# Eenvoudig protocol voor certificaatinschrijving

## Inhoud



## Inleiding

In dit document wordt het Simple certificaatprotocol (SCEP) beschreven, dat een protocol is dat wordt gebruikt voor inschrijving en andere PKI-activiteiten (Public Key Infrastructuur).

## Achtergrondinformatie

SCEP werd oorspronkelijk ontwikkeld door Cisco en is gedocumenteerd in een ontwerp van de Internet Engineering Task Force (IETF).

De belangrijkste kenmerken zijn:

- Aanvraag/antwoord-model gebaseerd op HTTP (GET methode; optionele ondersteuning voor POST-methode)
- Ondersteunt alleen op RSA gebaseerde cryptografie
- Gebruikt PKCS#10 als de indeling voor certificaataanvraag
- Gebruik PKCS#7 om cryptografisch ondertekende/versleutelde berichten over te brengen
- Ondersteunt asynchrone toekenning door de server, met regelmatige opiniepeiling door de verzoeker
- Heeft de beperkte steun van de Revocatielijst van het Certificaat (CRL) (de geprefereerde methode is door een vraag van het Verdelingspunt van CRL (CDP), om schaalbaarheidsredenen)
- Ondersteunt geen online herroeping van certificaten (moet offline geschieden via andere middelen)
- Vereist het gebruik van een **uitdagingswachtwoord** binnen de certificaataanvraag (CSR), die alleen tussen de server en de aanvrager moet worden gedeeld

In het algemeen volgt de inschrijving en het gebruik van SCEP deze werkstroom:

1. Verkrijg een kopie van het certificaat van de certificeringsinstantie (CA) en bevestig het.
2. Genereert een CSR en stuurt het veilig naar de CA.
3. Bezoek de SCEP server om te controleren of het certificaat is ondertekend.
4. Herinschrijving indien noodzakelijk om een nieuw certificaat te verkrijgen vóór het verstrijken van het huidige certificaat.
5. Neem het CRL indien nodig terug.

# CA-verificatie

Het SCEP gebruikt het CA-certificaat om de berichtuitwisseling voor de CSR te beveiligen. Bijgevolg moet een kopie van het CA-certificaat worden verkregen. De **GetCAC**inspoeling wordt gebruikt.

## Aanvragen

Het verzoek wordt verzonden als een HTTP GET aanvraag. Een pakketvastlegging voor het verzoek lijkt hierop:

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

## respons

De reactie is simpelweg het binair-gecodeerde CA certificaat (X.509). De cliënt moet valideren dat het CA-certificaat wordt vertrouwd door middel van een onderzoek van de vingerafdruk/hash. Dit moet gebeuren via een out-of-band methode (een telefoongesprek met een systeembeheerder of het vooraf instellen van de vingerafdruk binnen het trustpunt).

