Обзор протокола SCEP

Содержание

Введение

Этот документ описывает Протокол SCEP (SCEP), который является протоколом, используемым для регистрации и других операций Инфраструктуры открытых ключей (PKI).

Общие сведения

SCEP был первоначально разработан Cisco и задокументирован в Проект инженерной группы по развитию Интернета (IETF).

Его главные характеристики:

- Запрос/ответ, основанный на модели на HTTP (метод GET; дополнительная поддержка для метода POST)
- Только поддерживает основанную на RSA криптографию
- Использует PKCS#10 в качестве формата запроса сертификата
- Использует PKCS#7 для передачи криптографически со знаком / зашифрованные сообщения
- Поддерживает асинхронное предоставление сервером, с обычным опросом запрашивающей стороной
- Ограничил поддержку извлечения Списка отозванных сертификатов (CRL) (предпочтительный способ через запрос CRL Distribution Point (CDP) для причин масштабируемости),
- Не поддерживает онлайновое аннулирование сертификата (должен быть сделан оффлайн через другие средства),
- Требует использования поля **пароля вызова** в Запросе подписи сертификата (CSR), который должен быть разделен только между сервером и запрашивающей стороной

Регистрация и использование SCEP обычно придерживаются этого рабочего потока:

1. Получите копию сертификата Центра сертификации (CA) и проверьте его.
2. Генерируйте CSR и передайте его надежно к CA.
3. Опросите сервер SCEP, чтобы проверить, был ли подписан сертификат.
4. Повторно зарегистрируйтесь по мере необходимости для получения нового сертификата до истечения текущего сертификата.
5. Получите CRL по мере необходимости.

Аутентификация CA

SCEP использует сертификат CA для обеспечения обмена сообщениями для CSR. В результате необходимо получить копию сертификата CA. Операция **GetCACert** используется.

Запрос

Запрос отправлен как HTTP-запрос GET. Захват пакета для запроса выглядит подобным этому:

```
GET /cgi-bin/pkiclient.exe?operation=GetCACert
```

Ответ

Ответ является просто закодированным двоичными файлами сертификатом CA (X.509). Клиент должен проверить, что сертификату CA доверяют посредством исследования отпечатка пальца/хэша. Это должно быть сделано через внеполосный метод (телефонный звонок системному администратору или предварительное конфигурирование отпечатка пальца в точке доверия).

