Simple Certificate Enrollment Protocol の概要

内容

概要

このドキュメントでは、登録やその他の公開キー インフラストラクチャ（PKI）の操作で使用されるプロトコルである Simple Certificate Enrollment Protocol（SCEP）について説明します。

背景説明

最初は Cisco が SCEP を開発し、Internet Engineering Task Force（IETF）のドラフトに文書化されています。

主な特徴は次のとおりです。

- HTTP に基づく要求/応答モデル（GET 方式、POST 方式を任意でサポートします）。
- RSA ベースの暗号化のみをサポートします。
- 証明書要求形式として PKCS#10 を使用します。
- 暗号化署名済み/暗号化メッセージの伝達に PKCS#7 を使用します。
- サーバによる非同期付与を要求者による定期ポーリングでサポートします。
- Certificate Revocation List（CRL；証明書失効リスト）の取得サポートが制限されている(スケーラビリティ上の理由から、CRL分散ポイント(CDP)クエリを使用する方法が推奨されます)
- オンラインでの証明書失効はサポートしません（他の方法を使用してオフラインで実行する必要があります）。
- サーバと要求者間でのみ共有されるようにするため、証明書署名要求（CSR）内の**チャレンジ パスワード フィールドを使用する必要があります。**

通常、SCEP の登録および使用は次のワーク フローに従っています。

1. 認証局（CA）の証明書のコピーを取得して検証します。
2. CSR を生成して CA に安全に送信します。
3. SCEPサーバをポーリングして、証明書が署名されたかどうかを確認します。
4. 現在の証明書の期限が切れる前に新しい証明書を取得するため、必要に応じて再登録します。
5. 必要に応じて CRL を取得します。

CA 認証

CSR のメッセージ交換を保護するため、SCEP は CA 証明書を使用します。その結果、CA 証明書のコピーを取得する必要があります。GetCACert オペレーションが使用されます。

Request

要求は HTTP GET 要求として送信されます。要求のパケット キャプチャは次のようになります。

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

応答

応答は、バイナリエンコードされた CA 証明書（X.509）に似ています。 クライアントは、フィンガープリント/ハッシュの検査により CA 証明書が信頼できることを検証する必要があります。これはアウトオブバンド方式（システム管理者への電話またはトラストポイント内でのフィンガープリントの事前設定）で実行する必要があります。

