Présentation du protocole d'inscription de certificat simple

Contenu

[Introduction](#)
[Informations générales](#)
[Authentification CA](#)
[Demander](#)
[Réponse](#)
[Inscription client](#)
[Demander](#)
[Réponse](#)
[Réinscription du client](#)
[Renouvellement](#)
[Incrustation](#)
[Blocs de construction](#)
[PKCS n°7](#)
[Enveloppe signée (SignedData)](#)
[Données enveloppées (données enveloppées)](#)
[PKCS#10](#)
[Informations connexes](#)
[Annexe](#)
[Demandes SCEP](#)
[Format du message de demande](#)
[Vue Schéma](#)
[Réponses SCEP](#)
[Format du message de réponse](#)
[Types de contenu](#)
[Structure pkiMessage](#)
[OID SCEP](#)
[SCEP pkiMessage](#)
[SCEP messageType](#)
[SCEP pkiStatus](#)

Introduction

Ce document décrit le protocole SCEP (Simple Certificate Enrollment Protocol), qui est un protocole utilisé pour l'inscription et d'autres opérations PKI (Public Key Infrastructure).

Informations générales

Le SCEP a été développé à l'origine par Cisco et est documenté dans une ébauche de l'IETF

(Internet Engineering Task Force).

Ses principales caractéristiques sont les suivantes :

- Modèle de requête/réponse basé sur HTTP (méthode GET); prise en charge facultative de la méthode POST)
- Prend uniquement en charge la cryptographie basée sur RSA
- Utilise PKCS#10 comme format de demande de certificat
- Utilise PKCS#7 afin de transmettre des messages cryptés/signés cryptographiquement
- Prend en charge l'octroi asynchrone par le serveur, avec interrogation régulière par le demandeur
- Prise en charge limitée de la récupération des listes de révocation de certificats (CRL) (la méthode préférée est une requête CDP (CRL Distribution Point), pour des raisons d'évolutivité)
- Ne prend pas en charge la révocation de certificats en ligne (doit être effectuée hors connexion par d'autres moyens)
- Nécessite l'utilisation d'un champ de **mot de passe de confirmation** dans la demande de signature de certificat (CSR), qui doit être partagé uniquement entre le serveur et le demandeur

L'inscription et l'utilisation du SCEP suivent généralement ce processus :

1. Obtenir une copie du certificat de l'autorité de certification (AC) et le valider.
2. Générez une CSR et envoyez-la en toute sécurité à l’AC.
3. Interrogez le serveur SCEP afin de vérifier si le certificat a été signé.
4. S'inscrire à nouveau si nécessaire afin d'obtenir un nouveau certificat avant l'expiration du certificat actuel.
5. Récupérez la liste de révocation de certificats si nécessaire.

Authentification CA

Le SCEP utilise le certificat d'autorité de certification afin de sécuriser l'échange de messages pour le CSR. Par conséquent, il est nécessaire d'obtenir une copie du certificat d'AC. L'opération **GetCACert** est utilisée.

Demander

La requête est envoyée en tant que requête HTTP GET. Une capture de paquets pour la requête ressemble à ceci :

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

Réponse

La réponse est simplement le certificat CA codé en binaire (X.509). Le client doit valider que le certificat de l'autorité de certification est approuvé par un examen de l'empreinte digitale/hachage. Pour ce faire, vous devez utiliser une méthode hors bande (appel téléphonique à un administrateur système ou préconfiguration de l'empreinte digitale au sein du point de confiance).