Клиентская регистрация

Запрос

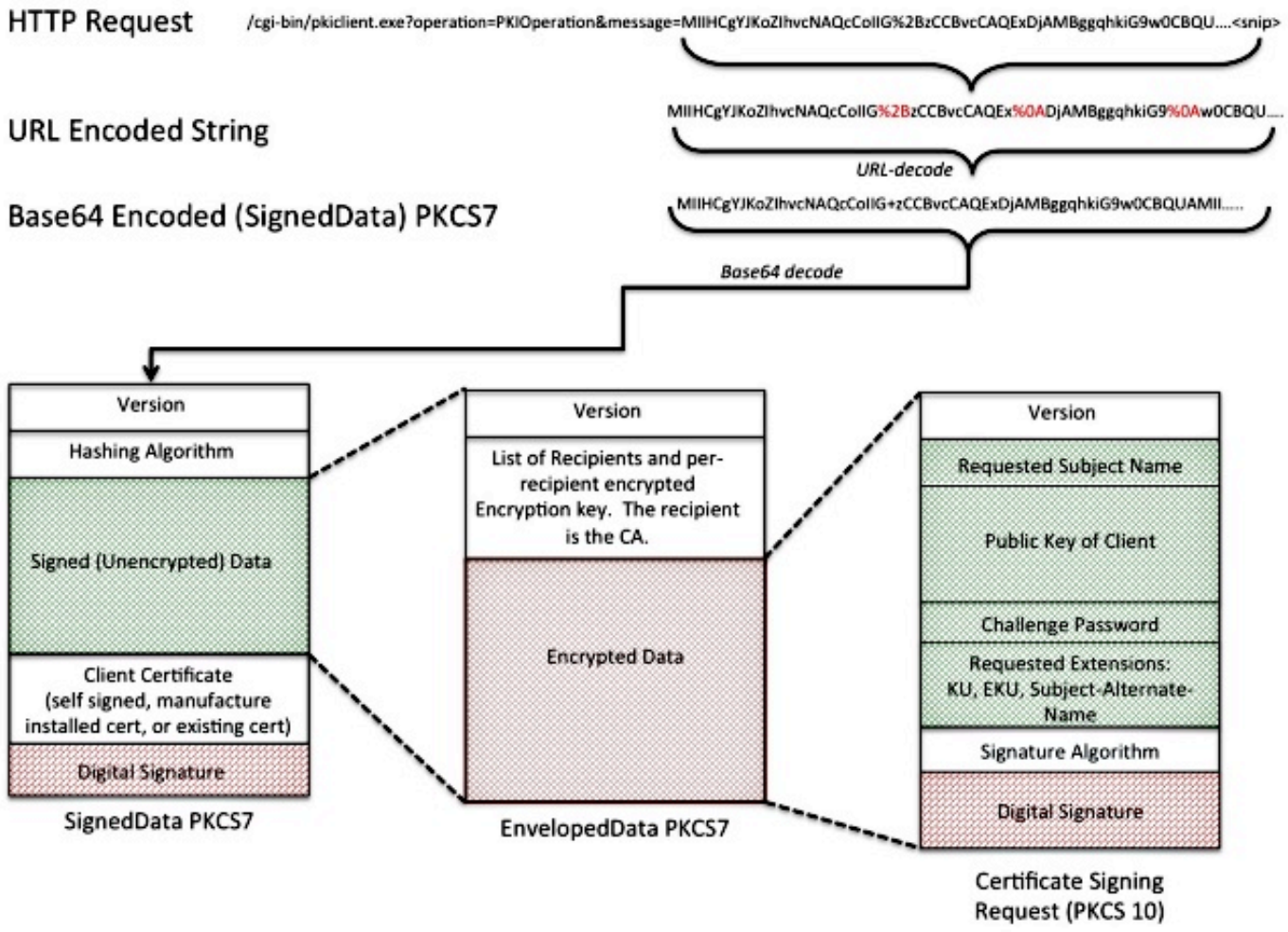
Запрос регистрации передается как HTTP-запрос GET. Захват пакета для запроса выглядит подобным этому:

```
/cgi-bin/pkiclient.exe?operation=PKIOperation&message=



MIIHCgYJKoZIhvcNAQcCoIIG%2BzCCBvcCAQExDjA.....<snip>
```