クライアント登録

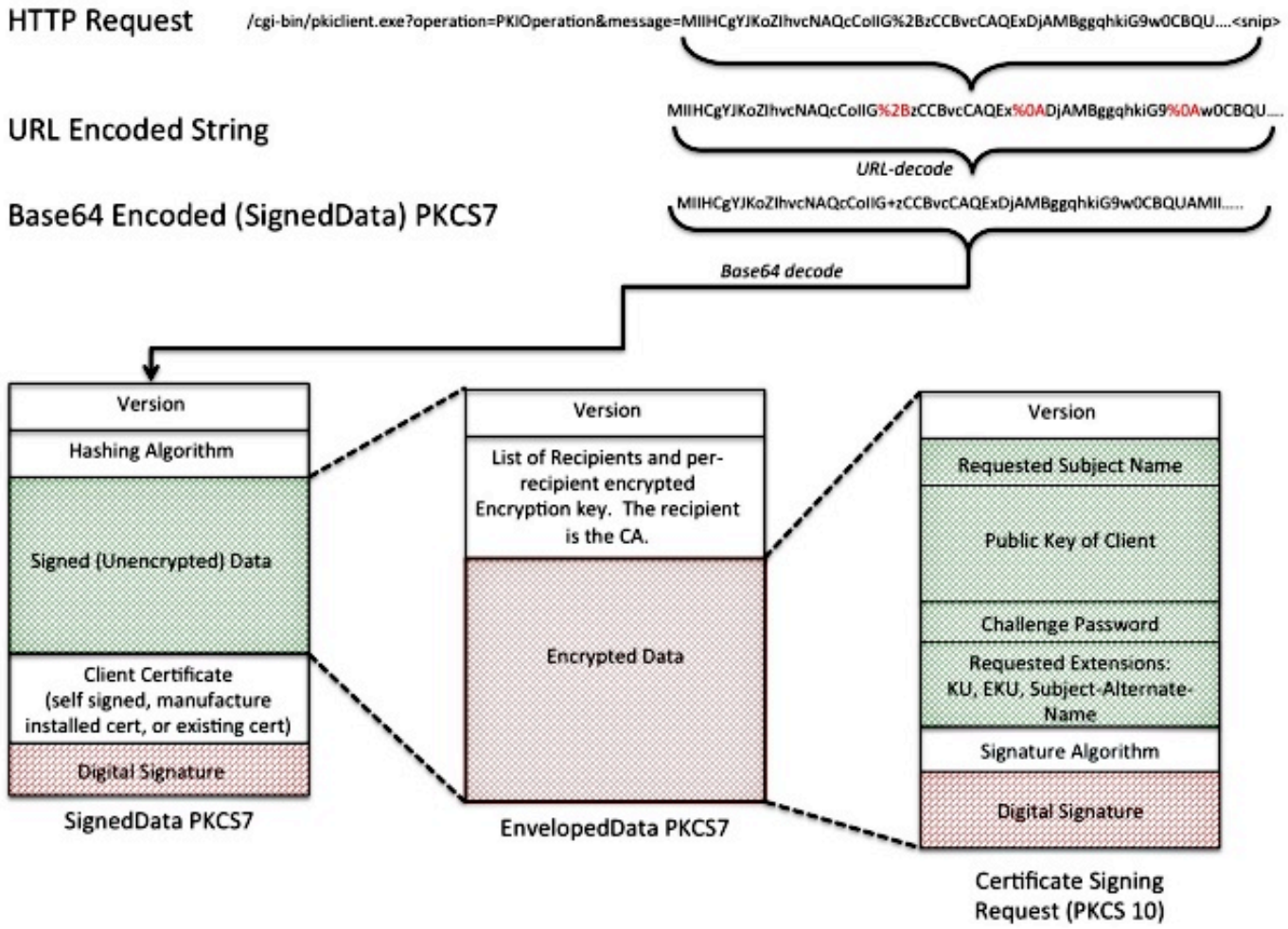
Request

登録要求はHTTP GET要求として送信されます。要求のパケットキャプチャは次のようになりま

す。

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

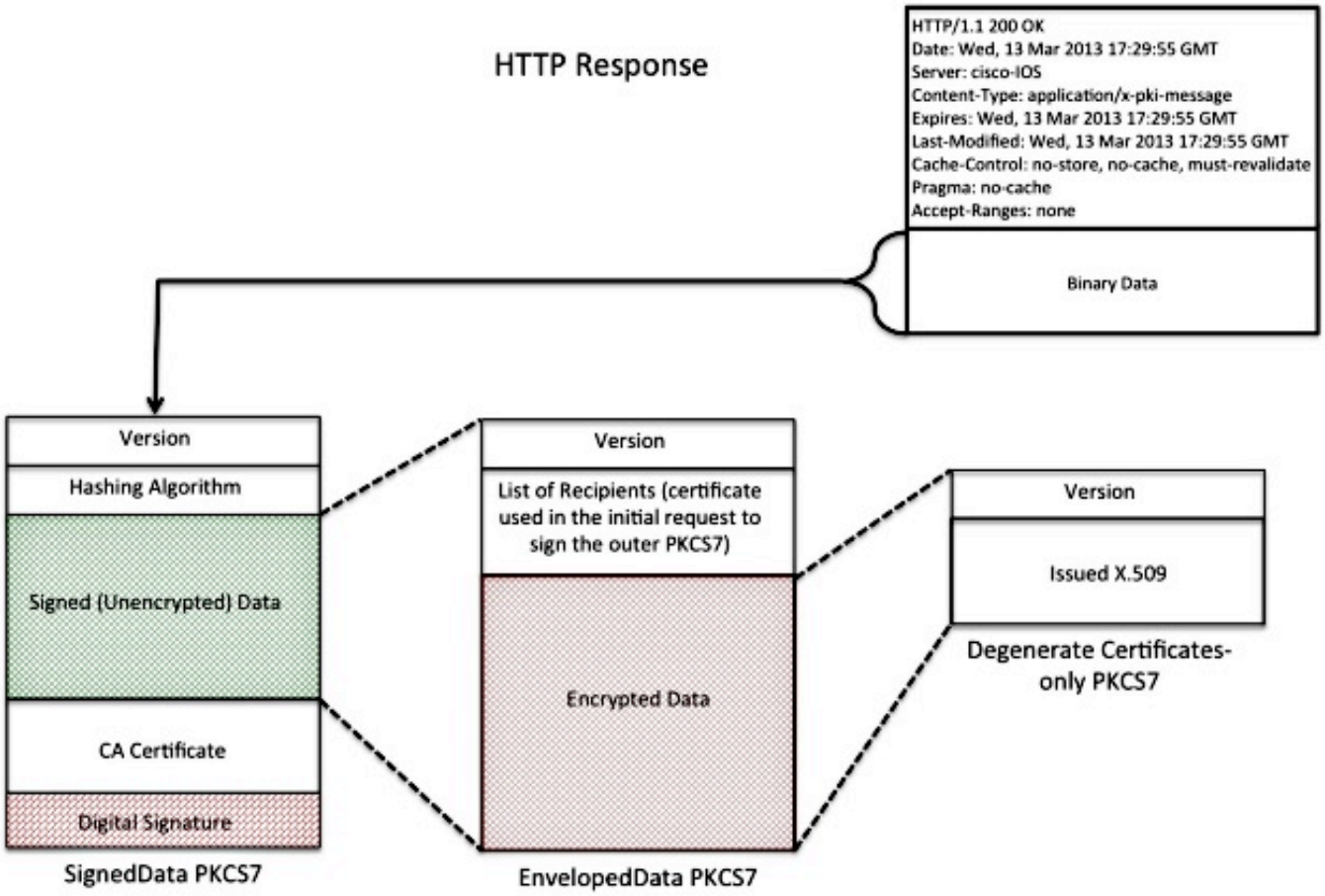
1. 「message=」の後のテキストは URL のエンコードされた文字列で、GET 要求文字列から抽出されます。
2. 次に、テキストは URL が ASCII テキスト文字列にデコードされます。そのテキスト文字列は Base64 エンコードの SignedData PKCS#7 です。
3. SignedData PKCS#7 は、次の証明書のいずれかを使用してクライアントによって署名されます。これは、クライアントが送信したものであること、および送信中に変更されていないことを証明するために使用されます。
   自己署名証明書（最初の登録時に使用）製造元でインストールされる証明書（MIC）期限切れ間近の現在の証明書（再登録）
4. SignedData PKCS#7 の「署名済みデータ」部分は EnvelopedData PKCS#7 です。
5. EnvelopedData PKCS#7 は「暗号化されたデータ」と「復号キー」を含むコンテナです。
   復号キーは、受信者の公開キーで暗号化されます。この特定のケースでは、受信者は CA です。その結果、「暗号化されたデータ」を実際に復号できるのは CA のみです。
6. エンベロープ PKCS#7 の「暗号化されたデータ」部分は CSR（PKCS#10）です。



応答

SCEP 登録要求への応答は、次の 3 つのタイプのいずれかです。

- **拒否：次のようないくつかの理由で、管理者によって要求が拒否されます。**
  無効なキー サイズ無効なチャンレンジ パスワードCA が要求を検証できませんでした。要求で求めた属性を CA が許可しませんでした。CA が信頼していないアイデンティティによって要求が署名されました。
- **保留**：CA 管理者が要求をまだ確認していません。
- **成功：要求が承認され、署名済み証明書が含まれています。**署名済み証明書は「Degenerate Certificates-Only PKCS#7」という特殊なタイプの PKCS#7 内に保持されます。このタイプの PKCS#7 は 1 つ以上の X.509 または CRL を保持できる特殊なコンテナですが、署名済みまたは暗号化されたデータのペイロードは含まれていません。