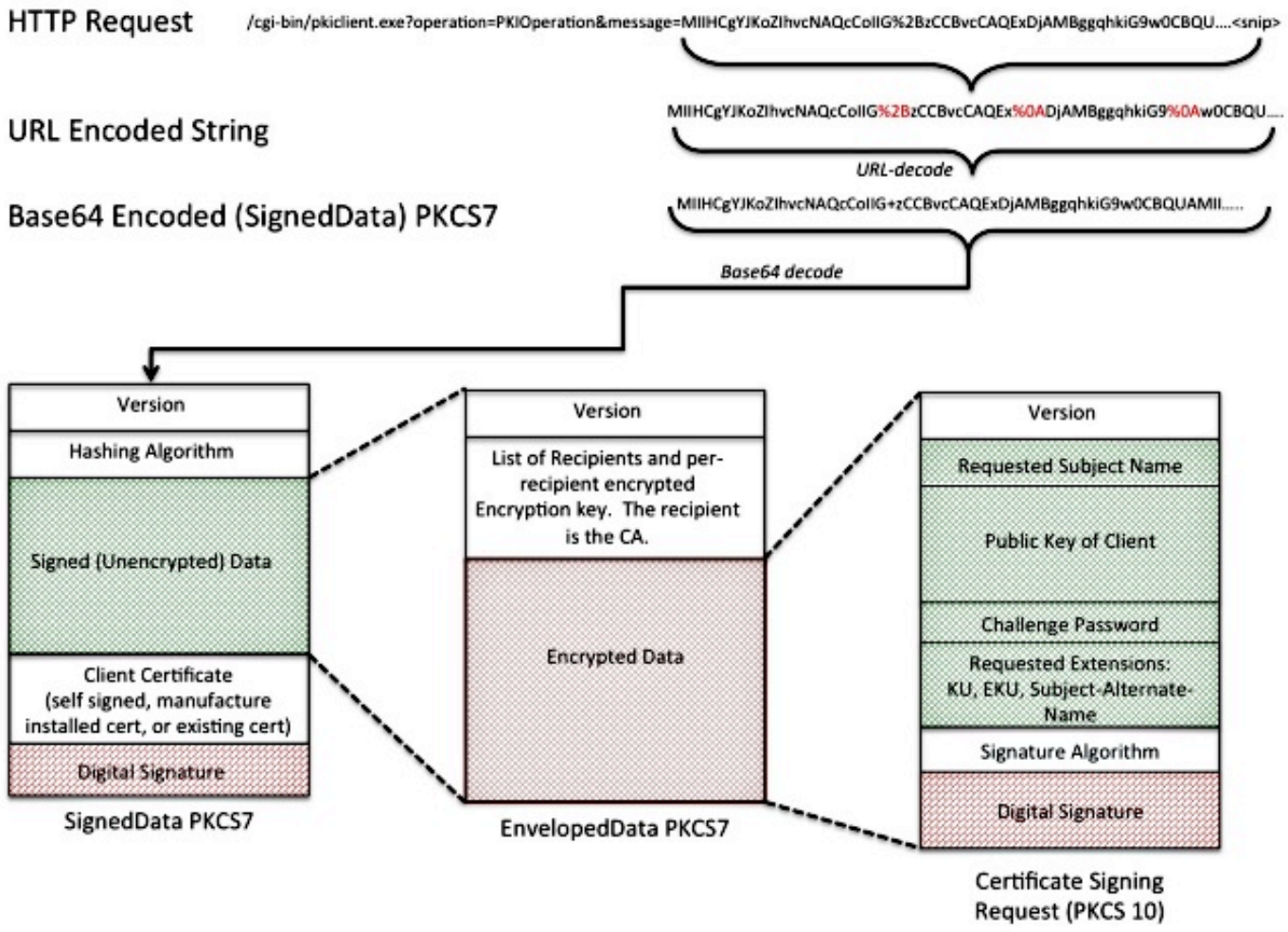
Inscription client

Demander

La demande d'inscription est envoyée en tant que requête HTTP GET. Une capture de paquets pour la requête ressemble à ceci :

