Руководство по развертыванию PKI IOS: одновременное нажатие клавиш сертификата - конфигурация и обзор операции

Содержание

Введение

Этот документ описывает одновременное нажатие клавиш сертификата на Серверах Инфраструктуры открытых ключей (PKI) Cisco IOS и Клиентах подробно.

Предварительные условия

Требования

Для этого документа отсутствуют особые требования.

Используемые компоненты

Сведения в документе приведены на основе данных версий аппаратного и программного обеспечения:

Аппаратные средства

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

Программное обеспечение

- IOS
  - Для ISR-G1 – последний 15.1 (4) M*
  - Для ISR-G2 – последний 15.4 (3) M
- XE IOS
  - XE 3.15 или 15.5 (2) S

Примечание: Общее обслуживание программного обеспечения для устройств ISR больше не активно, любые будущие исправления ошибки или усовершенствования функции потребовали бы модернизации оборудования к series маршрутизаторам ISR-4xxx или ISR 2.

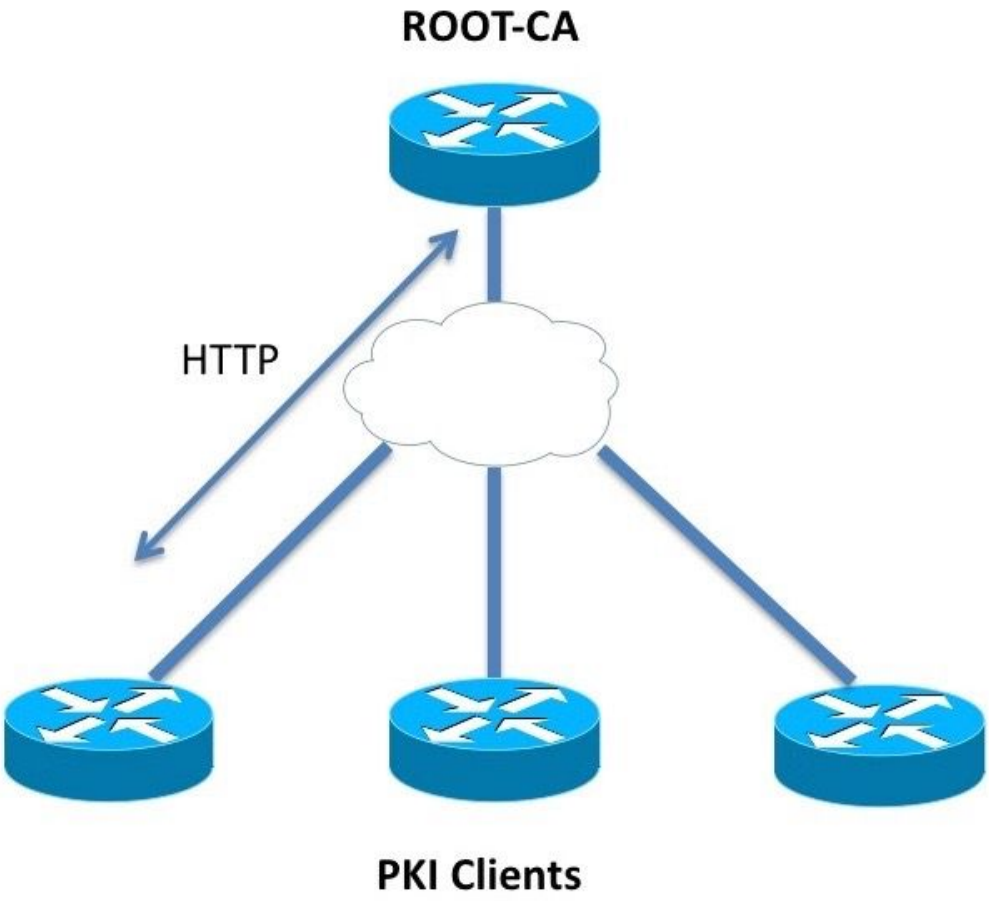
Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Общие сведения

Одновременное нажатие клавиш сертификата, также известное как операция обновления, гарантирует, что, когда сертификат истекает, новый сертификат готов вступить во владение. С точки зрения Сервера pki эта операция включает генерацию нового сертификата одновременного нажатия клавиш сервера заранее, чтобы удостовериться, что

все клиенты PKI получили новый клиентский сертификат одновременного нажатия клавиш, подписанный новым сертификатом одновременного нажатия клавиш сервера, прежде чем истечет текущий сертификат. С точки зрения клиента PKI, если сертификат клиента истекает, но сертификат Сервера Центра сертификации (CA) не является, запросы клиента для нового сертификата и заменяет старый сертификат, как только новый сертификат получен, и если сертификат клиента истекает в то же время, что и сертификат сервера CA, клиент удостоверяется, что получил сертификат одновременного нажатия клавиш сервера CA сначала, и затем это запрашивает на сертификат одновременного нажатия клавиш, подписанный новым сертификатом одновременного нажатия клавиш сервера CA, и оба будут активированы, когда истекут старые сертификаты.