クライアントの再登録

証明書の期限が切れる前に、クライアントは新しい証明書を取得する必要があります。更新と口ールオーバーの動作にはわずかな違いがあります。更新はクライアントの ID 証明書の期限が近づくと発生し、その有効期限は、CA 証明書の有効期限と同じではありません（それよりも前になります）。ロールオーバーは ID 証明書の期限が近づくと発生し、その有効期限は CA の証明書有効期限と同じです。

更新

ID 証明書の有効期限が近づくにしたがい、SCEP クライアントは新しい証明書を取得することが必要になる場合があります。クライアントは CSR を生成し、登録プロセスを（以前に定義したとおりに）実行します。 現在の証明書は、SignedData PKCS#7に署名するために使用され、

CAにIDが証明されます。新しい証明書が受信されると、クライアントは現在の証明書を即座に削除し、新しい証明書に置き換えます。この証明書の有効性は、すぐに開始されます。

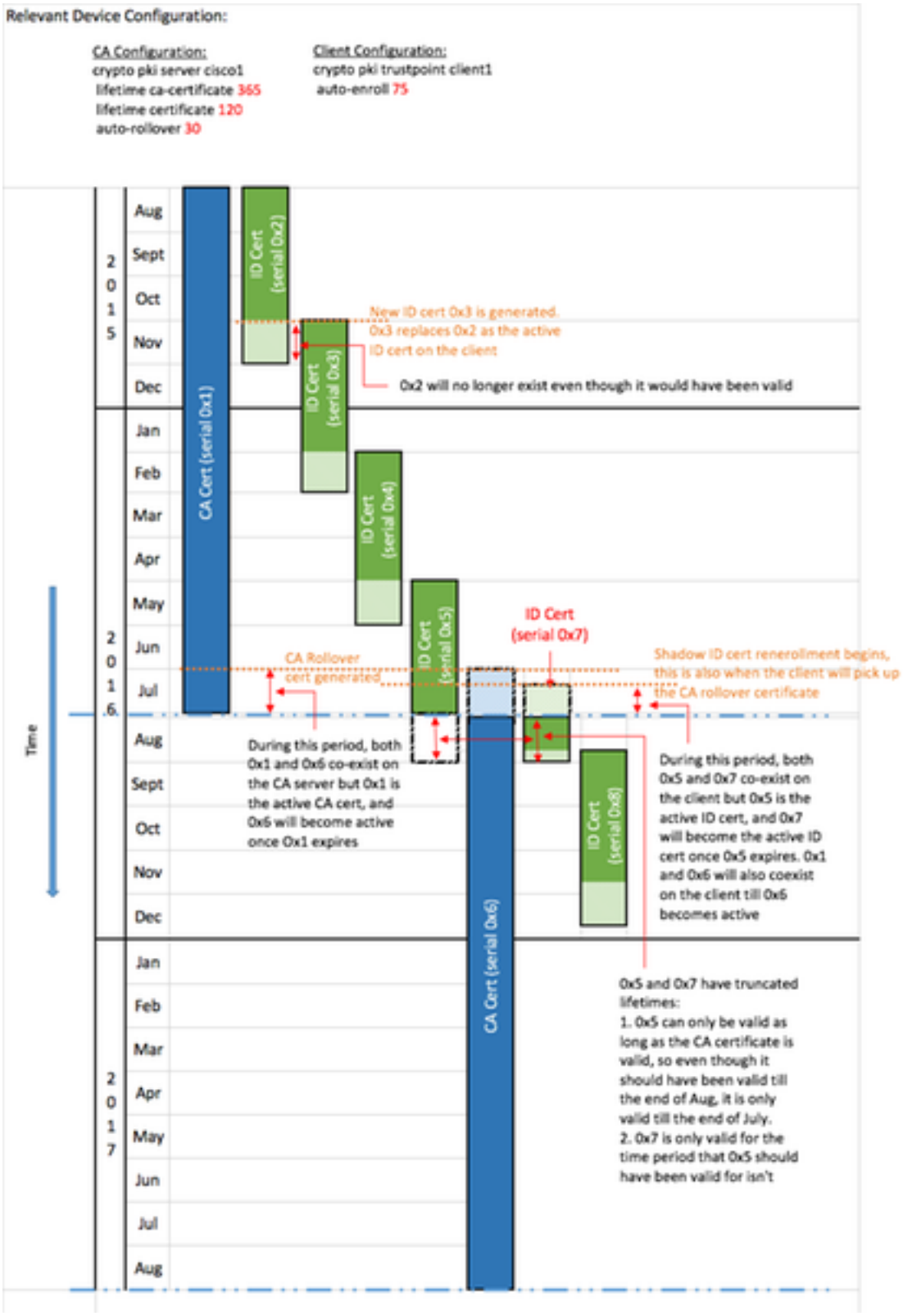
ロールオーバー

ロールオーバーは、CA 証明書の期限が切れ、新しい CA 証明書が生成される特殊なケースです。CA が新しい CA 証明書を生成し、現在の CA 証明書の期限が切れた時点でその証明書が有効になります。通常、CA はこの「シャドー CA」証明書をロールオーバーのタイミングよりも前に生成します。これは、クライアントの「シャドー ID」証明書を生成するために必要であるからです。

SCEP クライアント の ID 証明書の期限が近づくと、SCEP クライアントは CA に「シャドー CA」証明書について問い合わせます。これは、次に示すように、**GetNextCACert オペレーションで実行されます。**

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

SCEPクライアントが「シャドウCA」証明書を取得すると、通常の登録手順の後で「シャドウID」証明書を要求します。CA は「シャドー CA」証明書を使用して「シャドー ID」証明書に署名します。通常の更新要求とは異なり、返された「シャドー ID」証明書は CA 証明書の有効期限が切れた時点で有効になります (ロールオーバー) 。 その結果、クライアントは CA 証明書と ID 証明書の両方について、ロールオーバーの前と後の証明書のコピーを保管する必要があります。CA 証明書の期限が切れた (ロールオーバー) 時点で、SCEP クライアントは現在の CA 証明書と ID 証明書を削除し、それらを「シャドー」コピーに置き換えます。



構成要素

