# Vista geral do protocolo simple certificate enrollment

## Índice



## Introdução

Este documento descreve o protocolo simple certificate enrollment (SCEP), que é um protocolo usado para o registro e as outras operações do Public Key Infrastructure (PKI).

## Informações de Apoio

O SCEP foi desenvolvido originalmente por Cisco, e é documentado em um esboço do Internet

Engineering Task Force (IETF).

Suas características principal são:

- Modelo do pedido/resposta baseado em HTTP (método GET; suporte opcional para o método do CARGO)
- Somente os apoios RSA-basearam a criptografia
- Usos PKCS#10 como o formato do pedido do certificado
- Os usos PKCS#7 a fim transportar criptograficamente assinaram/mensagens codificada
- Apoia a concessão assíncrona pelo server, com votação regular pelo solicitador
- Limitou o apoio da recuperação do Certificate Revocation List (CRL) (o método preferido é com uma pergunta do CRL Distribution Point (CDP), para motivos de escalabilidade)
- Não apoia a revogação de certificado em linha (deve ser feito off line com os outros meios)
- Exige o uso de um campo de **senha do desafio** dentro da solicitação de assinatura de certificado (CSR), que deve ser compartilhada somente entre o server e o solicitador

O registro e o uso do SCEP seguem geralmente este fluxo de trabalho:

1. Obtenha uma cópia do certificado do Certificate Authority (CA) e valide-a.
2. Gerencia um CSR e envie-o firmemente a CA.
3. Vote o server SCEP a fim verificar se o certificado esteja assinado.
4. Re-registre como necessário a fim obter um certificado novo antes da expiração do certificado atual.
5. Recupere o CRL como necessário.

# Autenticação de CA

O SCEP usa o certificado de CA a fim fixar a troca da mensagem para o CSR. Em consequência, é necessário obter uma cópia do certificado de CA. A operação de **GetCACert** é usada.

## Requisição

O pedido é enviado como um pedido HTTP GET. Uma captura de pacote de informação para o pedido olha similar a esta:

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

## Resposta

A resposta é simplesmente o certificado de CA binário-codificado (X.509). O cliente precisa de validar que o certificado de CA está confiado através de um exame da impressão digital/mistura. Isto tem que ser feito através de um método out-of-band (uma chamada telefônica a um administrador de sistema ou da PRE-configuração da impressão digital dentro do ponto confiável).