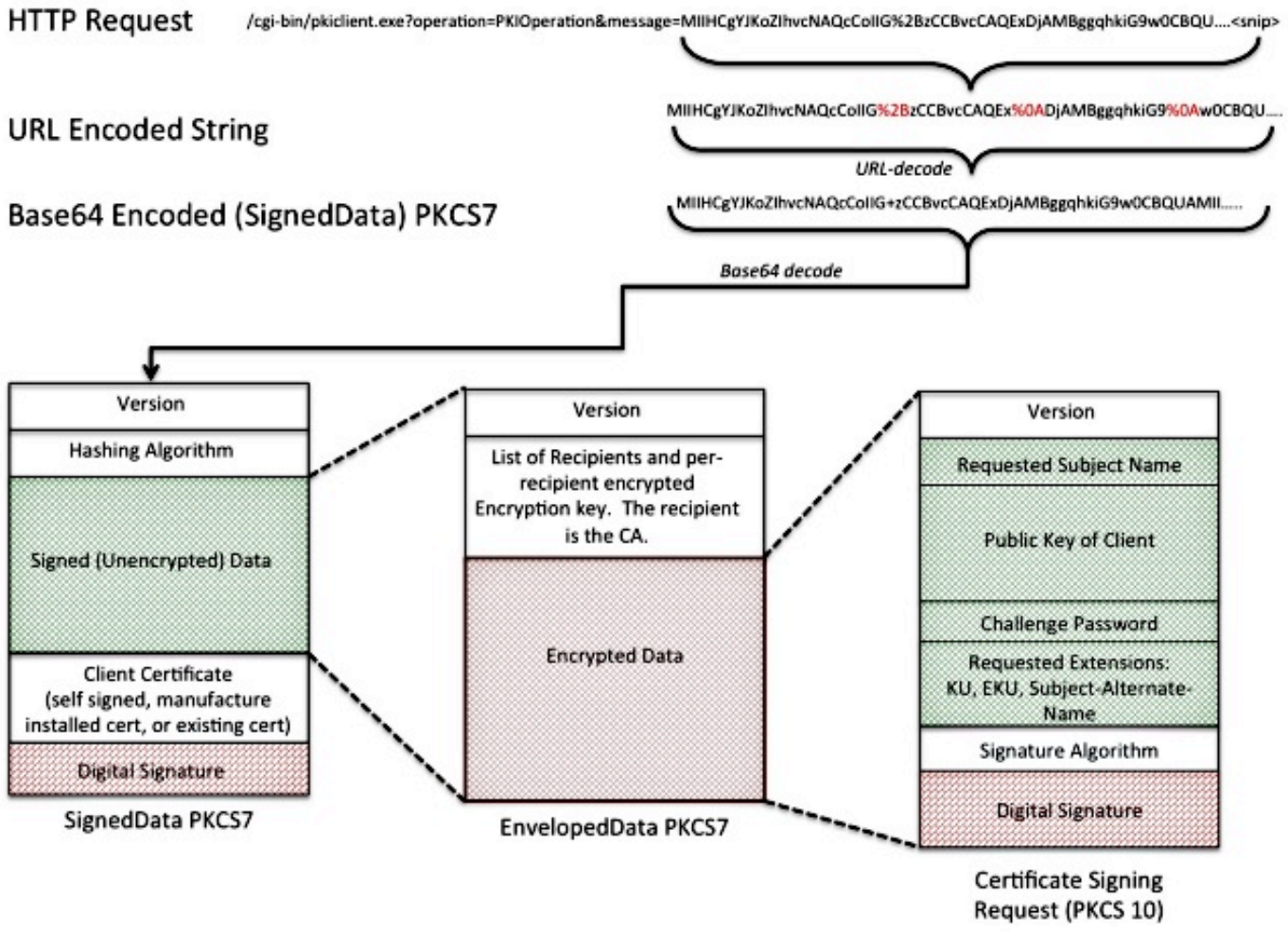
# Clientinschrijving

## Aanvragen

Het inschrijvingsverzoek wordt verstuurd als een HTTP GET aanvraag. Een pakketvastlegging voor het verzoek ziet er zo uit:

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

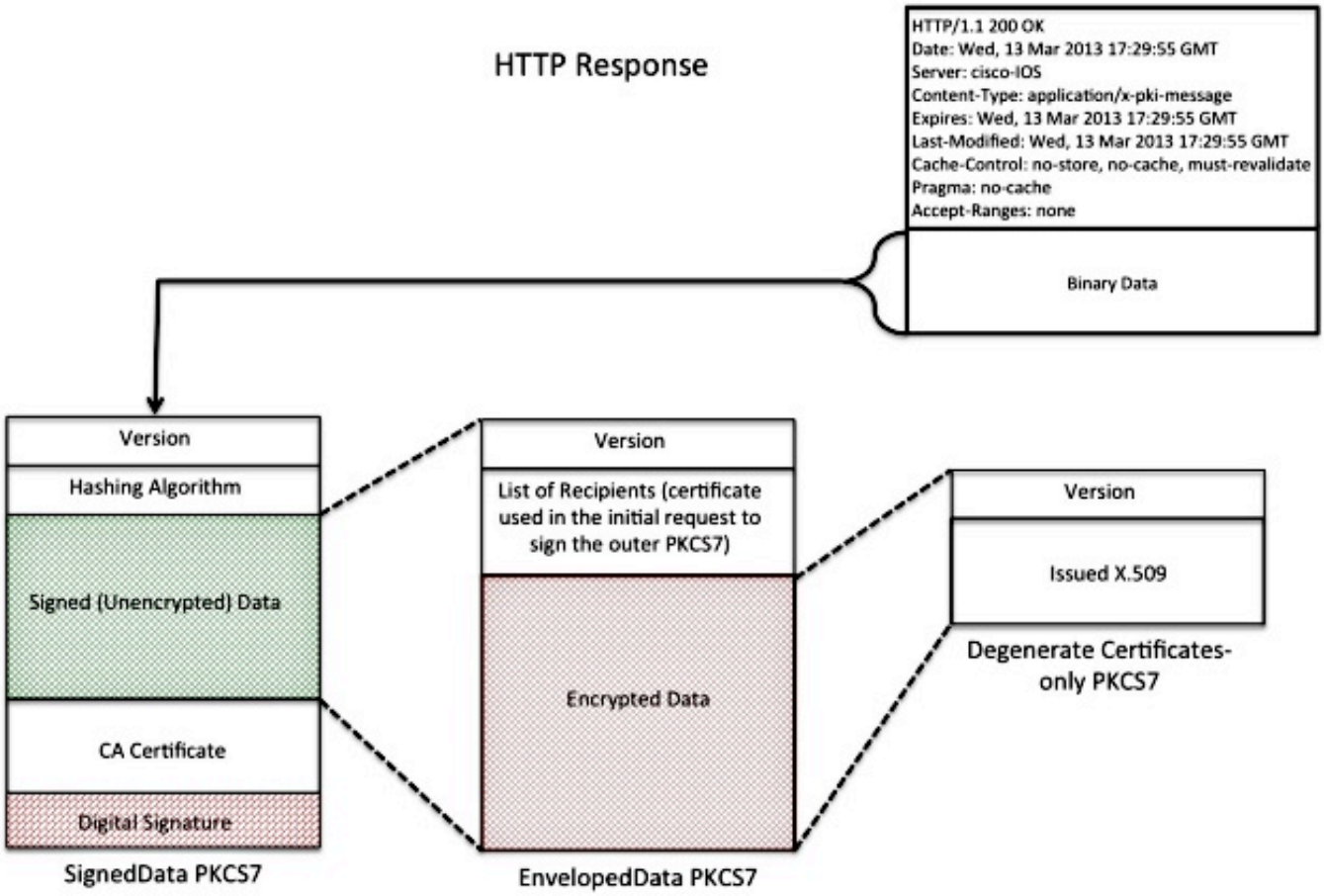
1. De tekst na "bericht=" is een URL Encoded String, die uit de GET request string wordt afgeleid.
2. De tekst wordt dan URL gedecodeerd in een ASCII tekststring. Die tekststring is een Base64-gecodeerde SignedData PKCS#7.
3. De SignedData PKCS#7 wordt door de cliënt met een van deze certificaten ondertekend; het bewijs wordt geleverd dat de cliënt het heeft verzonden en dat het niet is gewijzigd bij doorreis :
   Een zelfgetekend certificaat (gebruikt bij eerste inschrijving)Een door de fabrikant geïnstalleerd certificaat (MIC)Een huidige certificering die binnenkort afloopt (opnieuw inschrijven)
4. Het "Signed Data"-gedeelte van het SignedData PKCS#7 is een EnvelopedData PKCS#7.
5. The Enveloped Data PKCS#7 is een container die "Encrypted Data" en de "decryptie key" bevat. De decryptie sleutel is gecodeerd met de openbare sleutel van de ontvanger. In dit specifieke geval is de ontvanger de CA; dit heeft tot gevolg . Alleen CA kan de "Encrypted Data" decrypteren.
6. Het "Versleutelde gegevens"-gedeelte van de ontwikkelde PKCS#7 is de CSR (PKCS#10).