次の構造が SCEP の構成要素として使用されます。

注：PKCS#7 と PKCS#10 は SCEP 固有ではありません。

PKCS#7

PKCS#7 は、データを署名または暗号化できるように定義されたデータ形式です。このデータ形式には、元のデータと暗号化操作の実行に必要な関連メタデータが含まれています。

署名済みエンベロープ（SignedData）

署名済みエンベロープは、データを搬送し、カプセル化されたデータがデジタル署名を通じて送信中に変更されていないことを確認する形式です。内容は次のとおりです。

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

- バージョン番号：SCEP ではバージョン 1 が使用されます。
- 使用されるダイジェスト アルゴリズムのリスト：SCEP では、署名者は 1 人のみであるため、ハッシュ アルゴリズムも 1 つのみです。
- 署名された実際のデータ：SCEP では、PKCS#7 エンベロープデータ形式（暗号化されたエンベロープ）です。
- 署名者の証明書のリスト – SCEPでは、これは初期登録時の自己署名証明書または再登録の場合は現在の証明書です。
- 署名者と各署名者が生成したフィンガープリントのリスト：SCEP では、署名者は 1 つのみです。

カプセル化されたデータは暗号化も難読化もされません。この形式は、変更されているメッセージに対する保護のみを提供します。

エンベロープ データ（EnvelopedData）

エンベロープ データ形式は暗号化されたデータを搬送し、指定された受信者のみが復号できます。 内容は次のとおりです。

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

- バージョン番号：SCEP では、バージョン 0 が使用されます。
- 各受信者と関連する暗号化データおよび暗号化キーのリスト：SCEP では、受信者は 1 つのみです（要求の場合：CA サーバ、応答の場合：クライアント）。
- 暗号化データ：これは、ランダムに生成されたキー（受信者の公開キーを使用して暗号化されている）で暗号化されます。

PKCS#10

PKCS#10 は CSR の形式を説明します。CSR には、クライアントが証明書内に含めるように要求した情報が含まれています。

- 件名

- 公開キーのコピー
- チャレンジ パスワード（任意）
- 要求された次のような証明書拡張子
  キー使用の目的（KU）拡張キーの使用状況（EKU）サブジェクト代替名（SAN）ユーザ プリンシパル名（UPN）
- 要求のフィンガープリント