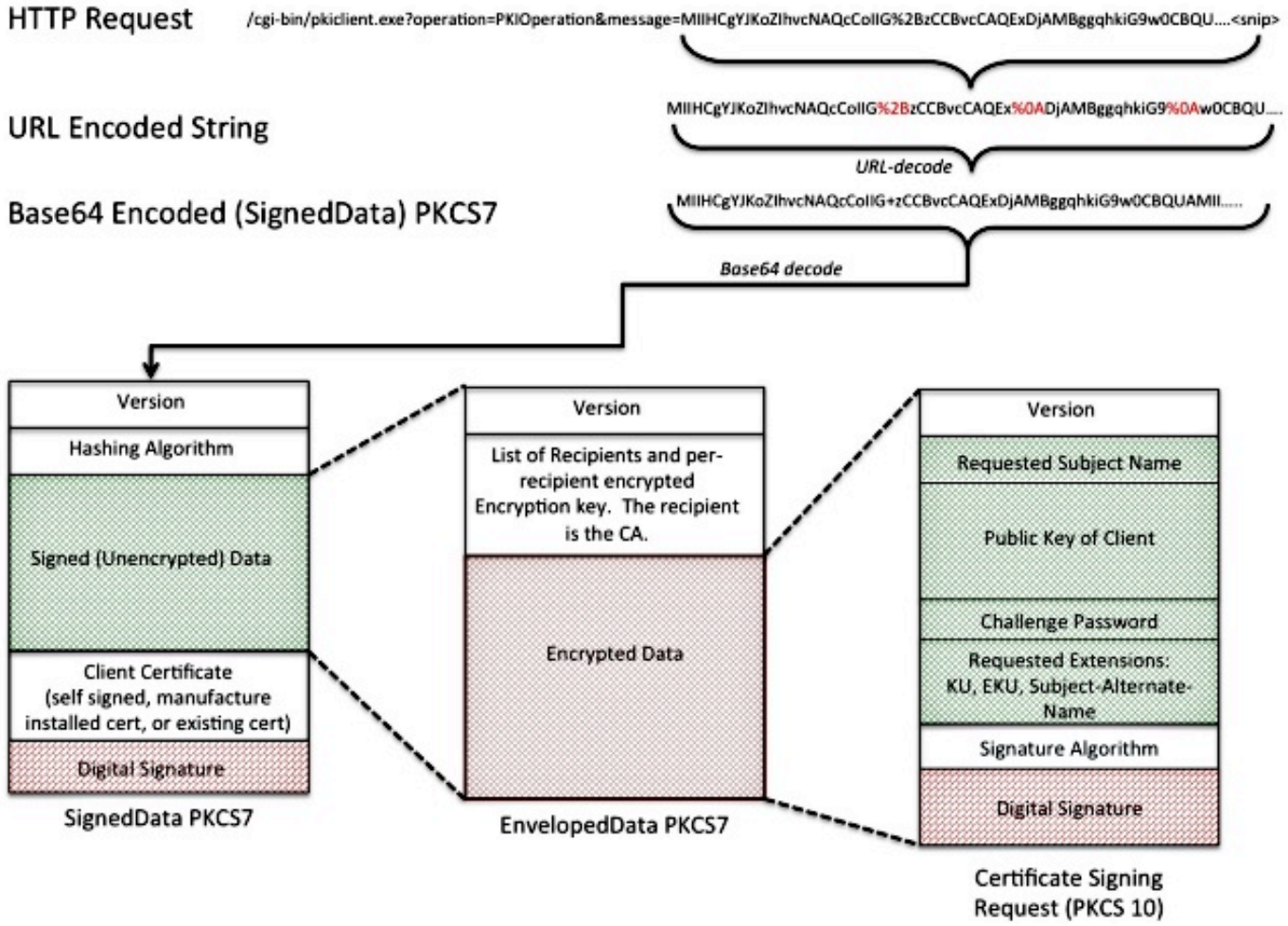
# Registro do cliente

## Requisição

O pedido do registro é enviado como um pedido HTTP GET. Uma captura de pacote de informação para o pedido olha similar a esta:

```
/cgi-bin/pkiclient.exe?operation=PKIOperation&message=
MIIHCgYJKoZIhvcNAQcCoIIG%2BzCCBvcCAQExDjA.....<snip>
```