Настройка



PKI и предпосылка Простого протокола регистрации Ceritificate (SCEP)

Надежный источник времени

В IOS по умолчанию источник синхронизации, как полагают, является неавторитетным, так

как аппаратные часы не являются лучшим источником времени. PKI, бывший время чувствительный, важно настроить допустимый источник времени с помощью NTP. В развертываниях PKI рекомендуется иметь всех клиентов, и Сервер синхронизируют их часы с одиночным сервером NTP, через множественные серверы NTP при необходимости. Больше на этом объяснен в Руководстве по развертыванию PKI IOS: Исходное проектное решение и Развертывания

IOS не инициализирует таймеры PKI без авторитетных часов. Несмотря на то, что NTP настоятельно рекомендован, поскольку временное измерение, администратор может отметить аппаратные часы как авторитетное использование:

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

Связь HTTP

Требование для активного Сервера pki IOS является сервером HTTP, который может быть включен с помощью этой команды уровня config:

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

Эта команда включает сервер HTTP на порту 80 по умолчанию, который может быть изменен как показано выше.

Клиенты PKI должны быть в состоянии связаться с Сервером pki по HTTP к настраиваемому порту.

Конфигурация PKI

Сервер- Одновременное нажатие клавиш

Сервер pki автоматическая конфигурация одновременного нажатия клавиш похож:

```
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
```

Параметр auto-rollover определен в днях. На более гранулированном уровне команда похожа:

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

Значение auto-rollover 90 указывает, что IOS создает серверный сертификат одновременного нажатия клавиш за 90 дней до истечения сертификата текущего сервера, и законность этого нового сертификата одновременного нажатия клавиш запускается в то же время, что и время истечения текущего активного сертификата.

Auto-rollover должен быть настроен с таким значением, которое удостоверяется, что сертификат CA одновременного нажатия клавиш генерируется на Сервере pki заранее, прежде чем любой клиент PKI в сети выполнит операцию GetNextCACert, как описано в разделе **обзора операции SHADOW** ниже.

Клиент - обновление

Клиент PKI автоматическая конфигурация обновления сертификата похож:

```
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
```

Здесь, **автоматические регистрации <процент> [восстанавливают]** состояния команды, что IOS должен выполнить обновление сертификата точно в 80% срока действия текущего сертификата.

Ключевое слово **восстанавливает** состояния, что IOS должен восстановить Открытые и секретные ключи криптосистемы RSA, известные как теневая пара согласованных ключей во время каждой операции обновления сертификата.

Меры должны быть приняты при настройке процента автоматической регистрации. На любом данном клиенте PKI в развертываниях, если условие возникает, где сертификат идентификации истекает в то же время, что и сертификат CA запуска, тогда значение автоматической регистрации должно всегда инициировать [теневую] операцию обновления после того, как CA создал сертификат одновременного нажатия клавиш. *См.* **PKI зависимости от Таймера** разделяют под Примерами развертывания.

Предварительные условия Обновления/Одновременного нажатия клавиш PKI

Этот документ обращается к операциям одновременного нажатия клавиш и обновления сертификата подробно, и следовательно эти события, как полагают, завершены успешно:

- Инициализация сервера pki с допустимым сертификатом CA.
- Клиенты PKI были зарегистрированы успешно с Сервером pki. т.е. у Каждого клиента PKI есть сертификат CA и сертификат идентификации иначе сертификат маршрутизатора.

Регистрация клиента включает эти события. Не добираясь слишком много в подробность:

- Аутентификация точки доверия
- Регистрация точки доверия

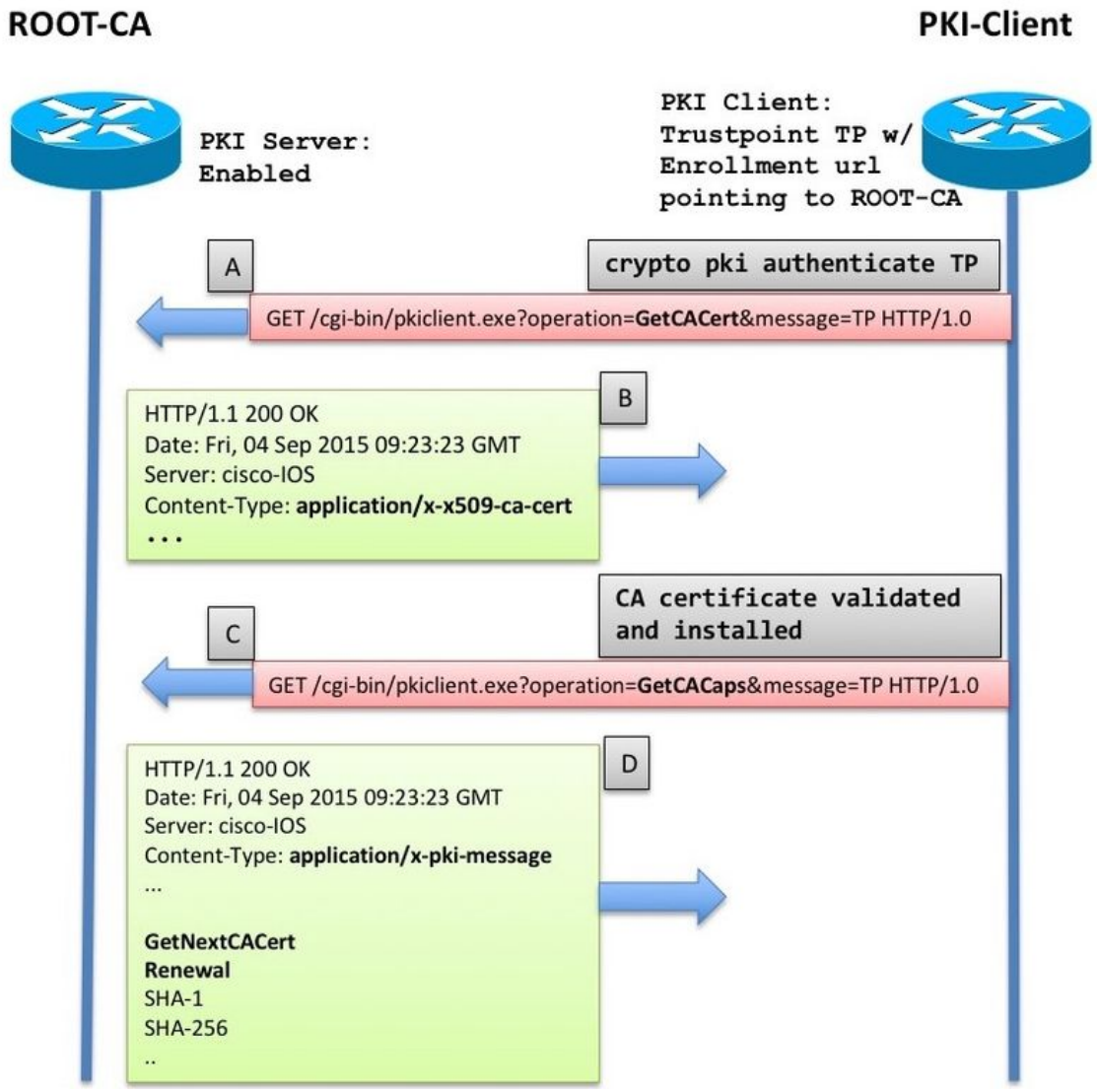
В IOS точка доверия является контейнером для сертификатов. Любая данная точка доверия может содержать один активный Сертификат идентификации и/или один активный сертификат CA. Точку доверия считают аутентифицируемой, если она содержит активный ceryificate CA. И считается зарегистрированным, если это содержит сертификат идентификации. Точка доверия должна аутентифицироваться перед регистрацией. Сервер pki и конфигурация клиента, наряду с аутентификацией точки доверия и регистрацией покрыты подробно в [Руководстве по развертыванию PKI IOS: Исходное проектное решение и Развертывания]

После извлечения/установки сертификата CA клиент PKI получает возможности Сервера pki прежде, чем выполнить регистрацию. CA извлечение возможностей объяснено в этом разделе.

CA возможности

В IOS, когда клиент PKI аутентифицирует CA, другими словами, когда администратор создает точку доверия на маршрутизаторе IOS, и выполняет **команду crypto pki authenticate <name> точки доверия,** эти события имеют место на маршрутизаторе:

- IOS отправляет запрос SCEP, содержащий тип операции GetCACert.
- Ожидаемый ответ здесь является сообщением HTTP с типом содержимого **application/x-x509-ca-cert** в случае развертываний CA, или **application/x-x509-ca-ra-cert** в случае RA и развертываний CA. И Тело HTTP содержит сертификат CA. [и сертификат RA в последнем случае].
- После извлечения сертификата CA/RA и установки, клиент инициирует автоматический запрос SCEP, содержащий операцию GetCACaps.
- Ожидаемый ответ здесь является сообщением HTTP с типом содержимого **application/x-pki-message**, который мог также быть **текстом/плоскостью**, и Тело HTTP содержит серию возможностей, поддерживаемых CA, разделенным символом перевода строки. Типичный ответ Сервера pki IOS находится как показано в приведенном ниже рисунке.



Ответ интерпретируется как это Клиентом PKI IOS:

```
CA_CAP_GET_NEXT_CA_CERT
CA_CAP_RENEWAL
CA_CAP_SHA_1
CA_CAP_SHA_256
```

Из этих Возможностей этот документ фокусируется на этих двух.

GetNextCACert

Когда эта возможность возвращена CA, IOS понимает, что CA поддерживает Одновременное нажатие клавиш Сертификата CA. С этой возвращенной возможностью, если **команда auto-enroll** не настроена под точкой доверия, IOS инициализирует набор таймера SHADOW к 90% периода достоверности сертификата CA.

Когда таймер SHADOW истекает, IOS выполняет операцию GetNextCACert SCEP для выборки сертификата CA Одновременного нажатия клавиш.

> **Примечание:** Если **команда auto-enroll** была настроена под точкой доверия наряду с **enrollment url**, таймер RENEW инициализируется даже прежде, чем аутентифицировать точку доверия, и это постоянно пытается зарегистрироваться с CA, расположенным в **enrollment url**, невзирая на то, что никакое фактическое сообщение приема [CSR] не передается, пока точка доверия не аутентифицируется.

> Даже если **auto-rollover** не настроен на serv, **Примечание:** GetNextCACert передается как возможность Сервером pki IOS

Обновление

С этой возможностью Сервер pki сообщает клиенту PKI, что это может использовать активный сертификат ID для подписания запроса подписи сертификата для возобновления существующего сертификата.

