Übersicht über das Simple Certificate Enrollment Protocol

Inhalt

Einführung

Dieses Dokument beschreibt das Simple Certificate Enrollment Protocol (SCEP), ein Protokoll für die Registrierung und andere PKI-Operationen (Public Key Infrastructure).

Hintergrundinformationen

SCEP wurde ursprünglich von Cisco entwickelt und ist in einem IETF-Entwurf (Internet

Engineering Task Force) dokumentiert.

Seine wichtigsten Merkmale sind:

- Anforderungs-/Antwortmodell basierend auf HTTP (GET-Methode) optionale Unterstützung für POST-Methode)
- Unterstützt nur RSA-basierte Verschlüsselung
- Verwendet PKCS#10 als Zertifikatsanforderungsformat
- Verwendet PKCS#7, um kryptografisch signierte/verschlüsselte Nachrichten zu übertragen
- Unterstützt die asynchrone Bereitstellung durch den Server, wobei der Anforderer regelmäßig fragt
- Unterstützung für eingeschränkten CRL-Abruf (Certificate Revocation List) (die bevorzugte Methode ist aus Gründen der Skalierbarkeit über eine CRL Distribution Point (CDP)-Abfrage)
- Keine Unterstützung für Online-Zertifikat-Widerruf (muss offline über andere Mittel erfolgen)
- Erfordert die Verwendung eines **Challenge-Kennwortfelds** im CSR (Certificate Signing Request), das nur vom Server und vom Anforderer gemeinsam genutzt werden darf

Die Registrierung und Nutzung von SCEP erfolgt in der Regel wie folgt:

1. Eine Kopie des Zertifikats der Zertifizierungsstelle (Certificate Authority, CA) einholen und validieren.
2. Erstellen Sie einen CSR, und senden Sie ihn sicher an die CA.
3. Überprüfen Sie auf dem SCEP-Server, ob das Zertifikat signiert wurde.
4. Registrieren Sie sich nach Bedarf erneut, um vor Ablauf des aktuellen Zertifikats ein neues Zertifikat zu erhalten.
5. Rufen Sie das CRL nach Bedarf ab.

CA-Authentifizierung

SCEP verwendet das Zertifizierungsstellenzertifikat, um den Nachrichtenaustausch für den CSR zu sichern. Daher ist es erforderlich, eine Kopie des Zertifizierungsstellenzertifikats zu erhalten. Der **GetCACert-**Vorgang wird verwendet.

Anfrage

Die Anforderung wird als HTTP GET-Anforderung gesendet. Eine Paketerfassung für die Anfrage sieht ähnlich aus wie folgt:

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

Antwort

Die Antwort ist einfach das binär kodierte CA-Zertifikat (X.509). Der Kunde muss durch eine Prüfung des Fingerabdrucks/Hashs überprüfen, ob das Zertifikat der Zertifizierungsstelle vertrauenswürdig ist. Dies muss über eine Out-of-Band-Methode erfolgen (Telefonanruf bei einem Systemadministrator oder Vorkonfiguration des Fingerabdrucks innerhalb des Trustpoints).