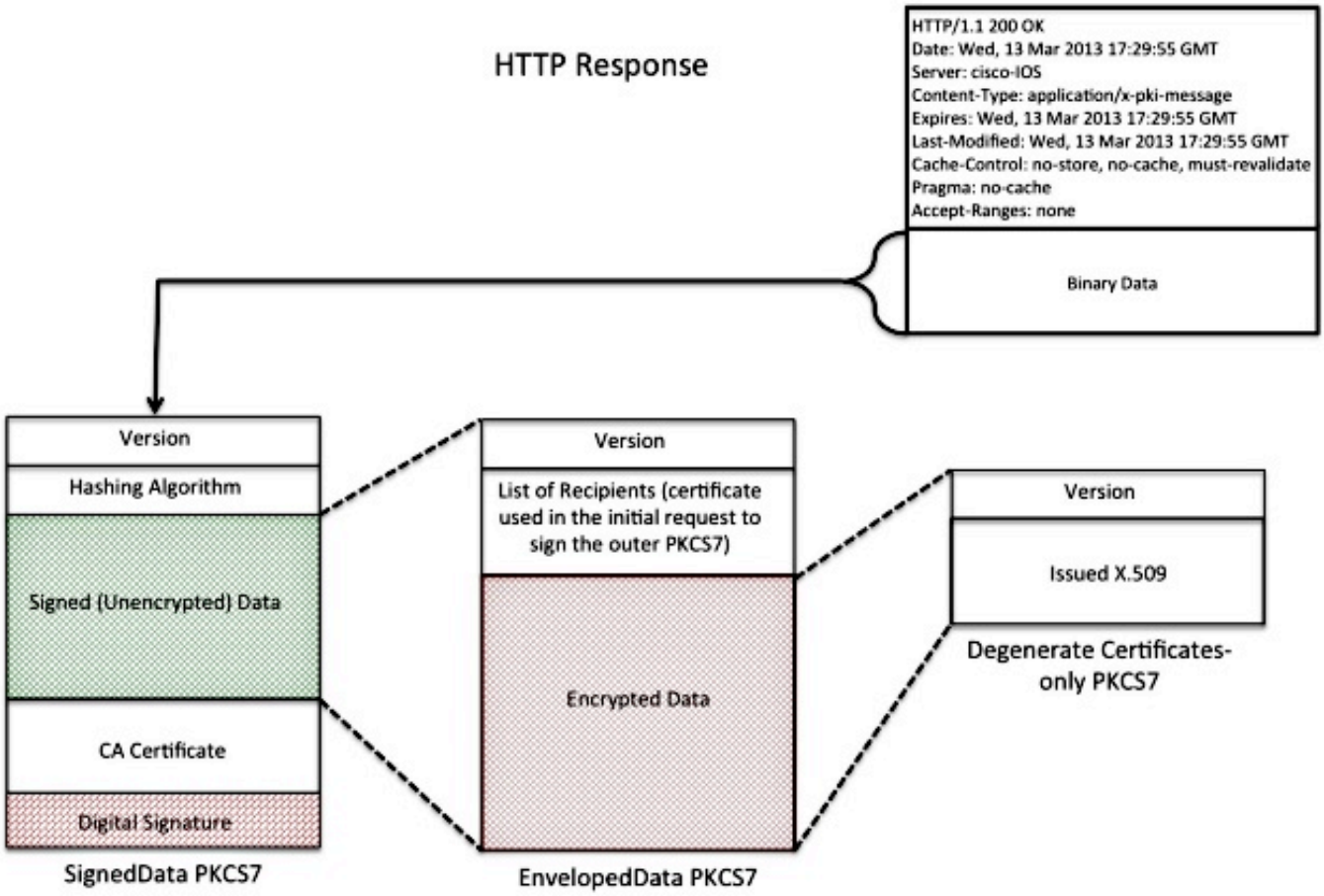
1. O texto depois que o “message=” é uma corda codificada URL, que esteja extraída da corda do pedido GET.
2. O texto é então URL descodificada em uma corda de texto de ASCII. Essa sequência de caracteres de texto é um Base64-encoded SignedData PKCS#7.
3. O SignedData PKCS#7 é assinado pelo cliente com um destes Certificados; usa-se para mostrar que o cliente o enviou e que não esteve alterado no trânsito:
   Um certificado auto-assinado (usado em cima do registro inicial)Um certificado instalado fabricante (MIC)Uma certificação atual que expire logo (nova inscrição)
4. “A parcela dos dados assinados” do SignedData PKCS#7 é um EnvelopedData PKCS#7.
5. O EnvelopedData PKCS#7 é um recipiente que contenha “dados criptografados” e a “chave de descriptografia.” A chave de descriptografia é cifrada com chave pública do receptor. Neste caso específico, o receptor é CA; em consequência. Somente CA pode realmente decifrar os “dados criptografados.”
6. A parcela dos “dados criptografados” do PKCS#7 envolvido é o CSR (PKCS#10).




## Resposta

A resposta ao pedido do registro SCEP é um de três tipos:

- **Rejeição** - O pedido é rejeitado pelo administrador para todo o número de razões, como: Tamanho chave inválidoSenha do desafio inválidaCA não podia validar o pedidoO pedido pediu os atributos que CA não autorizouO pedido foi assinado por uma identidade que CA não confiasse
- **Durante** - O administrador de CA não reviu o pedido ainda.
- **Sucesso** - O pedido é aceitado e o certificado assinado é incluído. O certificado assinado é guardado dentro de um tipo especial de PKCS#7 chamado “Certificados-Somente degenerate PKCS#7," que é um recipiente especial que possa guardar um ou vário o X.509 ou os CRL, mas não contém um payload assinada ou dos dados criptografados.