次に、CSR の例を示します。

```
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
```

関連情報

- SCEP IETF のドラフト
- CLI コンフィギュレーション ガイドを使用したレガシー SCEP
- BYOD のための SCEP サポートの設定

付録

SCEP 要求

要求メッセージ形式

要求は次の形式の HTTP GET を使用して送信されます。

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

場所：

- **CGI-path はサーバによって異なり、SCEP 要求を処理する Common Gateway Interface (CGI) プログラムを示します。** Cisco IOS® CA は空のパス文字列を使用します。Microsoft CA は **/certsrv/mscep/mscep.dll** を使用し、MSCEP/ネットワーク デバイス登録サービス (NDES) IIS サービスを示します。
- **operation は、実行されるオペレーションを特定します。**
- **message はそのオペレーションの追加データを搬送します (また、実際のデータが不要である場合は空にできます) 。**

GETメソッドを使用すると、メッセージ部分は、プレーンテキストまたはDistinguished Encoding Rules (DER)でエンコードされたPKCS#7でBase64に変換されます。POSTメソッドがサポートされている場合、GETで送信される内容はバイナリ形式です。

概略図

operation の可能な値とそれらの関連 message の値は次のとおりです。

- **operation = PKIOperation: メッセージ**は、PKCS#7に基づき、DERおよびBase64でエンコードされたSCEP **pkiMessage**構造です。pkiMessage の構造は次のいずれかのタイプです。**PKCSReq**：PKCS#10 CSR**GetCertInitial**：CSR 付与ステータスのポーリング**GetCert または GetCRL**：証明書または CRL の取得
- **operation = GetCACert、GetNextCACert、または (任意) GetCACaps： message は省略できます。または、CA を特定する名前に設定できます。**

SCEP 応答

応答メッセージ形式

SCEP 応答は、元の要求と返されるデータのタイプに応じて Content-Type で標準的な HTTP コンテンツとして返されます。DER コンテンツは (要求と同じ Base64 ではなく) バイナリとして返されます。PKCS#7 コンテンツには、暗号化/署名済みエンベロープ データが含まれていることも、含まれていないこともあります。含まれていない (一連の証明書のみが含まれている) 場合は、**劣化 PKCS#7 と呼ばれます。**

コンテンツ タイプ

Content-Type の可能な値

application/x-pki-message：

- **PKIOperation オペレーションへの応答でタイプが pkiMessage** ：**PKCSReq、GetCertInitial、GetCert、または GetCRL**
- 応答の本文が **pkiMessage タイプ：CertRep**

application/x-x509-ca-cert：

- GetCACert オペレーションへの応答
- 応答の本文が DER でエンコードされた X.509 CA 証明書

application/x-x509-ca-ra-cert：

- GetCACert オペレーションへの応答
- 応答の本文は、CA 証明書と RA 証明書を含む DER でエンコードされた劣化 PPKCS#7

application/x-x509-next-ca-cert：

- GetNextCACert オペレーションへの応答
- 応答の本文が pkiMessage タイプの変形：CertRep

pkiMessage 構造

SCEP OID

```
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（着信側pkcsPKIEnvelope、任意、メッセージ受信者に対して暗号化）
  messageData（CSR、cert、CRL、...）
- authenticatedAttributes の SignerInfo ：
  transactionID、messageType、senderNoncepkiStatus、recipientNonce（応答のみ）failInfo（応答 + 失敗のみ）

SCEP messageType

- 要求：
  PKCSReq (19)：PKCS#10 CSRGetCertInitial (20)：証明書登録のポーリングGetCert (21)：証明書の取得GetCRL (22)：CRL の取得
- 応答：
  CertRep (3)：証明書の CRL 要求への応答

SCEP pkiStatus

- SUCCESS (0)：許可された要求（pkcsPKIEnvelope での応答）
- FAILURE (2)：拒否された要求（failInfo 属性の詳細）
- PENDING (3)：要求は手動による承認待ち