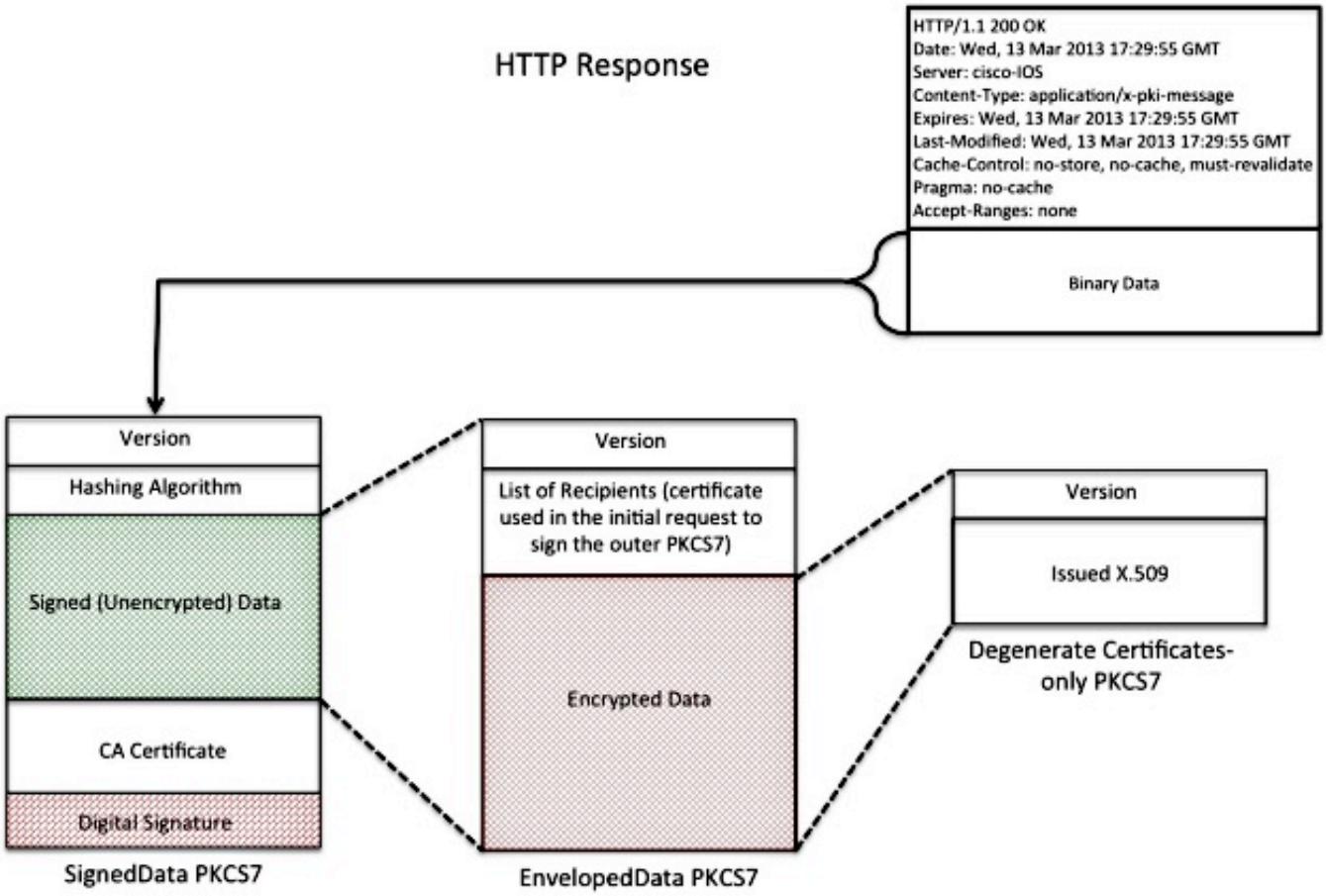
1. Текст после "сообщения =" является URL Закодированная Строка, которая извлечена из строки запроса GET.
2. Текст является тогда URL, Декодируемым в строку текста ASCII. Та текстовая строка является закодированным Base64 SignedData PKCS#7.
3. SignedData PKCS#7 подписан клиентом с одним из этих сертификатов; это используется, чтобы доказать, что клиент передал его и что это не было изменено в пути:
   Подписанный сертификат (используемый после начальной регистрации)Изготовитель установленный сертификат (MIC)Текущая сертификация, которая скоро истекает (повторное зачисление)
4. "Данные Со знаком" часть SignedData PKCS#7 являются EnvelopedData PKCS#7.
5. EnvelopedData PKCS#7 является контейнером, который содержит "Зашифрованные данные" и "ключ расшифровки". Ключ расшифровки зашифрован с Открытым ключом получателя. В этом конкретном случае получатель является CA; в результате. Только CA может фактически дешифровать "Зашифрованные данные".
6. Часть "Зашифрованных данных" Окутанного PKCS#7 является CSR (PKCS#10).



Ответ

Ответ на запрос регистрации SCEP является одним из трех типов:

- **Отклонение** - запрос отклонен администратором для любого количества причин, таких как:
  Недопустимый размер ключаНедопустимый пароль вызоваCA не мог проверить запросЗапрос попросил атрибуты, чтобы CA не авторизовалЗапрос был подписан идентичностью, которой не доверяет CA
- **При ожидании** - администратор CA еще не рассмотрел запрос.
- **Успех** - запрос принят, и подписанный сертификат включен. Подписанный сертификат проводится в специальном типе PKCS#7, названного "Вырожденным PKCS#7 Только для сертификатов", который является специальным контейнером, который может держать один или несколько X.509 или CRL, но не содержит информационное наполнение со знаком или информационное наполнение зашифрованных данных.