# Nova inscrição do cliente

Antes da expiração do certificado, o cliente precisa de obter um certificado novo. Há uma leve diferença comportável entre a renovação e o derrubamento. A renovação acontece quando o certificado ID do cliente aproxima a expiração, e sua data de expiração não é a mesma (mais cedo do que) que a data de expiração do certificado de CA. O derrubamento acontece quando o certificado ID aproxima a expiração, e sua data de expiração é a mesma que a data de expiração do certificado de CA.

## Renovação

Porque a data de expiração de um certificado ID se aproxima, um cliente SCEP pôde querer obter um certificado novo. O cliente gerencie um CSR e atravessa o processo do registro (como

definido previamente). O certificado atual é usado a fim assinar o SignedData PKCS#7, que prova por sua vez a identidade ao CA após recepção do certificado novo, o cliente suprime imediatamente do certificado atual e substitui-o com o novo, cujos começos de validez imediatamente.

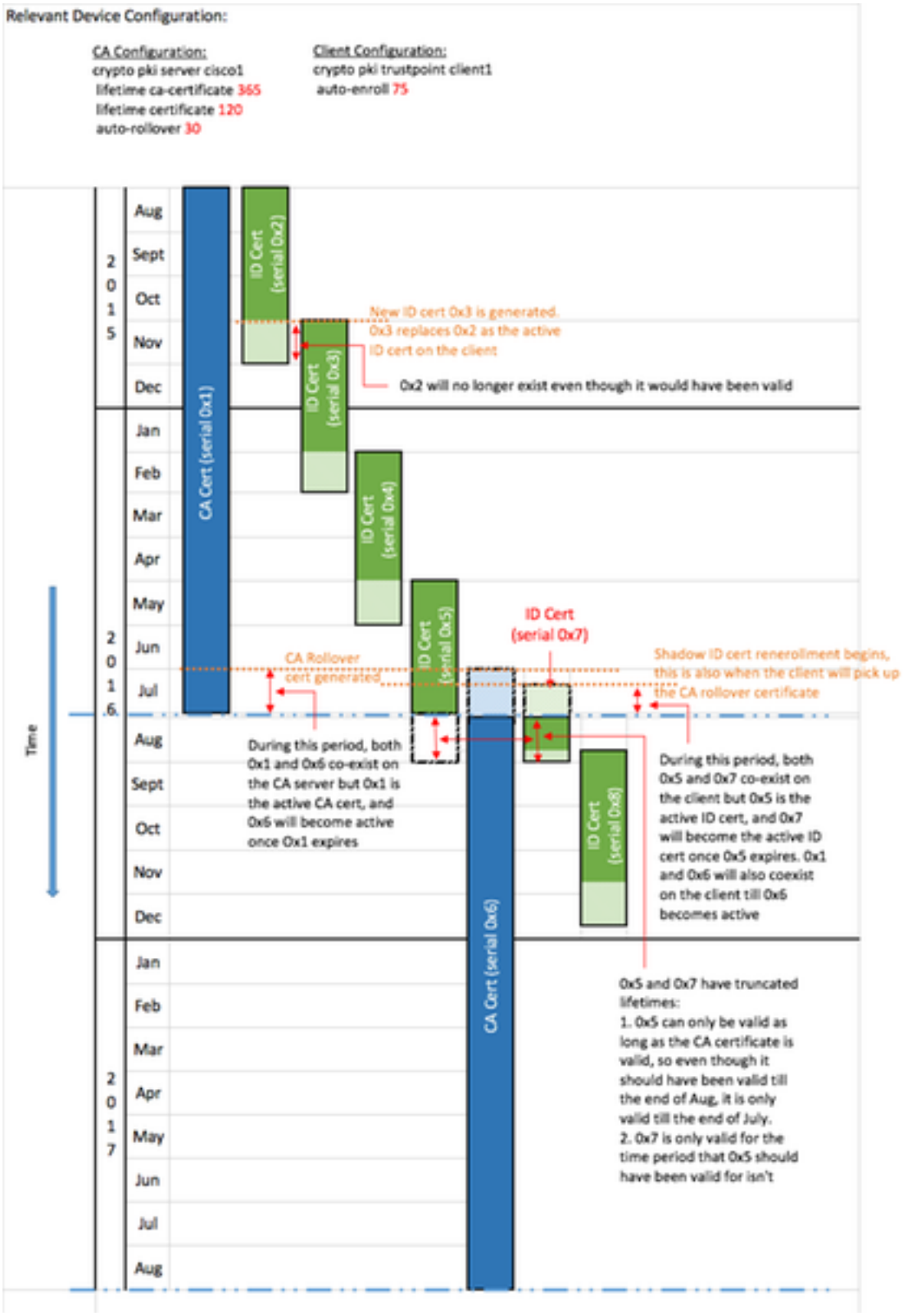
## Derrubamento

O derrubamento é um caso especial onde o certificado de CA expire e um certificado de CA novo seja gerado. CA gerencie um certificado de CA novo que se torne válido uma vez o certificado de CA atual expire. CA gerencie geralmente este da “certificado de CA sombra” alguma hora antes do tempo do derrubamento, porque é precisado a fim gerar da “Certificados sombra ID” para os clientes.

Quando o certificado ID do cliente SCEP aproximar a expiração, as consultas cliente SCEP CA para da “o certificado de CA sombra”. Isto é feito com a operação de **GetNextCACert** como mostrado aqui:

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

Uma vez que o cliente SCEP tem da “o certificado de CA sombra”, pede da “um certificado sombra ID” após o procedimento normal do registro. CA assina da “o certificado sombra ID” com da “o certificado de CA sombra”. Ao contrário de uma requisição de renovação normal, da “o certificado sombra ID” que é retornado torna-se válido na altura da expiração do certificado de CA (derrubamento). Em consequência, o cliente precisa de manter uma cópia do PRE e dos Certificados do cargo-derrubamento para CA e o certificado ID. Na altura da expiração de CA (derrubamento), o cliente SCEP suprime do certificado de CA atual e do certificado ID e substitui-os com as cópias da “sombra”.