### respons

Het antwoord op het verzoek om inschrijving van het SCEP bestaat uit drie typen:

- **Afwijzen** - het verzoek wordt door de beheerder verworpen om een aantal redenen, zoals: Ongeldige sleutelgrootteOngeldig uitdagingswachtwoordDe CA kon het verzoek niet validerenHet verzoek om eigenschappen waarvoor de CA geen toestemming heeft gegevenHet verzoek werd ondertekend door een identiteit die de CA niet vertrouwt
- **In afwachting** - De CA-beheerder heeft het verzoek nog niet beoordeeld.
- **Succes** - Het verzoek wordt ingewilligd en het ondertekende certificaat wordt meegeleverd. Het ondertekende certificaat wordt bewaard in een speciaal type PKCS#7 genaamd "Degenerate Certificates-Only PKCS#7", een speciale container die één of meer X.509 of CRL's kan bevatten, maar geen ondertekende of gecodeerde gegevenslading bevat.




## Clientherinschrijving

Vóór het verstrijken van het certificaat moet de cliënt een nieuw certificaat krijgen. Er is een klein gedragsverschil tussen vernieuwing en omvergooien. Verlenging gebeurt wanneer het ID-certificaat van de cliënt op verloopdatum afloopt en de vervaldatum ervan niet dezelfde is (vroeger dan) als de verloopdatum van het CA-certificaat. Rollover gebeurt wanneer het ID-certificaat op verloopdatum nadert en de vervaldatum ervan is dezelfde als de vervaldatum van het CA-certificaat.

## Verlengen

Aangezien de verloopdatum van een identiteitsbewijs naderbij komt, zou een SCEP-cliënt een nieuw certificaat willen verkrijgen. De client genereert een CSR en gaat door het inlogproces (zoals eerder gedefinieerd). Het huidige certificaat wordt gebruikt om de SignedData PKCS#7 te ondertekenen, die op zijn beurt de identiteit van de CA aantoont. Na ontvangst van het nieuwe certificaat schrapt de cliënt onmiddellijk het huidige certificaat en vervangt het door het nieuwe, waarvan de geldigheid onmiddellijk begint.