Клиентское повторное зачисление

До окончания срока действия сертификата клиент должен получить новый сертификат. Существует небольшое поведенческое различие между обновлением и одновременным нажатием клавиш. Когда сертификат ID клиентского истечения подходов, и его дата окончания действия не является тем же (ранее, чем) как дата окончания действия сертификата CA, обновление происходит. Когда истечение подходов сертификата ID, и его дата окончания действия совпадает с датой окончания срока действия сертификата CA, одновременное нажатие клавиш происходит.

Обновление

Поскольку дата окончания действия сертификата ID приближается, клиент SCEP мог бы хотеть получить новый сертификат. Клиент генерирует CSR и проходит процесс Регистрации (как определено ранее). Текущий сертификат используется для подписания SignedData PKCS#7, который в свою очередь удостоверяет личность к CA. На получение нового сертификата, клиент сразу удаляет текущий сертификат и заменяет его новым, законность которого сразу запускается.

Rollover

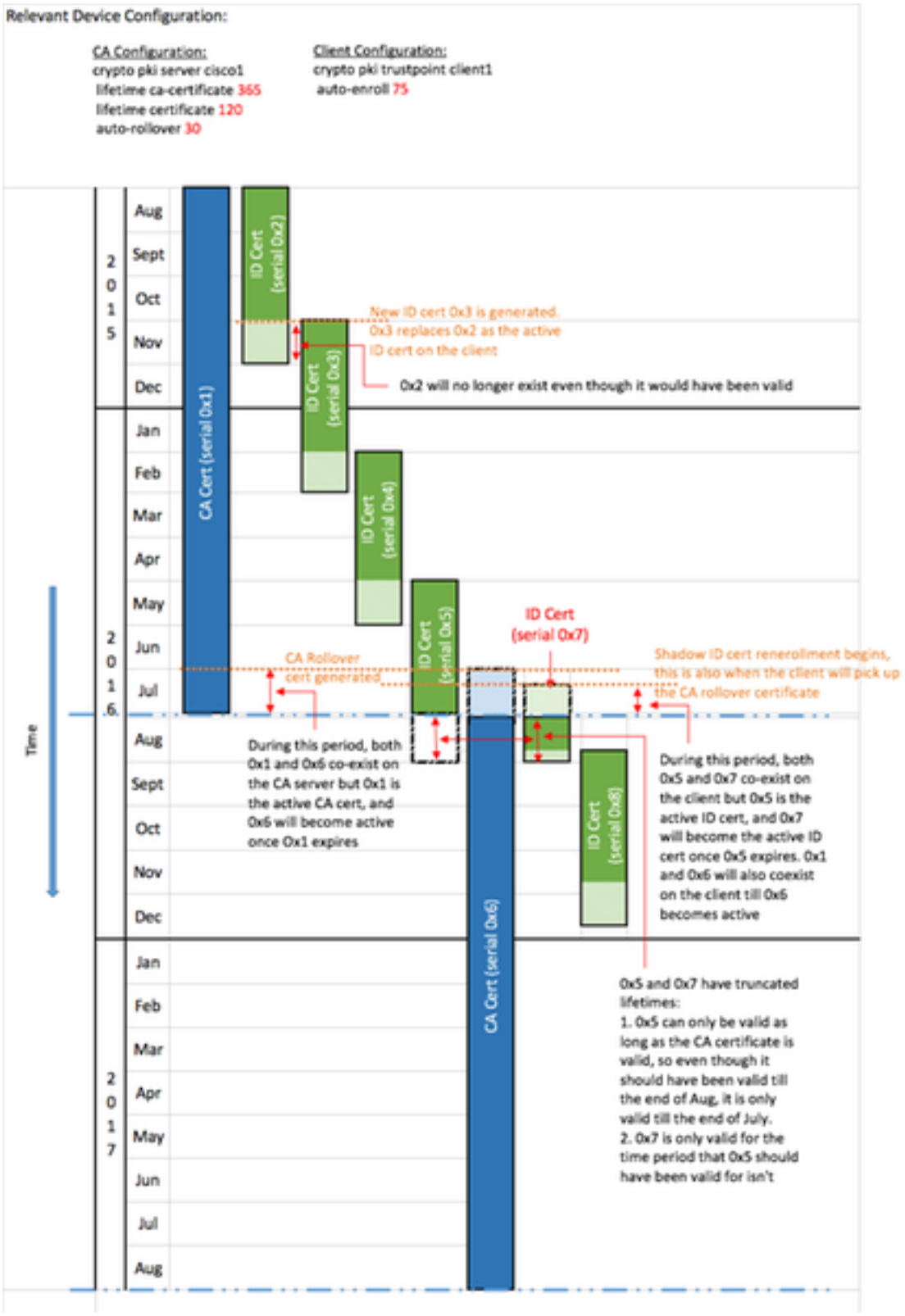
Одновременное нажатие клавиш является особым случаем, где сертификат CA истекает, и генерируется новый сертификат CA. CA генерирует новый сертификат CA, который