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

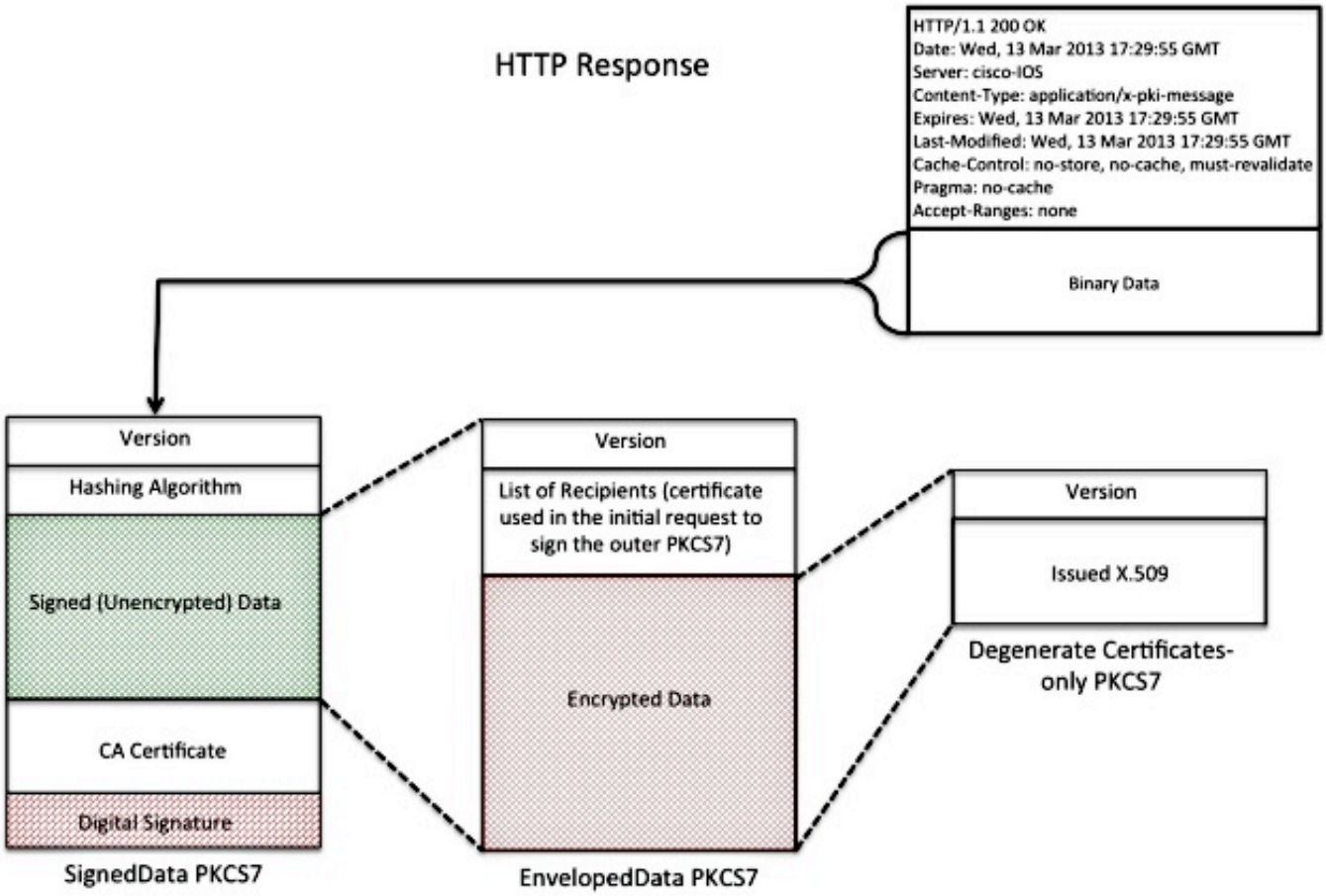
1. Le texte qui suit le « message=" est une chaîne encodée URL, qui est extraite de la chaîne de requête GET.
2. Le texte est ensuite URL décodé en chaîne de texte ASCII. Cette chaîne de texte est une clé PKCS#7 SignedData codée en Base64.
3. SignedData PKCS#7 est signé par le client avec l'un de ces certificats ; il sert à prouver que le client l'a envoyé et qu'il n'a pas été modifié en transit :
   Certificat auto-signé (utilisé lors de l'inscription initiale)Certificat installé par le fabricant (MIC)Une certification actuelle qui expire bientôt (réinscription)
4. La partie « Données signées » du PKCS 7 SignedData est une PKCS 7 EnvelopedData.
5. Le PKCS#7 EnvelopedData est un conteneur qui contient « Données chiffrées » et la « clé de déchiffrement ». La clé de déchiffrement est chiffrée avec la clé publique du destinataire. Dans ce cas précis, le destinataire est l'autorité de certification ; en conséquence. Seule l'autorité de certification peut réellement déchiffrer les données chiffrées.
6. La partie « Données chiffrées » du PKCS#7 est la CSR (PKCS#10).



Réponse

La réponse à la demande d'inscription SCEP est de trois types :

- **Rejeter** - La demande est rejetée par l'administrateur pour un certain nombre de raisons, telles que :
  Taille de clé non valideMot de passe de défi non valideL'autorité de certification n'a pas pu valider la demandeLa demande demandait des attributs que l'AC n'a pas autorisésLa demande a été signée par une identité en laquelle l'autorité de certification ne fait pas confiance
- **En attente** - L'administrateur de l'autorité de certification n'a pas encore examiné la demande.
- **Succès** - La demande est acceptée et le certificat signé est inclus. Le certificat signé est détenu dans un type spécial de PKCS#7 appelé « Certificats dégénérés-PKCS#7 uniquement », qui est un conteneur spécial qui peut contenir un ou plusieurs X.509 ou CRL, mais ne contient pas de données utiles signées ou chiffrées.