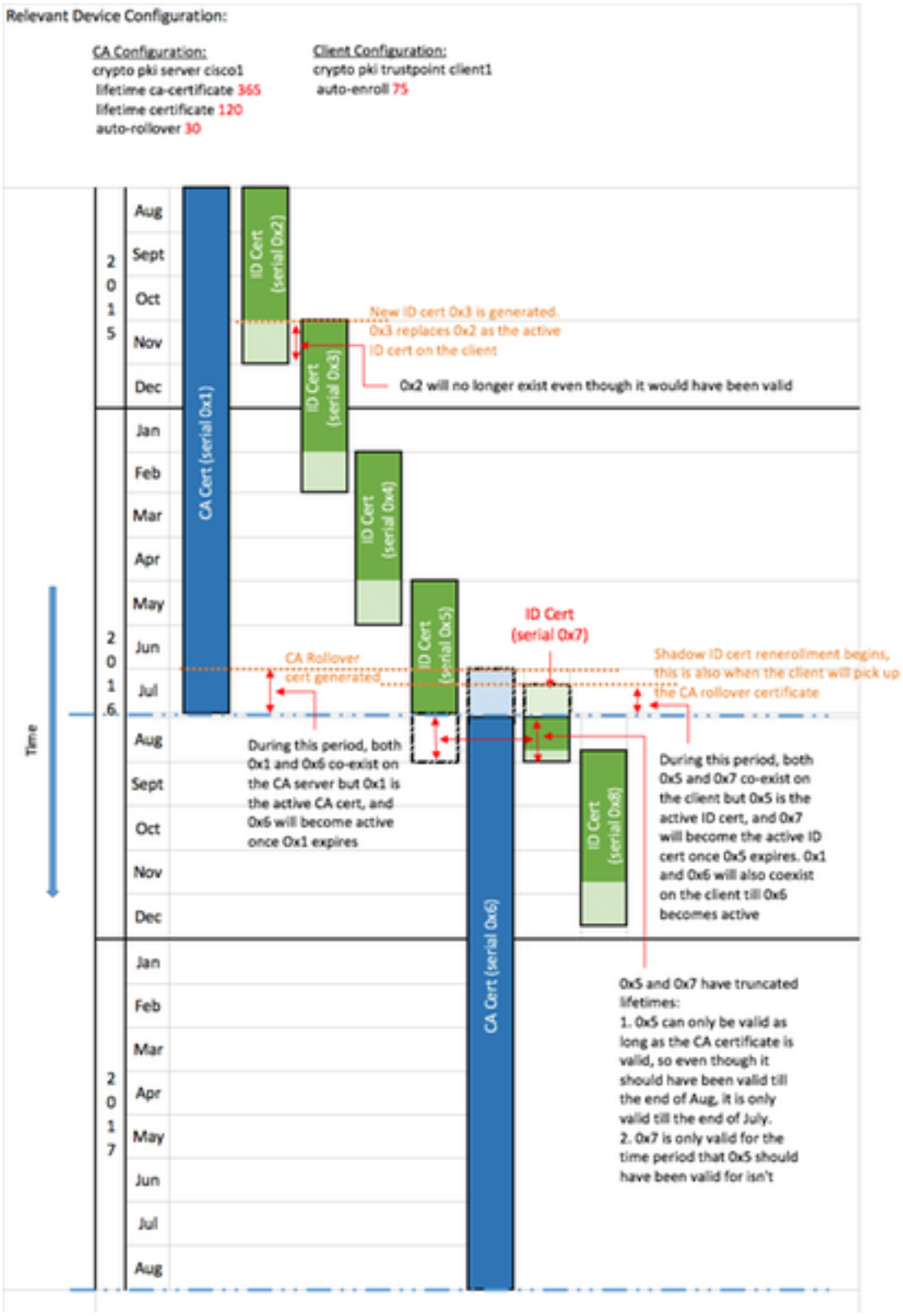
## Rollover

Rollover is een speciaal geval waarin het CA-certificaat afloopt en een nieuw CA-certificaat wordt gegenereerd. De CA genereert een nieuw CA-certificaat dat geldig wordt zodra het huidige CA-certificaat verstrijkt. CA genereert gewoonlijk dit "Shadow CA"-certificaat enige tijd voor de rollover-tijd, omdat dit nodig is om "Shadow ID"-certificaten voor de klanten te genereren.

Wanneer het certificaat van de SCEP client verloopdatum nadert, vraagt de SCEP client de CA voor het "Schaduw CA"-certificaat. Dit gebeurt met de werking **GetNextCACert** zoals hier wordt getoond:

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

Zodra de SCEP-client het "Shadow CA"-certificaat heeft, vraagt hij om een "Shadow ID"-certificaat na de normale inschrijving procedure. De CA tekent het "Schaduw-ID"-certificaat met het "Schaduw-CA"-certificaat. Anders dan een normaal hervernieuwingsverzoek wordt het "Schaduw-ID"-certificaat dat wordt teruggegeven, geldig op het moment dat de CA-certificaat is verlopen (rollover). Daarom moet de klant een kopie van de voor- en na-kantelcertificaten bewaren voor zowel het CA- als het ID-certificaat. Op het moment van CA-verloopdatum (rollover) verwijdert de SCEP-client het huidige CA-certificaat en de ID-certificering en vervangt deze door de "schaduwexemplaren".




# Bouwstenen

Deze structuur wordt gebruikt als bouwstenen van het SCEP.