Больше на этом в разделе **Автообновления Клиента PKI**.

Auto-Rollover сервера pki

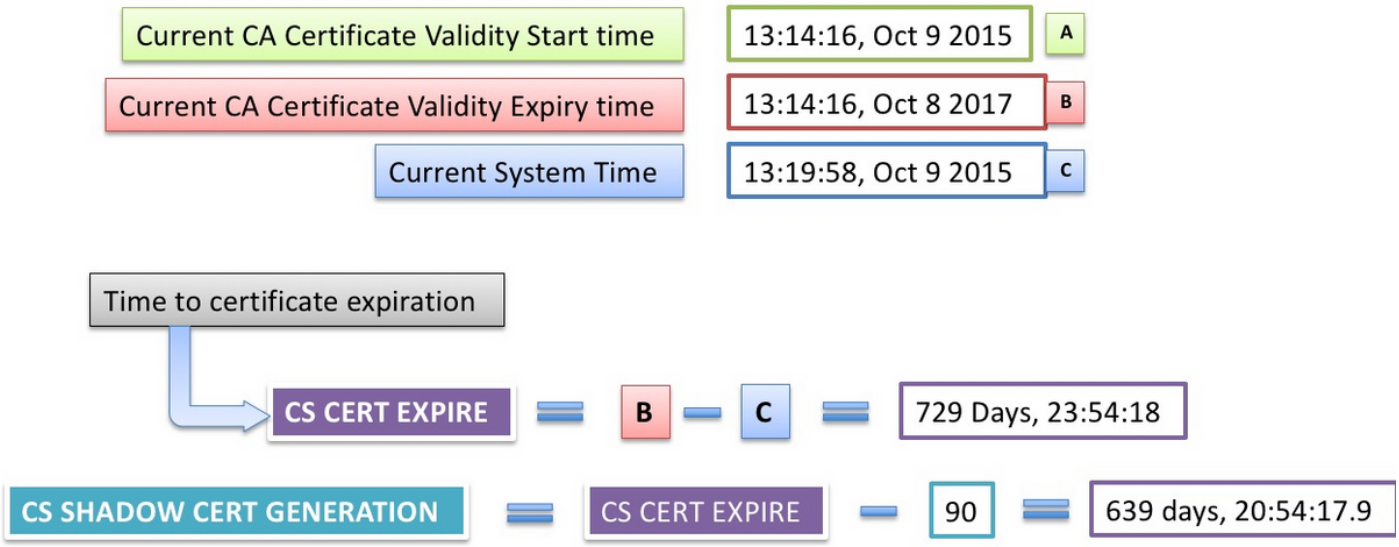
С вышеупомянутой конфигурацией на Сервере CA вы видите:

```
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
```

Заметьте это:



Операция одновременного нажатия клавиш

Когда истекает **CS CERT SHADOW** таймер **ГЕНЕРАЦИИ**:

- IOS генерирует пару согласованных ключей одновременного нажатия клавиш сначала – в настоящее время он имеет то же название как активная пара согласованных ключей с хэшем #, добавленным к нему.

```
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 тогда генерирует сертификат CA одновременного нажатия клавиш, где дата начала законности совпадает с датой завершения законности текущего активного сертификата CA.