# Montagens de bloco

Esta estrutura é usada como os montagens de bloco do SCEP.

**Note**: PKCS#7 e PKCS#10 não são SCEP-específicos.

## PKCS#7

PKCS#7 é um formato de dados definido que permita que os dados sejam assinados ou cifrados. O formato de dados inclui os dados originais e os metadata associados necessários a fim executar a operação criptográfica.

### Envelope assinado (SignedData)

O envelope assinado é um formato que leve dados e confirma que os dados encapsulados não estão alterados no trânsito através das assinaturas digital. Inclui esta informação:

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

- Número de versão - Com SCEP, versão 1 usada.
- Lista de algoritmos do resumo usados - Com SCEP, há somente um signatário e assim somente um algoritmo de hashing.
- Dados reais que são assinados - Com SCEP, este é um formato PKCS#7 Envolver-DATA (envelope cifrado).
- Lista de Certificados dos signatários - Com SCEP, este é um certificado auto-assinado no registro inicial ou o certificado atual se você re-se registra.
- Lista dos signatários e da impressão digital gerados por cada signatário - com SCEP, há somente um signatário.

Os dados encapsulados não são cifrados nem são confundidos. Este formato fornece simplesmente a proteção contra a mensagem que é alterada.

### Dados envolvidos (EnvelopedData)

O formato de dados envolvido leva os dados que são cifrados e podem somente ser decifrados pelo destinatário especificado. Inclui esta informação:

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

- Número de versão - Com SCEP, a versão 0 é usada.
- Lista de cada um dos receptores e do Data Encryption Key cifrado relacionado - com SCEP, há somente um receptor (para pedidos: o server de CA; para respostas: o cliente).
- Os dados criptografados - Isto é cifrado com uma chave aleatoriamente gerada (de que foi cifrado com a chave pública do receptor).