Réinscription du client

Avant l'expiration du certificat, le client doit obtenir un nouveau certificat. Il y a une légère différence de comportement entre le renouvellement et le roulement. Le renouvellement se produit lorsque le certificat d'ID du client approche l'expiration et que sa date d'expiration n'est pas identique (antérieure) à la date d'expiration du certificat de l'autorité de certification. Le survol se produit lorsque le certificat d'ID approche de l'expiration et que sa date d'expiration est identique à la date d'expiration du certificat de l'Autorité de certification.

Renouvellement

Àl'approche de la date d'expiration d'un certificat d'ID, un client SCEP peut vouloir obtenir un nouveau certificat. Le client génère un CSR et passe par le processus d'inscription (tel que défini précédemment). Le certificat actuel est utilisé afin de signer le PKCS 7 SignedData, qui à son tour prouve l'identité de l'autorité de certification. Dès réception du nouveau certificat, le client supprime immédiatement le certificat actuel et le remplace par le nouveau, dont la validité commence immédiatement.

Incrustation

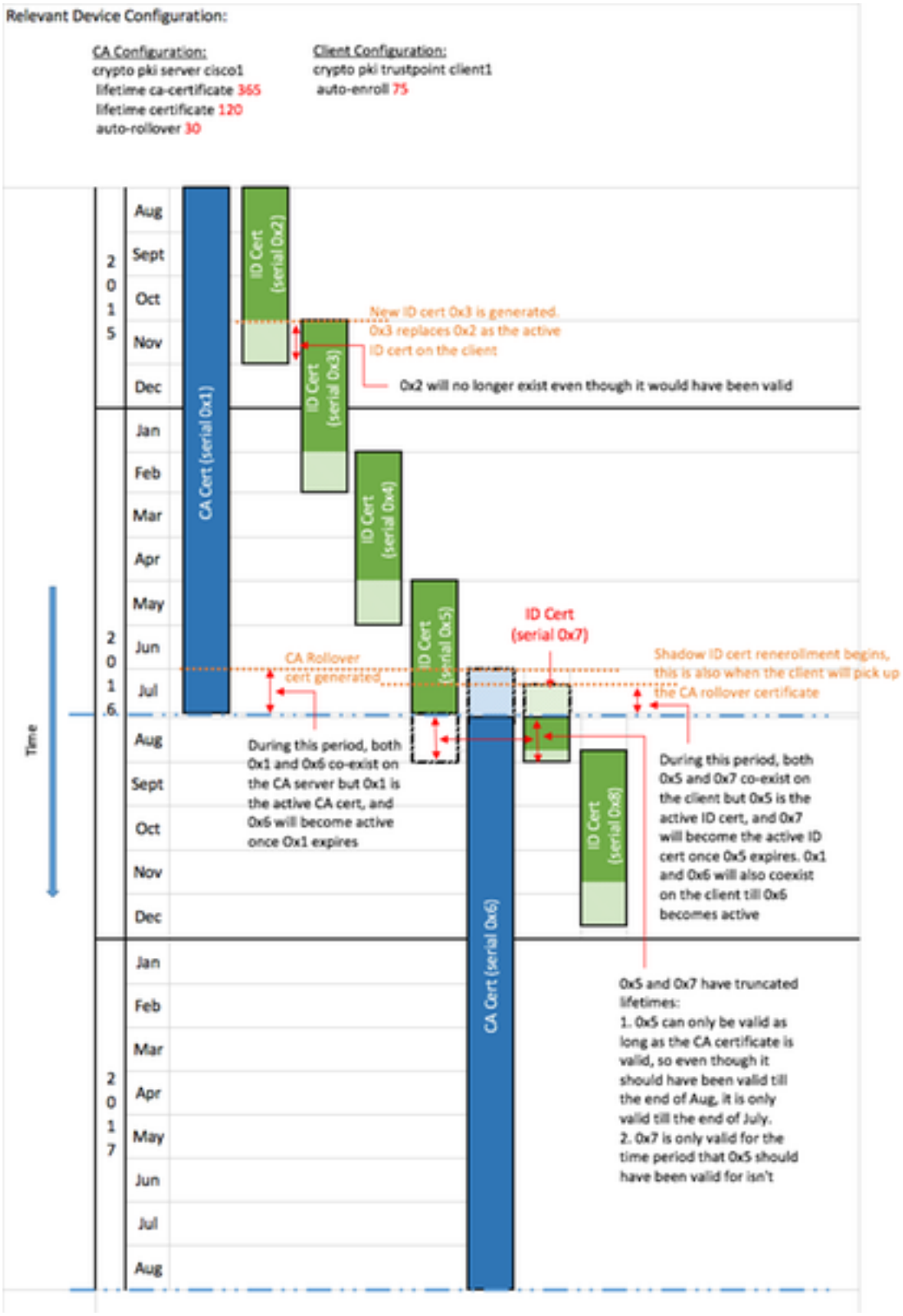
Le renvoi est un cas particulier où le certificat d'autorité de certification expire et un nouveau certificat d'autorité de certification est généré. L'autorité de certification génère un nouveau certificat d'autorité de certification qui devient valide une fois que le certificat d'autorité de