```
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
```

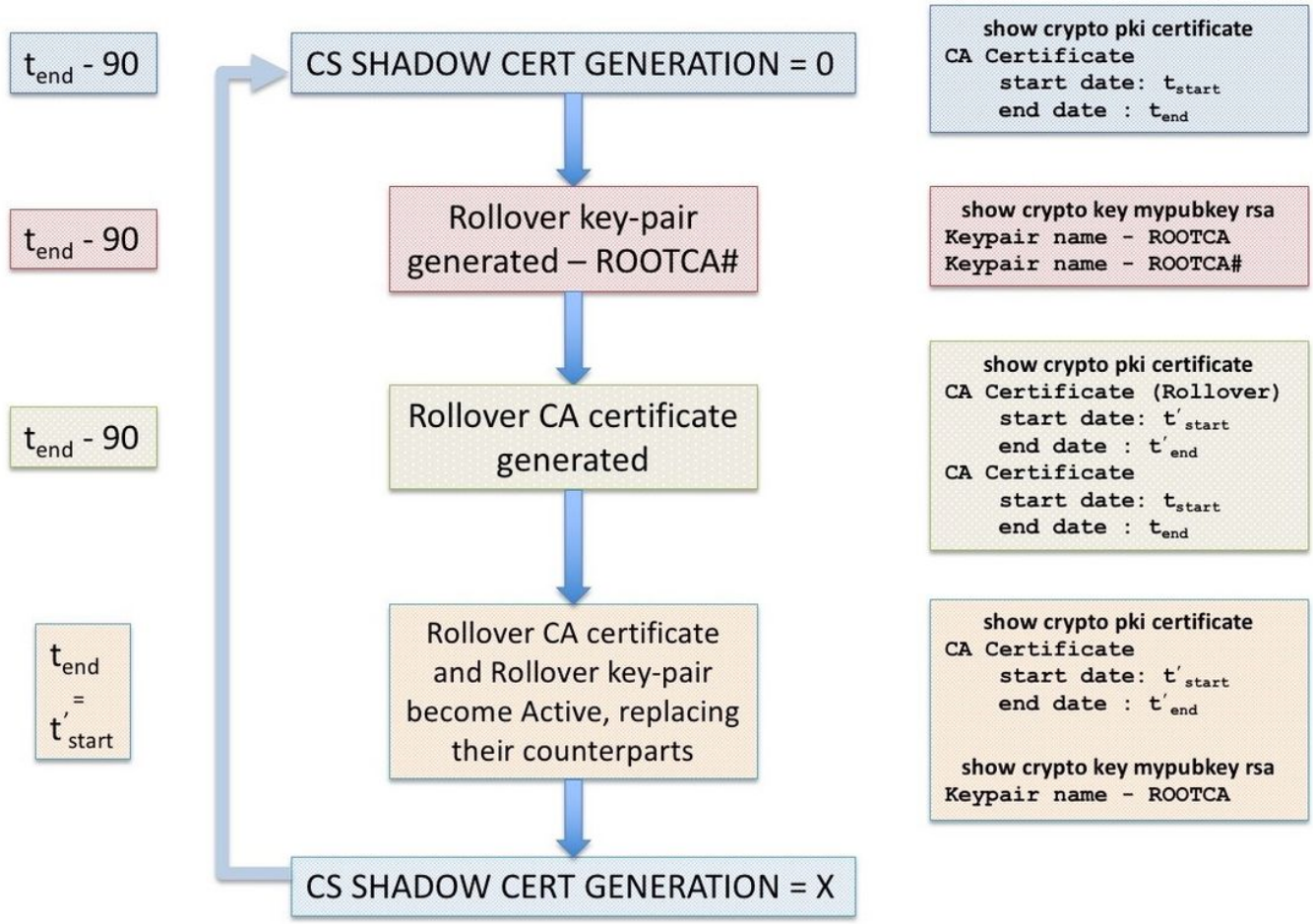
```
Root-CA# show crypto pki certificates
Load for five secs: 0%/0%; one minute: 0%; five minutes: 0%
Time source is NTP, 13:14:46.820 CET Mon Jul 10 2017