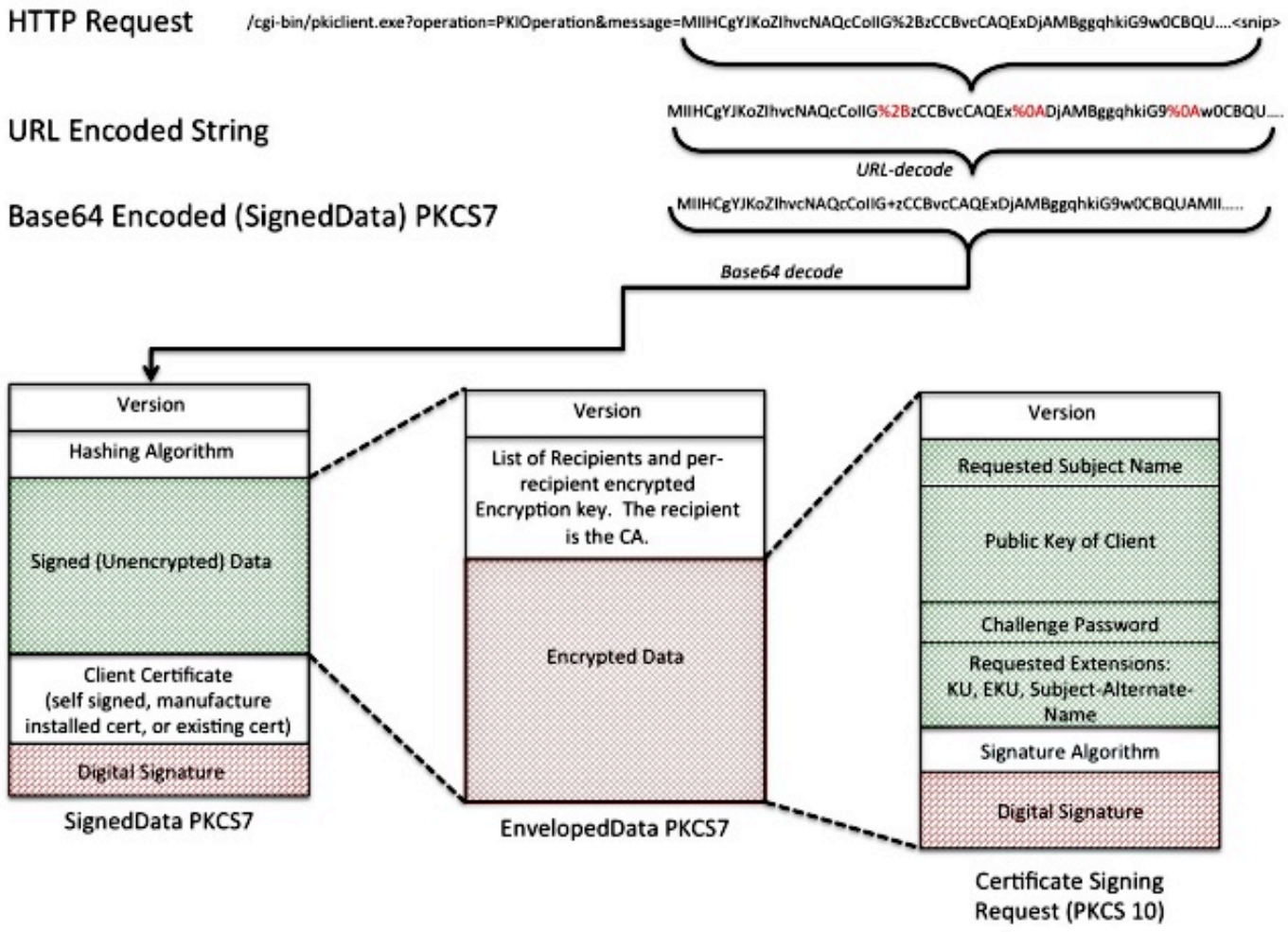
Client-Registrierung

Anfrage

Die Registrierungsanfrage wird als HTTP GET-Anforderung gesendet. Eine Paketerfassung für die Anfrage sieht ähnlich aus wie folgt:

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

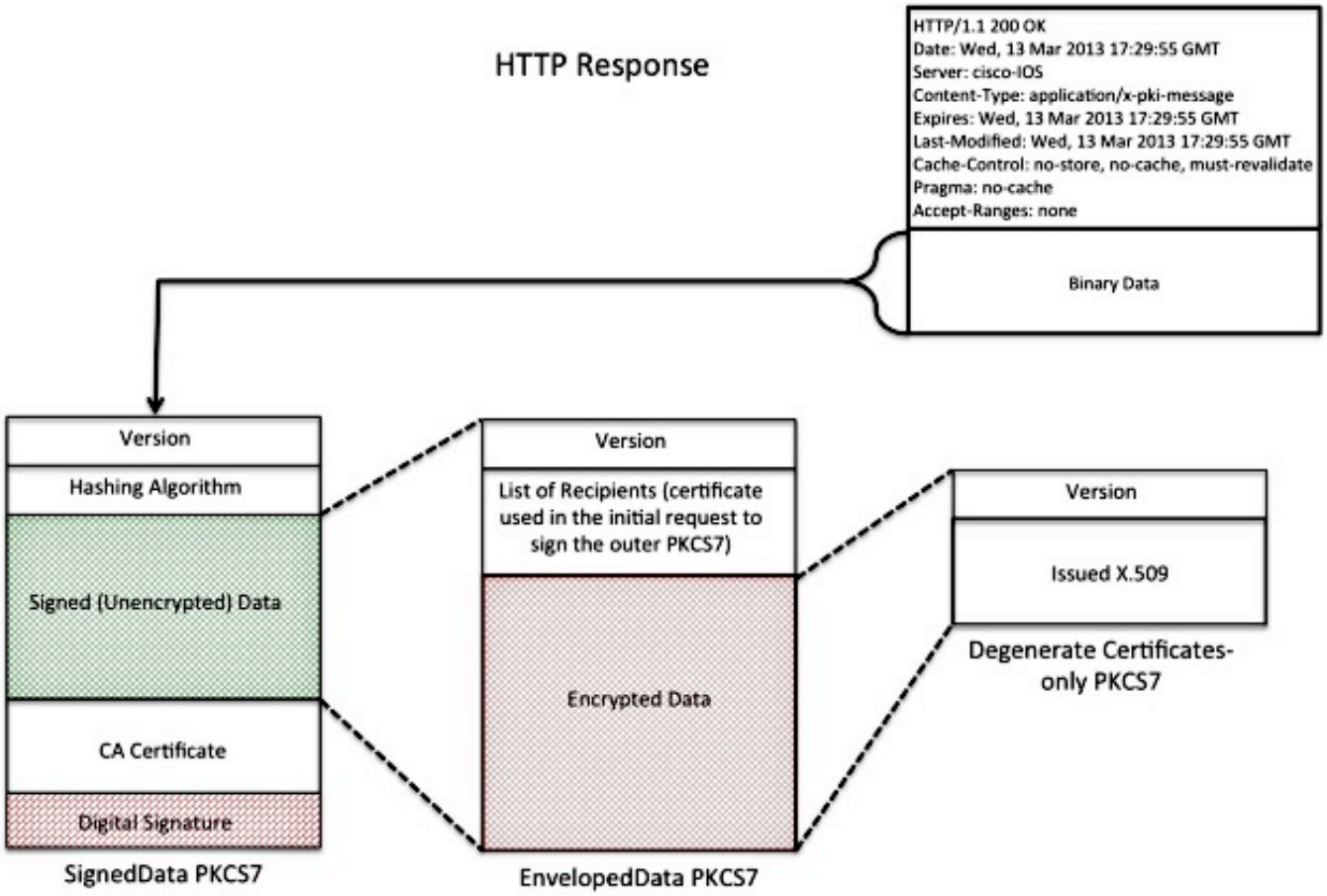
1. Der Text nach "message=" ist eine URL-kodierte Zeichenfolge, die aus der GET-Anforderungszeichenfolge extrahiert wird.
2. Der Text wird dann als URL in eine ASCII-Textzeichenfolge decodiert. Diese Textzeichenfolge ist eine Base64-codierte SignedData PKCS#7.
3. Die SignedData PKCS#7 wird vom Client mit einem dieser Zertifikate signiert. Sie wird verwendet, um nachzuweisen, dass der Kunde sie gesendet hat und dass sie bei der Übertragung nicht verändert wurde:
   Ein selbstsigniertes Zertifikat (wird bei der Erstregistrierung verwendet)Ein vom Hersteller installiertes Zertifikat (MIC)Eine aktuelle Zertifizierung, die bald abläuft (erneute Registrierung)
4. Der Teil "Signed Data" der SignedData PKCS#7 ist eine EnvelopedData PKCS#7.
5. Die EnvelopedData PKCS#7 ist ein Container, der "Encrypted Data" und den "Entschlüsselungsschlüssel" enthält. Der Entschlüsselungsschlüssel wird mit dem öffentlichen Schlüssel des Empfängers verschlüsselt. In diesem speziellen Fall ist der Empfänger die CA. als Ergebnis. Nur die CA kann die "verschlüsselten Daten" tatsächlich entschlüsseln.
6. Der Teil "Encrypted Data" (Verschlüsselte Daten) der umgeschlossenen PKCS#7 ist der CSR (PKCS#10).