> Opmerking: PKCS#7 en PKCS#10 zijn geen SCEP-specifiek.

## PKCS#7

PKCS#7 is een gedefinieerd gegevensformaat waarin gegevens kunnen worden getekend of versleuteld. Het gegevensformaat omvat de oorspronkelijke gegevens en de bijbehorende metagegevens die nodig zijn voor de uitvoering van de cryptografische handeling.

### Signed Envelope (SignedData)

De ondertekende enveloppe is een formaat dat gegevens draagt en bevestigt dat de ingekapselde gegevens niet worden gewijzigd in doorvoer via digitale handtekeningen. Het bevat deze informatie:

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

- Versienummer - Met SCEP, versie 1 gebruikt.
- Lijst met gebruikte algoritmen - Met SCEP is er slechts één handtekening en dus slechts één Hashing-algoritme.
- Feitelijke gegevens die worden ondertekend - Met SCEP is dit een PKCS#7 Enveloped-data-formaat (Encrypted Envelope).
- Lijst met certificaten van de signaleurs - Met SCEP is dit een zelfgetekend certificaat bij eerste inschrijving of het huidige certificaat bij herinschrijving.
- Een lijst van de signalers en de vingerafdruk die door elke ondertekenaar wordt gegenereerd - met het SCEP is er slechts één handtekening.

De ingekapselde gegevens worden niet versleuteld of verduisterd. Deze bestandsindeling biedt bescherming tegen het gewijzigde bericht.

### Geëvelde gegevens (EnvelopedData)

Het uitgebreide gegevensformaat bevat gecodeerde gegevens die alleen door de gespecificeerde ontvanger(s) kunnen worden gedecrypteerd. Het bevat deze informatie:

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

- Versienummer - Met SCEP wordt versie 0 gebruikt.
- Lijst van elk van de ontvangers en de hiermee samenhangende versleutelde gegevenscoderingssleutel - Met SCEP is er slechts één ontvanger (voor verzoeken: de CA-server; voor antwoorden : de cliënt).
- De gecodeerde gegevens - Dit is versleuteld met een willekeurige sleutel (die is versleuteld met de openbare sleutel van de ontvanger).

## PKCS#10

PKCS#10 beschrijft het formaat van een CSR. Een CSR bevat de informatie die klanten vragen in

hun certificaten op te nemen:

- Naam van het onderwerp
- Een kopie van de openbare sleutel
- Een challenge-wachtwoord (optioneel)
- Aangevraagde verlenging van de certificaten, zoals: Toetsengebruik (KU)Uitgebreid gebruik (EKU)Onderwerp Alternatieve naam (SAN)Universele hoofdnaam (UPN)
- Een vingerafdruk van het verzoek

Hier is een voorbeeld van een MVO:

```
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
```

# Gerelateerde informatie

- SCEP IETF-ontwerp
- Verouderde SCEP met behulp van de CLI-configuratiegids
- SCEP-ondersteuning voor BYOD configureren

# Bijlage

## SCEP-aanvragen

### Formaat bericht aanvragen

De aanvragen worden verzonden met een HTTP-KRIJG van het formulier:

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

Wanneer:

- **CGI-path** is afhankelijk van de server en verwijst naar het Common Gateway Interface (CGI) programma dat SCEP verzoeken behandelt: Cisco IOS® CA gebruikt een leeg pad string.Microsoft CA gebruikt **/certsrv/mscep/mscep.dll**, wat verwijst naar de MSCEP/Network Devices Encapsulation Service (NDES) ISIS-service.
- **Handeling** identificeert de handeling die wordt uitgevoerd.
- **Een bericht** bevat aanvullende gegevens voor die handeling (en kan leeg zijn indien er geen actuele gegevens vereist zijn).

Met de GET methode is het **bericht** onderdeel onbewerkte tekst of onderscheidde coderingsregels (DER)-gecodeerd PKCS#7 geconverteerd naar Base64. Als de POST-methode wordt ondersteund, kan inhoud die in Base64-codering met GET wordt verzonden in binaire indeling met POST worden verstuurd.