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
```



Ручное Одновременное нажатие клавиш сервера pki

Сервер pki IOS поддерживает ручное одновременное нажатие клавиш сертификата CA, т.е. администратор может инициировать генерацию сертификата CA одновременного нажатия клавиш заранее, не будучи должен настроить **auto-rollover** под конфигурацией Сервера pki. Это настоятельно рекомендовано для настройки **auto-rollover**, планирует ли каждый расширить срок действия первоначально развернутого сервера CA, чтобы быть на более безопасной стороне. **Клиенты PKI могут перегрузить CA без сертификата CA одновременного нажатия клавиш**. *См.* Зависимость Клиента операция SHADOW на Одновременном нажатии клавиш Сервера pki.

Ручное одновременное нажатие клавиш может быть инициировано с помощью команды уровня конфигурации:

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

И также, сертификат CA одновременного нажатия клавиш может быть отменен для генерации нового вручную, однако что-то, что admin не должен делать в производственной среде, с помощью:

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

Это удаляет открытые и секретные ключи криптосистемы RSA одновременного нажатия

клавиш и сертификат CA одновременного нажатия клавиш. Это рекомендуется против потому что:

- Как только CA генерирует сертификат одновременного нажатия клавиш, несколька клиентов могут загрузить сертификат ЦС одновременного нажатия клавиш, а также сертификат клиента одновременного нажатия клавиш, подписанный сертификатом CA одновременного нажатия клавиш.
- На данном этапе, если одновременное нажатие клавиш отменено, клиента, вероятно, придется повторно зарегистрировать.

Автообновление клиента PKI

Типы обновления сертификата клиента - RENEW и SHADOW

IOS на Сервере pki всегда удостоверяется, что время истечения сертификата ID, выполненного клиенту никогда, не проходит вне времени истечения сертификата CA.

На клиенте PKI IOS всегда принимает следующие таймеры во внимание прежде, чем планировать операцию обновления:

- Время истечения возобновляемого Сертификата идентификации
- Время истечения отправителя (CA) сертификат

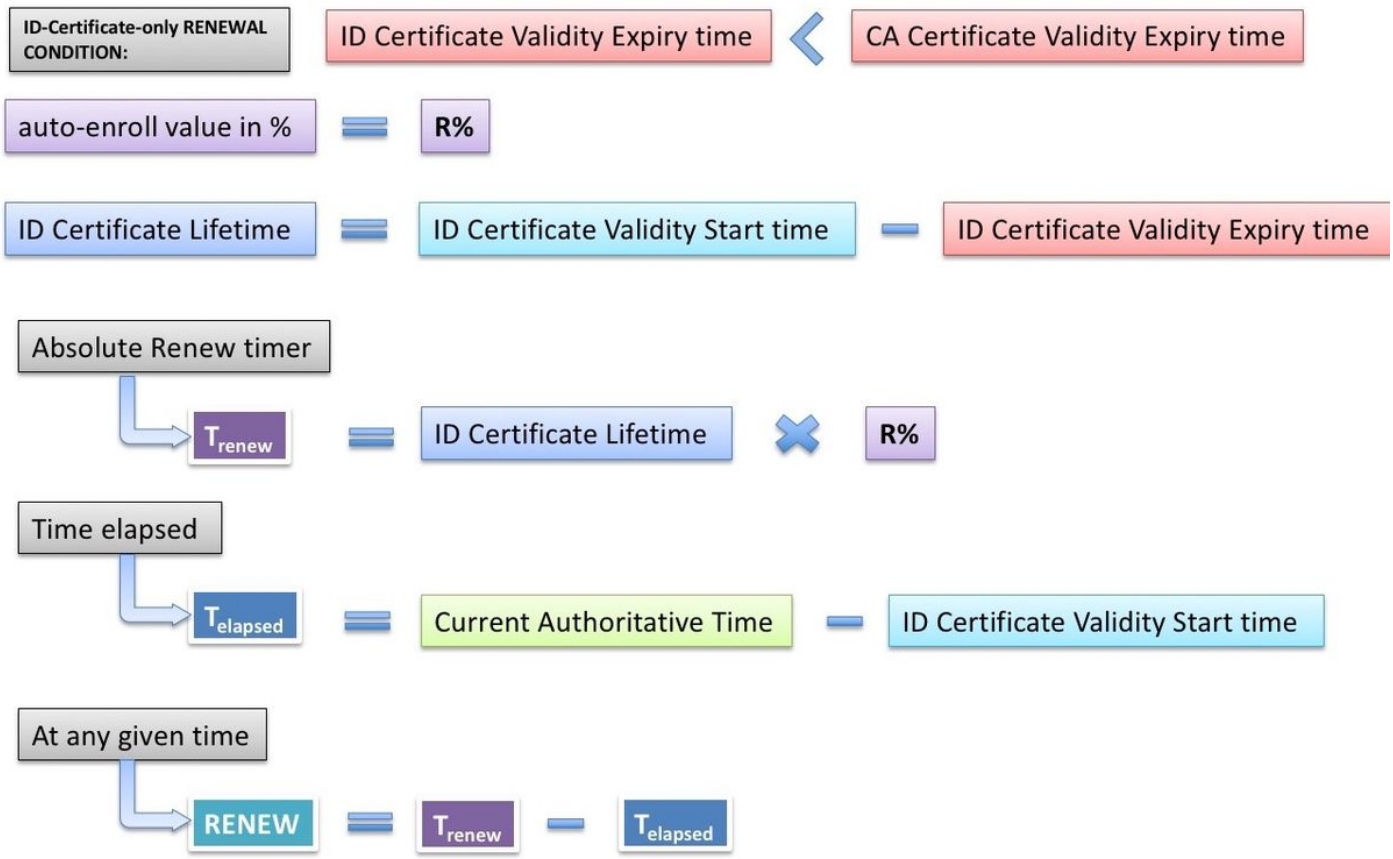
Если время Истечения сертификата идентификации не является тем же как временем истечения сертификата CA, IOS выполняет простую операцию обновления.

Если время Истечения сертификата идентификации совпадает со временем истечения сертификата CA, IOS выполняет теневую операцию обновления.

RENEW - Обновление сертификата идентификации маршрутизатора