Antwort

Die Antwort auf die SCEP-Registrierungsanfrage ist einer von drei Typen:

- **Ablehnen** - Der Antrag wird vom Administrator aus folgenden Gründen abgelehnt: Ungültige SchlüssellängeUngültiges Kennwort für die AbfrageDie CA konnte die Anforderung nicht validieren.In der Anforderung wurden Attribute angefordert, die von der CA nicht autorisiert wurden.Die Anforderung wurde von einer Identität signiert, der die CA nicht traut
- **Ausstehend** - Der CA-Administrator hat die Anfrage noch nicht überprüft.
- **Erfolg**: Die Anfrage wird angenommen, und das signierte Zertifikat ist enthalten. Das signierte Zertifikat befindet sich in einem speziellen PKCS#7-Typ, der als "Degenerate Certificates-Only PKCS#7" bezeichnet wird. Hierbei handelt es sich um einen speziellen Container, der ein oder mehrere X.509- oder CRLs enthalten kann, jedoch keine signierte oder verschlüsselte Datennutzlast enthält.



Client-erneute Registrierung

Vor Ablauf des Zertifikats muss der Kunde ein neues Zertifikat erhalten. Zwischen Verlängerung und Rollover besteht ein leichter Verhaltensunterschied. Die Verlängerung erfolgt, wenn das ID-Zertifikat des Clients vor dem Ablauf steht und das Ablaufdatum nicht dasselbe ist (früher als) wie das Ablaufdatum des Zertifizierungsstellenzertifikats. Der Rollover erfolgt, wenn das ID-Zertifikat vor Ablauf steht und sein Ablaufdatum mit dem Ablaufdatum des Zertifikats der Zertifizierungsstelle identisch ist.

Verlängerung

Wenn das Ablaufdatum eines ID-Zertifikats näher rückt, kann ein SCEP-Client ein neues Zertifikat anfordern. Der Client generiert einen CSR und durchläuft den Registrierungsprozess (wie zuvor definiert). Das aktuelle Zertifikat wird zum Signieren der SignedData PKCS#7 verwendet, die wiederum der CA Identität nachweist. Nach Erhalt des neuen Zertifikats löscht der Kunde sofort das aktuelle Zertifikat und ersetzt es durch das neue Zertifikat, dessen Gültigkeit sofort beginnt.