## PKCS#10

PKCS#10 descreve o formato de um CSR. Um CSR contém a informação que os clientes pedem sejam incluídos dentro de seus Certificados:

- Nome do sujeito
- Uma cópia da chave pública
- Uma senha do desafio (opcional)
- Algumas extensões de certificado reqested, como:
  Uso chave (KU)Uso chave prolongado (EKU)Nome alternativo sujeito (SAN)Nome principal universal (UPN)
- Uma impressão digital do pedido

Está aqui um exemplo de um 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
```

# Informações Relacionadas

- Esboço de IETF SCEP
- Legado SCEP usando o guia de configuração de CLI
- Configurando o apoio SCEP para BYOD

# Apêndice

## Pedidos SCEP

### Formato do mensagem request

Os pedidos são enviados com um HTTP GET do formulário:

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

Em que:

- **o CGI-PATH** é dependente do server e os pontos à interface de gateway comum (CGI) programam que os punhos SCEP pedem: O ® CA do Cisco IOS usa uma corda vazia do trajeto.Microsoft CA usa **/certsrv/mscep/mscep.dll**, que aponta ao serviço do serviço do registro do dispositivo de rede MSCEP/(NDE) IIS.
- **A operação** identifica a operação que é executada.
- **A mensagem** leva dados adicionais para essa operação (e ela pode estar vazio se nenhum dados real é exigido).

Com o método GET, a peça de **mensagem** é texto simples, ou as distintas regras da codificação (DER) - PKCS#7 codificado convertido a Base64. Se o método do CARGO é apoiado, satisfaça que seria enviado em Base64 que codifica com GET pôde ser enviado no formato binário com CARGO pelo contrário.

### Vista esquemática

Valores possíveis para **operações** e seus valores associados da **mensagem**:

- **operação** = `PKIOperation`: **messageis uma** estrutura do **pkiMessage** SCEP, com base em PKCS#7 e codificado com DER e Base64.a estrutura do **pkiMessage** pode ser destes tipos: **PKCSReq**: PKCS#10 CSR**GetCertInitial**: vatação para o CSR que concede o estado**GetCert** ou **GetCRL**: certificado ou recuperação CRL
- **operação** = **GetCACert**, **GetNextCACert**, ou **GetCACaps** (opcional): **a mensagem** pode ser omitida, ou pode ser ajustada a um nome que identifique CA.

## Respostas SCEP

### Formato do mensagem de resposta

As respostas SCEP são retornadas como o índice padrão HTTP, com um **tipo de conteúdo** que dependa da solicitação original e do tipo de dados retornados. O índice DER é retornado como o binário (não em Base64 quanto para ao pedido). O índice PKCS#7 pôde ou não pôde conter dados envolvidos cifrados/assinados; se não faz (contém somente um grupo de Certificados), está referido como um PKCS#7 **degenerate**.

### Tipos de conteúdo

Valores possíveis para o **tipo de conteúdo**:

**application/x-pki-message:**

- em resposta à operação de **PKIOperation**, com o **pkiMessage do** tipo: **PKCSReq, GetCertInitial, GetCert ou GetCRL**
- o corpo da resposta é um **pkiMessage do** tipo: **CertRep**

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

- em resposta à operação de **GetCACert**
- o corpo da resposta é o certificado de CA X.509 DER-codificado

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

- em resposta à operação de **GetCACert**
- o corpo da resposta é um PKCS#7 degenerate DER-codificado que contenha os Certificados de CA e RA

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

- em resposta à operação de **GetNextCACert**
- o corpo da resposta é uma variação de um **pkiMessage do** tipo: **CertRep**

## A estrutura do pkiMessage

### SCEP OID

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

### PkiMessage SCEP

- PKCS#7 **SignedData**
- PKCS#7 EnvelopedData (chamado **pkcsPKIEnvelope**; opcional, cifrado ao destinatário da mensagem)
  **messageData** (CSR, CERT, CRL,…)
- **SignerInfo** com **authenticatedAttributes**:
  **transactionID**, **messageType**, **senderNoncepkiStatus**, **recipientNonce** (resposta somente)**failInfo** (resposta + falha somente)

### MessageType SCEP

- pedido:
  **PKCSReq** (19): PKCS#10 CSR**GetCertInitial** (20): votação do certificado de registro**GetCert** (21): recuperação de certificado**GetCRL** (22): Recuperação CRL
- resposta:
  **CertRep** (3): resposta a certificate ou pedido CRL

### PkiStatus SCEP

- **SUCESSO** (0): pedido concedido (resposta no pkcsPKIEnvelope)
- **FALHA** (2): pedido rejeitado (detalhes no atributo do failInfo)
- **DURANTE** (3): o pedido espera a aprovação manual