становится допустимым, как только истекает текущий сертификат CA. CA обычно генерирует эту "Тень CA" сертификат некоторое время до времени одновременного нажатия клавиш, потому что это необходимо для генерации "Теневого ID" сертификаты для клиентов.

Когда сертификат ID клиента SCEP приближается к истечению, запросы клиента SCEP CA для "Тени CA" Сертификат. Это сделано с операцией **GetNextCACert** как показано здесь:

```
GET /cgi-bin/pkiclient.exe?operation=GetNextCACert
```

Как только у клиента SCEP есть "Тень CA" сертификат, она запрашивает "Теневой ID" сертификат после обычной процедуры регистрации. CA подписывает "Теневой ID" сертификат с "Тенью CA" сертификат. В отличие от обычного запроса на обновление, "Теневой ID" сертификат, который возвращен, становится допустимым во время истечения сертификата CA (одновременное нажатие клавиш). В результате клиент должен поддержать копию пред - и сертификаты постодновременного нажатия клавиш и для CA и для сертификата ID. Во время истечения CA (одновременное нажатие клавиш) клиент SCEP удаляет текущий сертификат CA и сертификат ID и заменяет их "Теневыми" копиями.



Составляющий компоненты

Эта структура используется в качестве составляющий компонентов SCEP.

Примечание: PKCS#7 и PKCS#10 не специфичны для SCEP.

PKCS#7

PKCS#7 является определенным форматом данных, который позволяет данным быть подписанными или зашифрованными. Формат данных включает исходные данные и связанные метаданные, необходимые для выполнения криптографической операции.

Конверт со знаком (SignedData)

Конверт со знаком является форматом, который несет данные и подтверждает, что инкапсулированные данные не изменены в пути через цифровые подписи. Это включает эту информацию:

```
SignedData &colon;:= SEQUENCE {
version CMSVersion,
digestAlgorithms DigestAlgorithmIdentifiers,
encapContentInfo EncapsulatedContentInfo,
certificates [0] IMPLICIT CertificateSet OPTIONAL,
crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
signerInfos SignerInfos }
```

- Номер версии - С SCEP, версия 1 используется.
- Список Используемых Алгоритмов выборки сообщений - С SCEP, существует только одно Подписывающее лицо и таким образом только один Алгоритм хеширования.
- Реальные данные, которые подписаны - С SCEP, это - Окутанный формат данных PKCS#7 (Зашифрованный Конверт).
- Если вы повторно регистрируетесь, список сертификатов подписывающих лиц - С SCEP, это - подписанный сертификат на начальной регистрации или текущем сертификате.
- Список подписывающих лиц и отпечатка пальца, генерируемого каждым подписывающим лицом - С SCEP, существует только одно подписывающее лицо.