Rollover

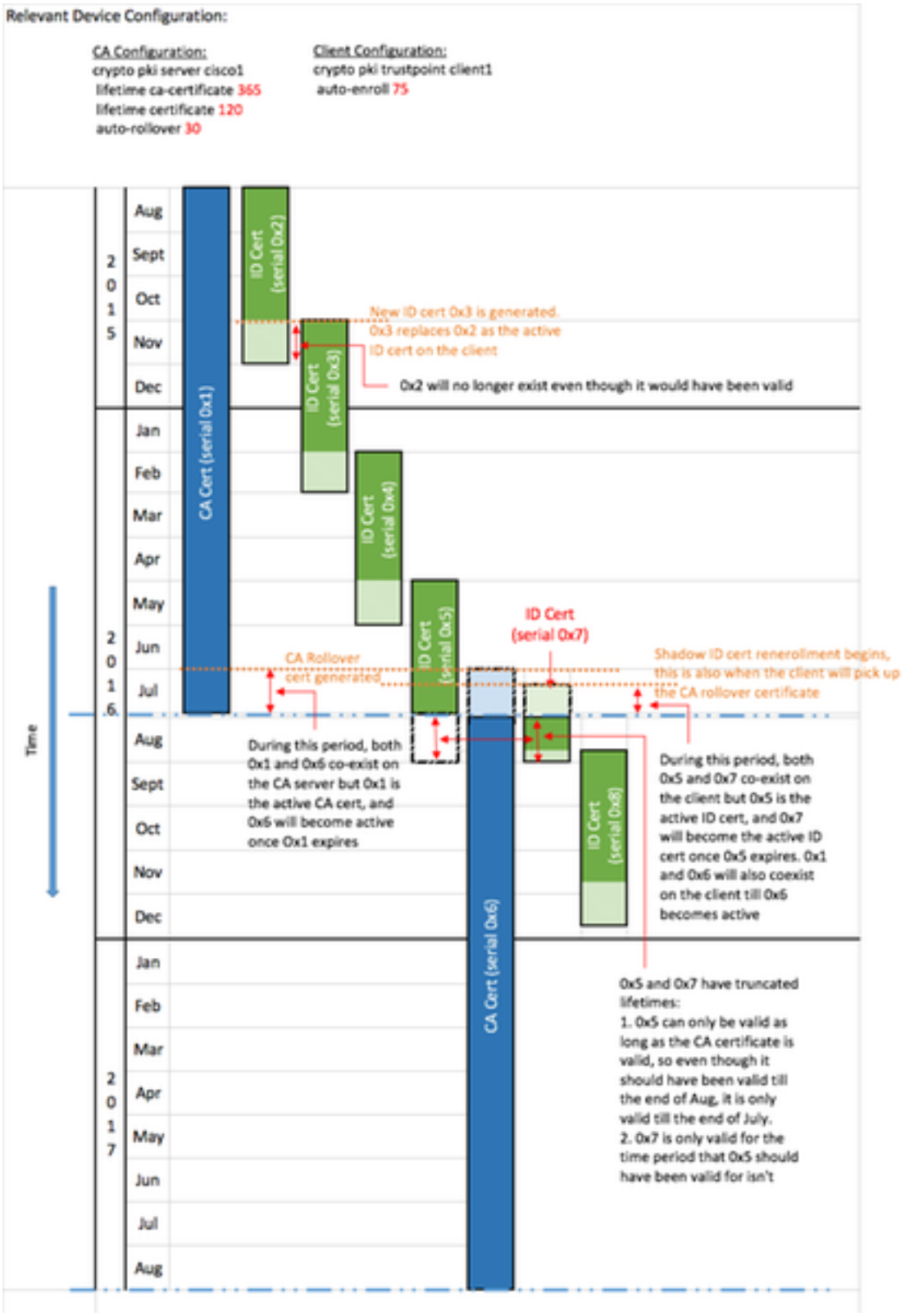
Rollover ist ein Sonderfall, bei dem das Zertifizierungsstellenzertifikat abläuft und ein neues Zertifizierungsstellenzertifikat generiert wird. Die Zertifizierungsstelle generiert ein neues Zertifizierungsstellenzertifikat, das nach Ablauf des aktuellen Zertifizierungsstellenzertifikats gültig wird. Die CA generiert dieses Zertifikat in der Regel einige Zeit vor der Rollover-Zeit, da es zum

Generieren von "Shadow-ID"-Zertifikaten für die Clients benötigt wird.

Wenn das ID-Zertifikat des SCEP-Clients vor Ablauf steht, fragt der SCEP-Client die CA nach dem Zertifikat der "Shadow CA" ab. Dies erfolgt mit der **GetNextCACert-**Operation, wie hier gezeigt:

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

Wenn der SCEP-Client über das Zertifikat "Shadow CA" verfügt, fordert er nach dem normalen Anmeldeverfahren ein "Shadow ID"-Zertifikat an. Die CA signiert das Zertifikat "Shadow ID" mit dem Zertifikat "Shadow CA". Im Gegensatz zu einer normalen Verlängerungsanfrage wird das zurückgegebene "Shadow ID"-Zertifikat zum Zeitpunkt des Ablaufs des Zertifizierungsstellenzertifikats (Rollover) gültig. Aus diesem Grund muss der Kunde eine Kopie der vor und nach dem Rollover gültigen Zertifikate sowohl für die Zertifizierungsstelle als auch für das ID-Zertifikat aufbewahren. Zum Zeitpunkt des Ablaufs der Zertifizierungsstelle (Rollover) löscht der SCEP-Client das aktuelle Zertifizierungsstellenzertifikat und das aktuelle Zertifizierungsstellen-ID-Zertifikat und ersetzt diese durch die "Shadow"-Kopien.