### Schematische weergave

Mogelijke waarden voor **operaties** en de bijbehorende **berichtwaarden**:

- **Bediening** = `PKIOperation`: **bericht**is een SCEP PkiBerichtstructuur, gebaseerd op PKCS#7 en gecodeerd met DER en Base64.de structuur van **de** pki - berichten kan van deze types zijn : **PKCSReq**: PKCS#10 CSR**GaCertInitiaal**: stemming over de toekenning van de status van MVO**GetCert** of **GetCRL**: certificaat of CRL-opname
- **operatie** = **GetCACert**, **GetNextCACert**, of (optioneel) **GetCACaps**: **bericht** kan worden weggelaten of wordt ingesteld op een naam die de CA identificeert.

## SCEP-antwoorden

### Format van antwoordbericht

SCEP responsen worden teruggegeven als standaard HTTP-inhoud, met een **Content-Type** dat afhankelijk is van het oorspronkelijke verzoek en het type teruggegeven gegevens. De inhoud van DER wordt teruggegeven als binair (niet in Base64 zoals voor het verzoek). inhoud PKCS#7 kan gecodeerde/ondertekende ingekapselde gegevens bevatten of kan niet bevatten; indien dit niet het geval is (slechts een reeks certificaten bevat), wordt het een **gedegenereerde** PKCS#7 genoemd.

### Inhoud

Mogelijke waarden voor **contenttype**:

**sollicitatie/x-pki-bericht:**

- in reactie op de **PKIO**-operatie, met **Pki**-bericht van het type: **PKCSReq, GetCertinitiële, GetCert of GetCRL**
- de responsinstantie is een **pogrom** van het type: **CertRep**

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

- in reactie op de werking van **GetCACert**
- responsinstantie is het DER-gecodeerde X.509 CA-certificaat

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

- in reactie op de werking van **GetCACert**
- responsinstantie is een DER-gecodeerde degenererende PKCS#7 die de CA- en RA-certificaten bevat

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

- als antwoord op de bediening **GetNextCAC**
- de responsinstantie is een variatie van een **pki-bericht** van het type: **CertRep**

## De PkiBerichtstructuur

### SCEP OID's

```
2.16.840.1.113733.1.9.2 scep-messageType
2.16.840.1.113733.1.9.3 scep-pkiStatus
2.16.840.1.113733.1.9.4 scep-failInfo
2.16.840.1.113733.1.9.5 scep-senderNonce
2.16.840.1.113733.1.9.6 scep-recipientNonce
2.16.840.1.113733.1.9.7 scep-transId
2.16.840.1.113733.1.9.8 scep-extensionReq
```

### SCEP-bericht

- **ondertekende** PKCS#7**-gegevens**
- PKCS#7 EnvelopedData (**pkcsPKIEnvelope** genoemd; optioneel, versleuteld naar bericht ontvanger)
  **berichtData** (CSR, cert, CRL, ...)
- **SignerInfo** met **echtheidskenmerken**:
  **transactieID**, **berichtType**, **zenderNoncepkiStatus**, **ontvangerNonce** (alleen respons)**faalinformatie** (alleen respons + falen)

### SCEP berichtType

- verzoek:
  **PKCSReq** (19): PKCS#10 CSR**GetCertInitial** (20): polis voor inschrijving van certificaten**GetCert** (21): herwinning van certificaten**GetCRL** (22): CRL-herkenning
- antwoord :
  **CertRep** (3): antwoord op een certificaat of een CRL-verzoek

### SCEP PKIstatus

- **SUCCES** (0): verzoek ingewilligd (antwoord in pkcsPKIEnvelope)
- **MISLUKKING** (2): verzoek verworpen (details in defaultInfo attribuut)
- **PENNEN** (3): verzoek wacht op handmatige goedkeuring