certification actuel expire. L'autorité de certification génère généralement ce certificat d'« autorité de certification fantôme » un certain temps avant le temps de transfert, car il est nécessaire pour générer des certificats d'« identification fantôme » pour les clients.

Lorsque le certificat d'ID du client SCEP approche de l'expiration, le client SCEP demande à l'autorité de certification le certificat d'« autorité de certification fantôme ». Cela se fait avec l'opération **GetNextCACert** comme indiqué ici :

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

Une fois que le client SCEP a le certificat « Shadow CA », il demande un certificat « Shadow ID » après la procédure d'inscription normale. L'autorité de certification signe le certificat « Shadow ID » avec le certificat « Shadow CA ». Contrairement à une demande de renouvellement normale, le certificat « ID d'ombre » retourné devient valide au moment de l'expiration du certificat de l'autorité de certification (inversement). Par conséquent, le client doit conserver une copie des certificats pré- et post-renversement pour l'AC et le certificat d'ID. Au moment de l'expiration de l'autorité de certification (inversement), le client SCEP supprime le certificat et l'ID de l'autorité de certification actuels et les remplace par les copies « fantômes ».



Blocs de construction

Cette structure est utilisée comme éléments de base du SCEP.

Note: PKCS#7 et PKCS#10 ne sont pas spécifiques à SCEP.

PKCS n°7

PKCS#7 est un format de données défini qui permet de signer ou de chiffrer des données. Le format des données inclut les données d'origine et les métadonnées associées nécessaires pour effectuer l'opération de cryptographie.

Enveloppe signée (SignedData)

L'enveloppe signée est un format qui transporte les données et confirme que les données encapsulées ne sont pas modifiées en transit par le biais de signatures numériques. Elle comprend ces informations :

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

- Numéro de version : avec SCEP, version 1 utilisée.
- Liste des algorithmes Digest utilisés - Avec SCEP, il n'y a qu'un seul signataire et donc un seul algorithme de hachage.
- Données réelles signées - Avec SCEP, il s'agit d'un format de données enveloppées PKCS#7 (Enveloped Envelope).
- Liste des certificats des signataires - Avec SCEP, il s'agit d'un certificat auto-signé lors de l'inscription initiale ou du certificat actuel si vous vous réinscrivez.
- Liste des signataires et empreintes digitales générées par chaque signataire - Avec SCEP, il n'y a qu'un seul signataire.

Les données encapsulées ne sont ni chiffrées ni masquées. Ce format offre simplement une protection contre le message modifié.

Données enveloppées (données enveloppées)

Le format Données enveloppées transporte des données chiffrées et ne peuvent être décryptées que par le ou les destinataires spécifiés. Elle comprend ces informations :

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

- Numéro de version : avec SCEP, la version 0 est utilisée.
- Liste de chacun des destinataires et clé de cryptage des données cryptées associée : avec SCEP, il n'y a qu'un seul destinataire (pour les demandes : le serveur AC ; pour les réponses : le client).
- Les données chiffrées : elles sont chiffrées à l'aide d'une clé générée aléatoirement (chiffrée à l'aide de la clé publique du destinataire).