Инкапсулировавшие данные не зашифрованы или запутаны. Этот формат просто обеспечивает защиту против сообщения, которое изменено.

Окутанные данные (EnvelopedData)

Окутанный Формат данных несет данные, которые зашифрованы и могут только быть дешифрованы указанным получателем (получателями). Это включает эту информацию:

```
EnvelopedData &colon;:= SEQUENCE {
version CMSVersion,
originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
recipientInfos RecipientInfos,
encryptedContentInfo EncryptedContentInfo,
unprotectedAttrs [1] IMPLICIT UnprotectedAttributes OPTIONAL }
```

- Номер версии - С SCEP, версия 0 используется.
- Список каждого из получателей и связанного зашифрованного ключа шифрования данных - С SCEP, существует только один получатель (для запросов: сервер CA; для ответов: клиент).
- Зашифрованные данные - Это зашифровано со случайным образом генерируемым ключом (который был зашифрован с открытым ключом получателя).

PKCS#10

PKCS#10 описывает формат CSR. CSR содержит информацию, которую клиенты запрашивают быть включенными в их сертификатах:

- Имя субъекта
- Копия открытого ключа
- (Дополнительный) пароль вызова
- Любые расширения сертификата reqested, такие как:
  Ключевое использование (KU)Расширенное использование ключа (EKU)Альтернативное имя субъекта (SAN)Универсальное главное имя (UPN)
- Отпечаток пальца запроса

Вот пример CSR:

```
Certificate Request:
Data&colon;
Version: 0 (0x0)
Subject: CN=scepclient
Subject Public Key Info:

Public Key Algorithm: rsaEncryption Public-Key: (1024 bit)
Modulus:
00:cd:46:5b:e2:13:f9:bf:14:11:25:6d:ff:2f:43:
64:75:89:77:f6:8a:98:46:97:13:ca:50:83:bb:10:
cf:73:a4:bc:c1:b0:4b:5c:8b:58:25:38:d1:19:00:
a2:35:73:ef:9e:30:72:27:02:b1:64:41:f8:f6:94:
7b:90:c4:04:28:a1:02:c2:20:a2:14:da:b6:42:6f:
e6:cb:bb:33:c4:a3:64:de:4b:3a:7d:4c:a0:d4:e1:
b8:d8:71:cc:c7:59:89:88:43:24:f1:a4:56:66:3f:
10:25:41:69:af:e0:e2:b8:c8:a4:22:89:55:e1:cb:
00:95:31:3f:af:51:3f:53:ad
Exponent: 65537 (0x10001)
Attributes:
challengePassword :
Requested Extensions:
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Subject Alternative Name:
DNS:webserver.example.com
Signature Algorithm: sha1WithRSAEncryption
8c:d6:4c:52:4e:c0:d0:28:ca:cf:dc:c1:67:93:aa:4a:93:d0:
d1:92:d9:66:d0:99:f5:ad:b4:79:a5:da:2d:6a:f0:39:63:8f:
e4:02:b9:bb:39:9d:a0:7a:6e:77:bf:d2:49:22:08:e2:dc:67:
ea:59:45:8f:77:45:60:62:67:64:1d:fe:c7:d6:a0:c3:06:85:
e8:f8:11:54:c5:94:9e:fd:42:69:be:e6:73:40:dc:11:a5:9a:
f5:18:a0:47:33:65:22:d3:45:9f:f0:fd:1d:f4:6f:38:75:c7:
a6:8b:3a:33:07:09:12:f3:f1:af:ba:b7:cf:a6:af:67:cf:47: 60:fc
```

Дополнительные сведения

- Проект IETF SCEP
- Устаревший SCEP с помощью Руководства Конфигурации интерфейса командой строки
- Поддержка SCEP Настройки BYOD

Приложение

Запросы SCEP

Формат сообщения запроса

Запросы отправлены с GET HTTP формы :

```
GET CGI-path/pkiclient.exe?operation=operation&message=message HTTP/version
```