Как упомянуто прежде, клиент PKI IOS выполняет простую операцию обновления, если время истечения сертификата идентификации не является тем же как временем истечения сертификата CA, другими словами сертификат идентификации, истекающий, прежде чем сертификат отправителя инициирует простое обновление сертификата идентификации.

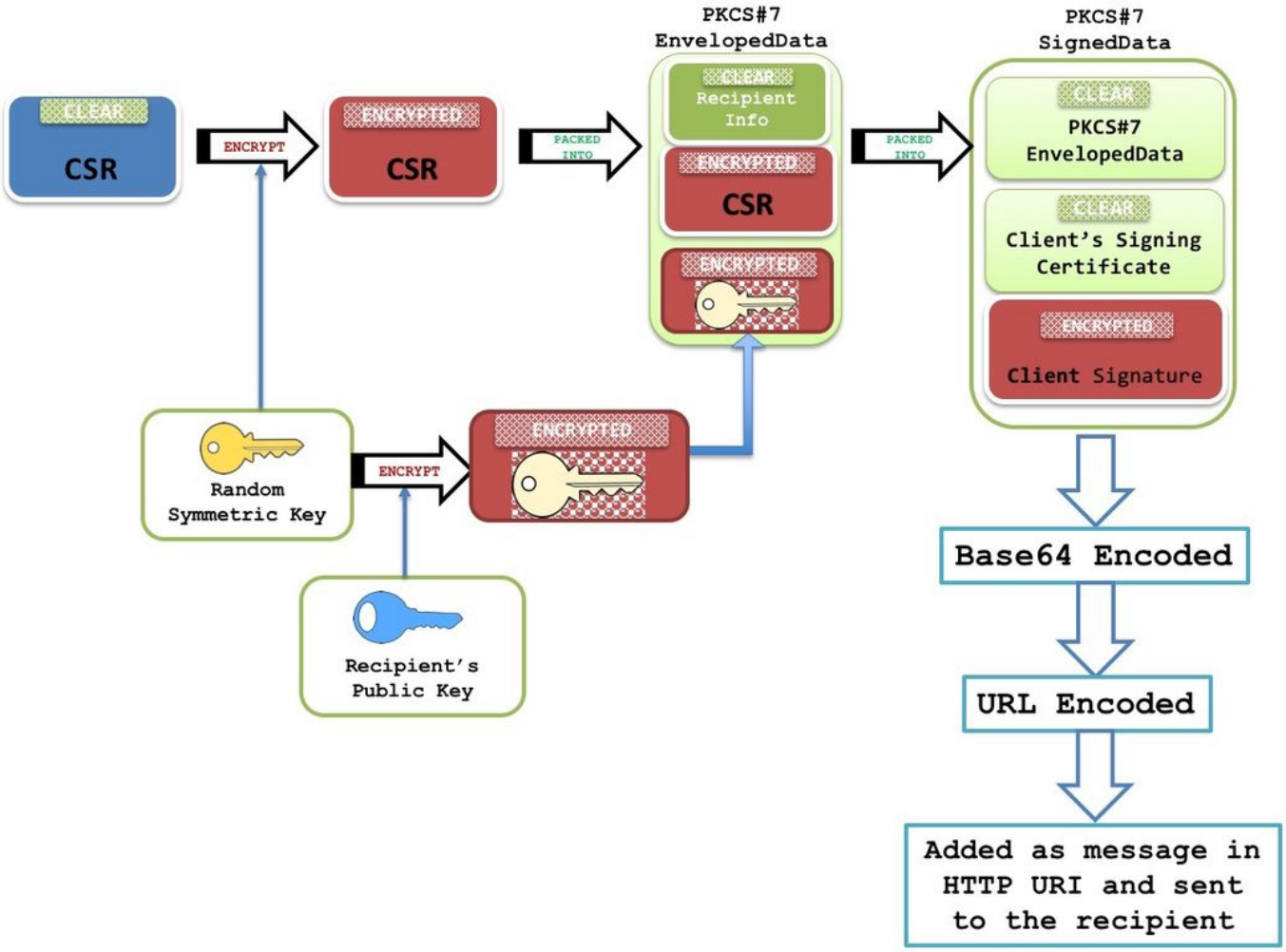
Как только сертификат идентификации установлен, IOS вычисляет таймер RENEW для определенной точки доверия как показано ниже:



Current-Authoritative-Time означает, что системные часы должны быть авторитетным источником времени, как описано здесь. (свяжитесь с разделом надежного источника времени), таймеры PKI не будут инициализироваться без авторитетного источника времени. И как следствие, операция обновления не будет иметь место.

Когда таймер RENEW истекает, следующие события имеют место:

- IOS генерирует теневую пару согласованных ключей, если **восстанавливают**, настроен [пример: автоматические регистрации 80 восстанавливают]. Без **восстанавливают** повторные использования IOS в настоящее время активные Открытые и секретные ключи криптосистемы RSA.
- IOS создает отформатированный запрос сертификата PKCS10, который тогда зашифрован в конверт PKCS-7. Этот конверт также содержит RecipientInfo, который является subject-name и серийным номером CA. запуска, Этот конверт PKCS7 в свою очередь упакован в данные со знаком PKCS-7. Во время начальной регистрации IOS использует подписанный сертификат для подписания этого сообщения. И во время последующих регистраций, т.е. повторного зачисления, IOS использует активный сертификат идентификации для подписания сообщения. PKCS7 подписался, данные также встроены с сертификатом подписания, т.е. или подписанный сертификат или сертификат идентификации.



Для получения дополнительной информации об этой структуре пакета обращаются к [Обзорному документу SCEP]

> **Примечание:** Основной информацией здесь является RecipientInfo, который является subject-name и серийным номером запуска CA, и открытый ключ этого CA используется для шифрования симметричного ключа. CSR в конверте PKCS7 зашифрован с помощью этого симметричного ключа.

Этот зашифрованный симметричный ключ дешифрован получением CA с помощью его секретного ключа, и этот симметричный ключ используется для дешифрования конверта PKCS7, раскрывающего CSR.

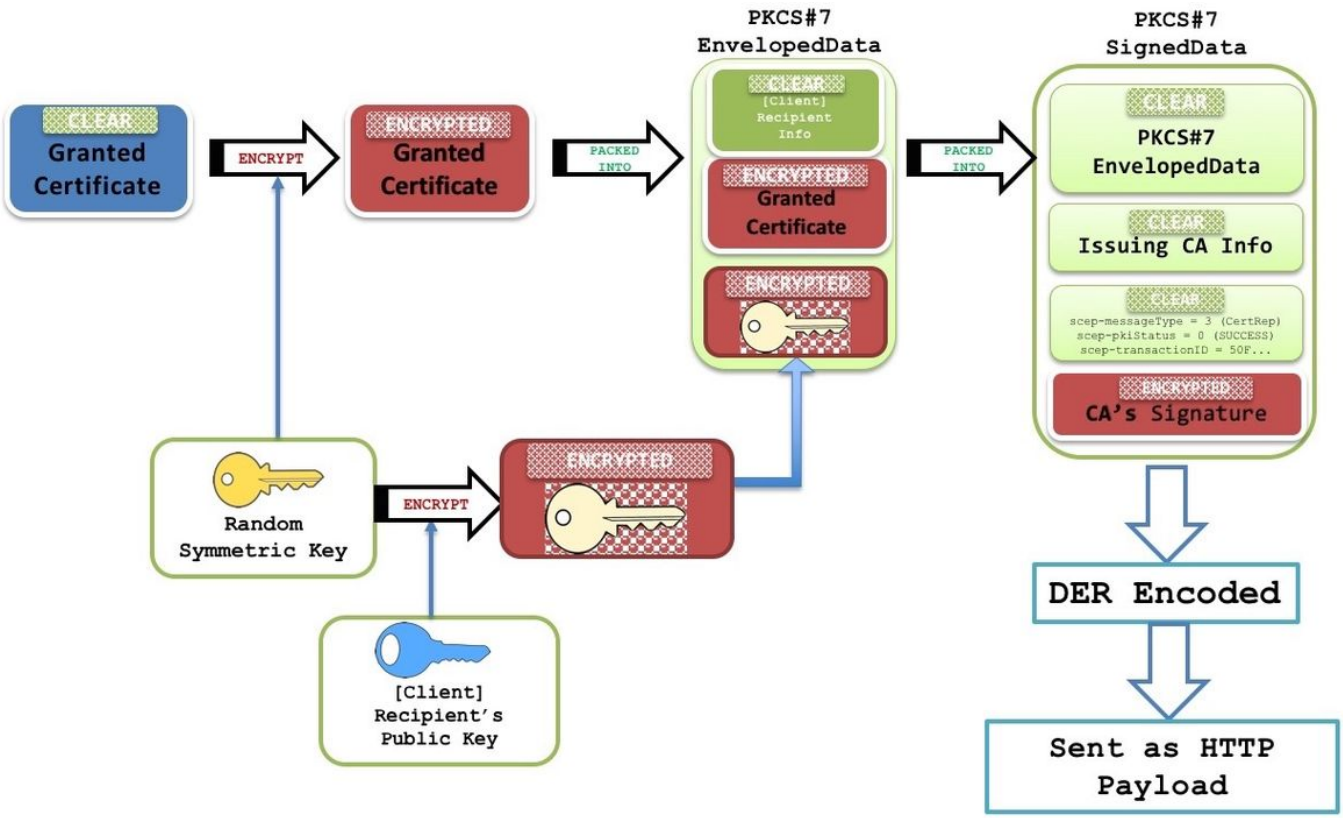
- Этот Запрос подписи сертификата (CSR), упакованный в формате PKCS7, тогда передается CA с message-type SCEP PKCSReq и операции SCEP под названием PKIOperation.