Bausteine

Diese Struktur wird als Bausteine von SCEP verwendet.

Hinweis: PKCS#7 und PKCS#10 sind nicht SCEP-spezifisch.

PKCS7

PKCS#7 ist ein definiertes Datenformat, mit dem Daten signiert oder verschlüsselt werden können. Das Datenformat enthält die Originaldaten und die zugehörigen Metadaten, die für die Ausführung des kryptografischen Vorgangs erforderlich sind.

Signed Envelope (SignedData)

Der signierte Umschlag ist ein Format, das Daten enthält und bestätigt, dass die gekapselten Daten bei der Übertragung über digitale Signaturen nicht verändert werden. Dazu gehören folgende Informationen:

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

- Versionsnummer - Bei SCEP wird Version 1 verwendet.
- Liste der verwendeten Digest-Algorithmen - Bei SCEP gibt es nur einen Signer und somit nur einen Hashing-Algorithmus.
- Tatsächliche Daten, die signiert werden - Bei SCEP handelt es sich um ein PKCS#7-Enveloped-Data-Format (verschlüsselter Umschlag).
- Liste der Zertifikate der Signaturen - Bei SCEP handelt es sich um ein selbstsigniertes Zertifikat bei der Erstregistrierung oder das aktuelle Zertifikat, wenn Sie sich erneut anmelden.
- Liste der Signaturen und des von jedem Signierer generierten Fingerabdrucks - Mit SCEP gibt es nur einen Signierer.

Die gekapselten Daten werden nicht verschlüsselt oder verschleiert. Dieses Format bietet lediglich Schutz gegen die Nachricht, die geändert wird.

Umgeleitete Daten (EnvelopedData)

Das Format Umgeleiteter Daten enthält verschlüsselte Daten, die nur von den angegebenen Empfängern entschlüsselt werden können. Dazu gehören folgende Informationen:

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

- Versionsnummer: Mit SCEP wird Version 0 verwendet.
- Liste aller Empfänger und des zugehörigen verschlüsselten Datenverschlüsselungsschlüssels - Mit SCEP gibt es nur einen Empfänger (für Anfragen: den CA-Server; Antworten: Client).
- Die verschlüsselten Daten - Diese werden mit einem zufällig generierten Schlüssel verschlüsselt (der mit dem öffentlichen Schlüssel des Empfängers verschlüsselt wurde).

PKCS#10

PKCS#10 beschreibt das Format einer CSR. Ein CSR enthält die Informationen, die Kunden in ihren Zertifikaten anfordern:

- Betreffname
- Eine Kopie des öffentlichen Schlüssels
- Kennwort für die Anrufweiterleitung (optional)
- Alle beantragten Zertifikatserweiterungen, z. B.: Schlüsselverwendung (KU)Extended Key Usage (EKU)Alternativer Betreff-Name (SAN)Universal Principal Name (UPN)
- Fingerabdruck der Anfrage

Hier ein Beispiel für eine CSR-Anfrage:

```
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
```

Zugehörige Informationen

- SCEP-IETF-Entwurf
- Legacy SCEP mit CLI-Konfigurationsleitfaden
- Konfigurieren der SCEP-Unterstützung für BYOD

Anhang

SCEP-Anfragen

Anforderungsnachrichtenformat

Anfragen werden mit einem HTTP GET-Formular gesendet:

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

Wo:

- **CGI-path** ist vom Server abhängig und verweist auf das Common Gateway Interface (CGI)-Programm, das SCEP-Anfragen verarbeitet: Die Cisco IOS® CA verwendet eine leere Pfadzeichenfolge.Microsoft CA verwendet **/certsrv/mscep/mscep.dll**, der auf den IIS-Dienst MSCEP/ Network Device Enrollment Service (NDES) verweist.
- **Der Vorgang** identifiziert den ausgeführten Vorgang.
- **Nachricht** enthält zusätzliche Daten für diesen Vorgang (und kann leer sein, wenn keine tatsächlichen Daten erforderlich sind).