Где:

- **CGI - путь** зависит от сервера и точек к программе Общего шлюзового интерфейса (CGI), которая обрабатывает запросы SCEP: Cisco IOS® CA использует пустую строку пути.Microsoft CA использует**/certsrv/mscep/mscep.dll**, который указывает к службе Network Device Enrollment Service (NDES) MSCEP/сервис IIS.
- **Операция** определяет операцию, которая выполнена.
- **Сообщение** несет дополнительные данные для той операции (и это может быть пусто, если никакие реальные данные не требуются).

С методом GET **часть сообщения** является или открытым текстом, или Выдающиеся правила кодирования (DER) - закодировали PKCS#7, преобразованный в Base64. Если бы метод POST поддерживается, содержание, которое было бы передано в кодировании Base64 с GET, могло бы быть передано в двоичном формате с POST вместо этого.

Схематическое представление

Возможные значения для **операций** и их связанные значения **сообщения**:

- **операция** = `PKIOperation`: **messageis структура SCEP pkiMessage** , на основе PKCS#7 и закодированный с DER и Base64. **pkiMessage** структура может иметь эти типы: **PKCSReq**: CSR PKCS#10**GetCertInitial**: опрос для статуса предоставления CSR**GetCert** или **GetCRL**: сертификат или извлечение CRL
- **операция** = **GetCACert**, **GetNextCACert** или (дополнительный) **GetCACaps**: **сообщение** может быть опущено или может быть установлено в название, которое определяет CA.

Ответы SCEP

Формат ответного сообщения

Ответы SCEP возвращены как стандартное содержание HTTP с **Типом содержимого**, который зависит от исходного запроса, и тип данных возвратился. Содержание DER возвращено как двоичные файлы (не в Base64 что касается запроса). Содержание PKCS#7 могло бы или не могло бы содержать, шифровал/подписывал окутанные данные; если это не делает (только содержит ряд сертификатов), это упоминается как **вырожденный** PKCS#7.

Типы содержимого

Возможные значения для **Типа содержимого**:

application/x-pki-message:

- в ответ на операцию **PKIOperation**, с **pkiMessage** типа: **PKCSReq, GetCertInitial, GetCert или GetCRL**
- орган по ответу является **pkiMessage** типа: **CertRep**

application/x-x509-ca-cert:

- в ответ на операцию **GetCACert**
- орган по ответу является закодированным DER сертификатом CA X.509

application/x-x509-ca-ra-cert:

- в ответ на операцию **GetCACert**
- орган по ответу является закодированным DER вырожденным PKCS#7, который содержит сертификаты RA и CA

application/x-x509-next-ca-cert:

- в ответ на операцию **GetNextCACert**
- орган по ответу является изменением **pkiMessage** типа: **CertRep**

pkiMessage Структура

OID SCEP

```
GET CGI-path/pkiclient.exe?operation=operation&message=message HTTP/version
```

SCEP pkiMessage

- **PKCS#7 SignedData**
- PKCS#7 EnvelopedData (названный **pkcsPKIEnvelope**; дополнительный, зашифрованный получателю сообщения)
  messageData (CSR, свидетельство, CRL...)
- **SignerInfo** с **authenticatedAttributes**:
  transactionID, **messageType**, **senderNoncepkiStatus**, **recipientNonce** (только ответ)**failInfo** (ответ + только сбой)

SCEP messageType

- запрос:
  PKCSReq (19): CSR PKCS#10**GetCertInitial** (20): опрос хранилища сертификатов**GetCert** (21): извлечение сертификата**GetCRL** (22): извлечение CRL
- ответ:
  CertRep (3): ответ на сертификат или запрос CRL

SCEP pkiStatus

- **SUCCESS** (0): предоставленный запрос (ответ в pkcsPKIEnvelope)
- **СБОЙ** (2): запрос отклонил (детализирует в атрибуте failInfo),
- **ОЖИДАЯ** (3): запрос ждет ручного утверждения