- Если CA отклоняет запрос, IOS останавливает таймер RENEW. С этого момента, для возобновления сертификата идентификации администратор должен выполнить, ручное обновление (свяжитесь с разделом **Ручного Обновления клиента PKI**),
- Если CA передает Статус SCEP как **ожидание**, IOS на клиенте PKI запускает таймер ОПРОСА, запускающийся в 60 секунд или 1 минуту. Каждый раз, когда таймер ОПРОСА истекает, IOS передает сообщение GetCertInitial SCEP посредством операции PKIOperation. Когда первый таймер ОПРОСА истекает, если на сообщение GetCertInitial отвечают с Состоянием ожидания SCEP, алгоритм экспоненциальной задержки

устанавливает первый интервал между попытками таймера ОПРОСА в 1 минуту, второй интервал между попытками таймера ОПРОСА к 2 минутам, третий интервал между попытками таймера ОПРОСА к 4 минутам и так далее для следующих 999 повторных попыток по умолчанию или пока не истекает сертификат CA Запуска.
Количество опроса и сначала повторяет период, может быть настроен с помощью:

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

- Когда сертификат предоставляют на Сервере pki, на следующее сообщение GetCertInitial SCEP отвечают с сообщением HTTP типа содержимого **application/x-pki-message**, и тело, содержащее PKCS#7 со знаком, подписало данные. Этот PKCS7 подписался, данные содержат Статус SCEP как **Предоставленный**, и также PKCS7 окутал данные. Окутанные данные этого PKCS содержат предоставленный сертификат и RecipientInfo, который является subject-name и серийным номером подписанного сертификата во время начальной регистрации и активного сертификата идентификации во время повторного зачисления.

  PKCS7 окутал данные, также содержит симметричный ключ, зашифрованный с открытым ключом получателя (для которого новый сертификат предоставили). Принимающий маршрутизатор дешифрует его с помощью секретного ключа. Этот ясный симметричный ключ тогда используется для дешифрования окутанных данных PKCS#7, показывая новый сертификат идентификации.



- На данном этапе IOS заменяет существующий сертификат идентификации новым сертификатом сразу. И если **восстанавливают**, был настроен, теневая пара согласованных ключей заменяет активную пару согласованных ключей также.

- Кроме того, дата завершения нового сертификата по сравнению с датой завершения

сертификата CA, чтобы определить, должен ли таймер RENEW инициализироваться, или таймер SHADOW должен инициализироваться, как объяснено здесь <href Типы Обновления Сертификата клиента - RENEW и SHADOW>