PKCS#10

PKCS#10 décrit le format d'une CSR. Une RSE contient les informations que les clients

demandent à être incluses dans leurs certificats :

- Nom du sujet
- Une copie de la clé publique
- Mot de passe de confirmation (facultatif)
- Toutes les extensions de certificat requises, telles que :
  Utilisation des clés (KU)Utilisation de clé étendue (EKU)Nom alternatif du sujet (SAN)Nom principal universel (UPN)
- Une empreinte de la demande

Voici un exemple de RSE :

```
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
```

Informations connexes

- Projet SCEP IETF
- SCEP hérité à l'aide du guide de configuration CLI
- Configuration de la prise en charge SCEP pour le BYOD

Annexe

Demandes SCEP

Format du message de demande

Les requêtes sont envoyées avec un HTTP GET du formulaire :

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

Where:

- **CGI-path** dépend du serveur et pointe vers le programme CGI (Common Gateway Interface) qui gère les requêtes SCEP : Cisco IOS® CA utilise une chaîne de chemin vide.Microsoft CA utilise **/certsrv/mscep/mscep.dll**, qui pointe vers le service IIS MSCEP/NDES (Network Device Enrollment Service).
- **Opération** identifie l'opération qui est effectuée.
- **Le message** transporte des données supplémentaires pour cette opération (et il peut être vide si aucune donnée réelle n'est requise).

Avec la méthode GET, la partie **du message** est soit du texte brut, soit du PKCS#7 codé par des règles de codage différencié (DER) converti en Base64. Si la méthode POST est prise en charge, le contenu qui serait envoyé en codage Base64 avec GET peut être envoyé au format binaire avec POST à la place.

Vue Schéma

Valeurs possibles pour **les opérations** et leurs valeurs **de message** associées :

- **opération =** `opération PKIO` : **message**est une structure SCEP **pkiMessage**, basée sur PKCS#7 et codée avec DER et Base64.la structure **pkiMessage** peut être de ces types : **PKCSReq** : PKCS#10 CSR**GetCertInitial** : interrogation du statut d'octroi CSR**GetCert** ou **GetCRL** : récupération de certificat ou de liste de révocation de certificats
- **opération = GetCACert**, **GetNextCACert** ou (facultatif) **GetCACaps** : **message** peut être omis ou défini sur un nom qui identifie l'autorité de certification.

Réponses SCEP

Format du message de réponse

Les réponses SCEP sont retournées sous forme de contenu HTTP standard, avec un **type de contenu** qui dépend de la requête d'origine et du type de données retournées. Le contenu DER est retourné sous forme binaire (pas dans Base64 comme pour la requête). Le contenu PKCS#7 peut contenir ou non des données enveloppées cryptées/signées ; s'il ne contient pas (contient uniquement un ensemble de certificats), il est appelé PKCS#7 **dégénéré**.

Types de contenu

Valeurs possibles pour **Content-Type** :

application/x-pki-message :

- en réponse à l'opération **PKIOopération**, avec **pkiMessage** de type : **PKCSReq, GetCertInitial,**

GetCert ou GetCRL
- le corps de la réponse est un **pkiMessage** de type : **CertRep**

application/x-x509-ca-cert :

- en réponse à l'opération **GetCACert**
- Le corps de réponse est le certificat CA X.509 codé DER

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

- en réponse à l'opération **GetCACert**
- Le corps de réponse est un PKCS#7 dégénéré codé en DER qui contient les certificats CA et RA

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

- en réponse à l'opération **GetNextCACert**
- le corps de la réponse est une variante d'un **pkiMessage** de type : **CertRep**

Structure pkiMessage

OID SCEP

```
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 pkiMessage

- PKCS#7 **SignedData**
- PKCS#7 EnvelopedData (appelé **pkcsPKIEnvelope** ; facultatif, chiffré au destinataire du message)
  messageData (CSR, cert, CRL, ...)
- **SignerInfo** avec **authenticatedAttributes** :
  transactionID, **messageType**, **senderNoncepkiStatus**, **destinataireNonce** (réponse uniquement)**failInfo** (réponse + échec uniquement)

SCEP messageType

- demande :
  PKCSReq (19) : PKCS#10 CSR**GetCertInitial** (20) : interrogation d'inscription de certificat**GetCert** (21) : récupération de certificat**GetCRL** (22) : Récupération CRL
- réponse :
  CertRep (3) : réponse à une demande de certificat ou de liste de révocation de certificats

SCEP pkiStatus

- **SUCCÈS** (0) : demande accordée (réponse dans pkcsPKIEnvelope)
- **ÉCHEC** (2) : demande rejetée (détails dans l'attribut failInfo)
- **EN ATTENTE** (3) : demande en attente d'approbation manuelle