be configured using:

```
   crypto pki trustpoint <TP>
     enrollment retry count <total retry count>
enrollment retry period <first retry period in minutes>
```

- When the certificate is granted on the PKI Server, the next GetCertInitial SCEP message is responded to with an HTTP message of content type **application/x-pki-message** and a body containing a signed PKCS#7 signed data. This PKCS7 signed data contains the SCEP status as **Granted**, and also a PKCS7 enveloped data. This PKCS enveloped data contains the granted certificate and the RecipientInfo, which is the subject-name and the serial number of the self-signed certificate during the initial enrollment and of the active identity certificate during re-enrollments.

  The PKCS7 enveloped data also contains a symmetric key encrypted with the recipient's public key (for which the new certificate was granted). Receiving router decrypts it using the private key. This clear symmetric key is then used to decrypt the PKCS#7 enveloped data, revealing the new identity certificate.



- At this stage, IOS replaces the existing identity certificate with the new certificate immediately. And if **regenerate** was configured, the shadow key-pair replaces the active key-pair as well.

- Also, the end-date of the new certificate is compared with the end-date of the CA certificate to determine if RENEW timer has to be initialized or a SHADOW timer has to be initialized as explained here <href **Types of Client Certificate Renewal - RENEW and SHADOW>**