Bei der GET-Methode ist der **Nachrichtenteil** entweder Klartext oder PKCS#7-kodierte (Distinguished Encoding Rules, DER), die in Base64 konvertiert werden. Wenn die POST-Methode unterstützt wird, können Inhalte, die in Base64-Codierung mit GET gesendet werden, stattdessen im Binärformat mit POST gesendet werden.

Schematische Ansicht

Mögliche Werte für **Vorgänge** und die zugehörigen **Nachrichtenwerte**:

- **operation** = `PKIOperation`: **Meldung**ist eine SCEP **pkiMessage-**Struktur, die auf PKCS#7 basiert und mit DER und Base64 kodiert ist.Die **pkiMessage-**Struktur kann folgende Typen aufweisen: **PKCSReq**: PKCS#10 CSR**GetCertInitial**: Umfrage zur Gewährung des CSR-Status**GetCert** oder **GetCRL**: Zertifikat- oder CRL-Abruf
- **operation** = **GetCACert**, **GetNextCACert** oder (optional) **GetCACaps**: **kann** weggelassen oder auf einen Namen gesetzt werden, der die CA identifiziert.

SCEP-Antworten

Antwortnachrichtenformat

SCEP-Antworten werden als standardmäßiger HTTP-Inhalt zurückgegeben. Der **Inhaltstyp** hängt von der ursprünglichen Anforderung und dem zurückgegebenen Datentyp ab. DER-Inhalt wird als Binärdatei zurückgegeben (nicht in Base64 wie für die Anforderung). PKCS#7-Inhalte enthalten möglicherweise verschlüsselte/signierte verschlüsselte Daten. Wenn dies nicht der Fall ist (nur einen Satz Zertifikate enthält), wird dies als **generierte** PKCS#7 bezeichnet.

Inhaltstypen

Mögliche Werte für **Inhaltstyp**:

Anwendung/x-pki-Nachricht:

- als Antwort auf die **PKIOperation-**Operation mit **pkiMessage** vom Typ: **PKCSReq, GetCertInitial, GetCert oder GetCRL**
- Antworttext ist eine **pkiMessage** vom Typ: **CertRep**

application/x-x509-ca-cert:

- als Antwort auf die **GetCACert-**Operation
- Antworttext ist das DER-codierte X.509 CA-Zertifikat.

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

- als Antwort auf die **GetCACert-**Operation
- Der Antworttext ist eine DER-codierte degenerierende PKCS#7, die die CA- und RA-Zertifikate enthält.

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

- als Antwort auf den **GetNextCACert-**Vorgang
- Antworttext ist eine Variante einer **pkiMessage** vom Typ: **CertRep**

Die pkiMessage-Struktur

SCEP-OIDs

```
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 (als **pkcsPKIEnvelope** bezeichnet; optional, verschlüsselt auf Nachrichtenempfänger)
  messageData (CSR, cert, CRL, ...)
- **SignerInfo** mit **authentifiziertenAttributen**:
  transactionID, **messageType**, **senderNoncepkiStatus**, **empfängerNonce** (nur Antwort)**failInfo** (nur Antwort + Fehler)

SCEP-Nachrichtentyp

- Anforderung:
  PKCSReq (19): PKCS#10 CSR**GetCertInitial** (20): Abfrage der Zertifikatsregistrierung**GetCert** (21): Zertifikatsabruf**GetCRL** (22): CRL-Abruf
- Antwort:
  CertRep (3): Antwort auf Zertifikat oder CRL-Anfrage

SCEP-pkiStatus

- **ERFOLG** (0): Anfrage wird erteilt (Antwort in pkcsPKIEnvelope)
- **FEHLER** (2): Anfrage abgelehnt (Details im failInfo-Attribut)
- **AUSSTEHEND** (3): Anfrage wartet auf manuelle Genehmigung