certificate Enrollment over Secure Transport (EST)

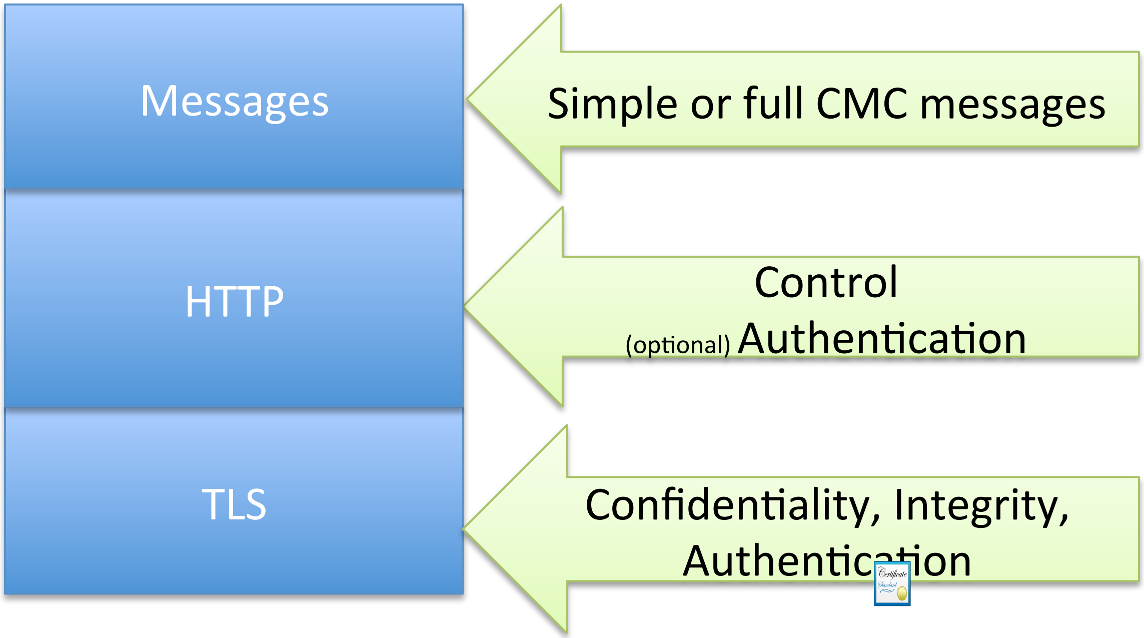
An enrollment protocol profile

draft-pritikin-est-02

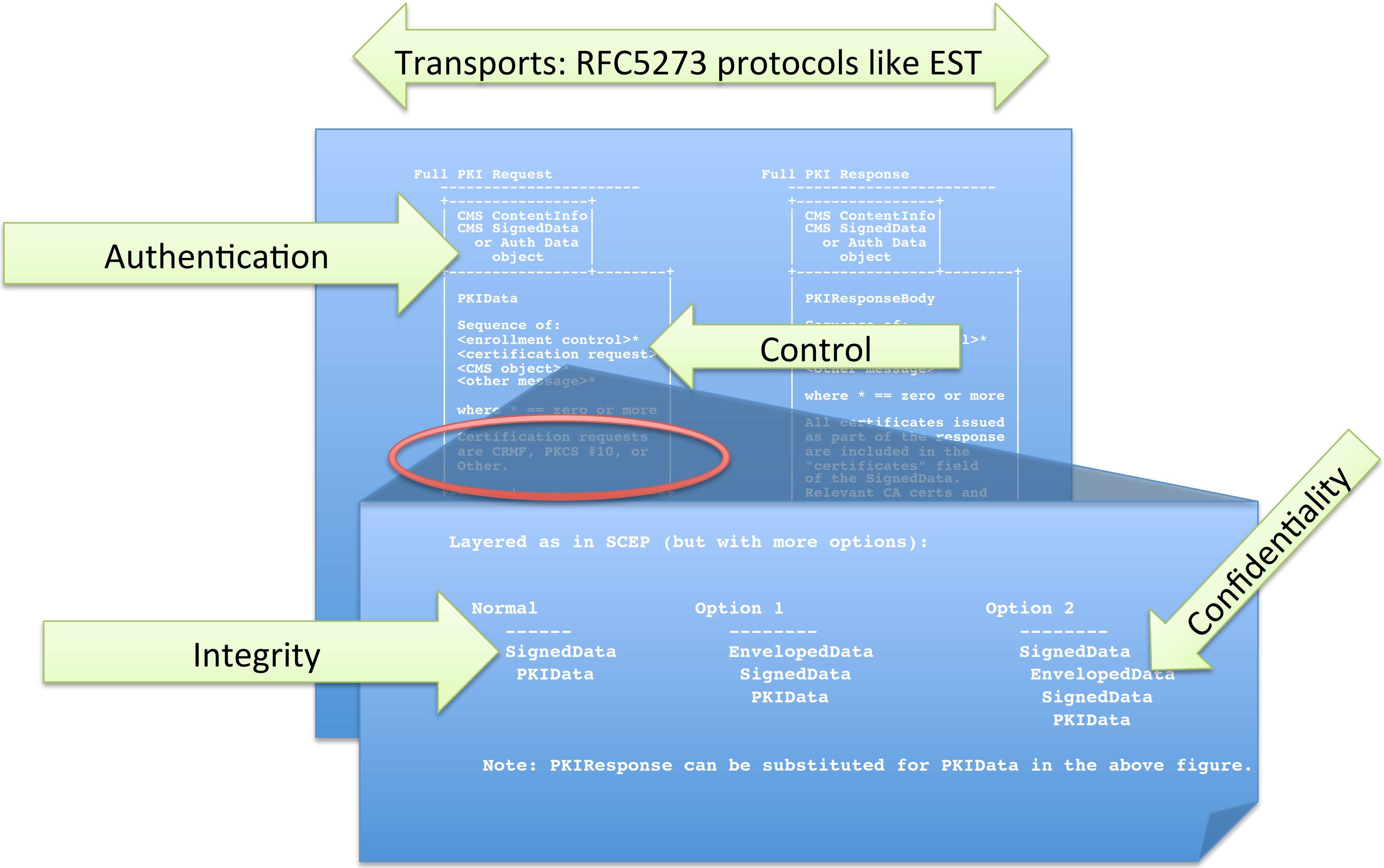
EST Motivation

(summary, synopsis, recap of IETF80 EST presentation)

- **Simple** implementation
- **Profile** of CMC & CMC:Transport
- **Accentuate** Re-key/re-enroll from CMP


Messages
HTTP
TLS
Simple or full CMC messages
Control
(optional) Authentication
Confidentiality, Integrity,
Authentication

Full CMC messages



```
Full PKI Request                       Full PKI Response
  -----------------------                -----------------------
  +----------------+                     +----------------+
  | CMS ContentInfo|                     | CMS ContentInfo|
  | CMS SignedData |                     | CMS SignedData |
  |   or Auth Data |                     |   or Auth Data |
  |     object     |                     |     object     |
  +----------------+--------+            +----------------+--------+
  |                         |            |                         |
  | PKIData                 |            | PKIResponseBody         |
  |                         |            |                         |
  | Sequence of:            |            | Sequence of:            |
  | <enrollment control>*   |            |                 l>*     |
  | <certification request> |            |                         |
  | <CMS object>*           |            | <other message>*        |
  | <other message>*        |            |                         |
  |                         |            | where * == zero or more |
  | where * == zero or more |            |                         |
  |                         |            | All certificates issued |
  | Certification requests  |            | as part of the response |
  | are CRMF, PKCS #10, or  |            | are included in the     |
  | Other.                  |            | "certificates" field    |
  |                         |            | of the SignedData.      |
  |                         |            | Relevant CA certs and   |
  +-------------------------+
```

```
Layered as in SCEP (but with more options):

Normal              Option 1               Option 2
------              --------               --------
SignedData          EnvelopedData          SignedData
 PKIData             SignedData             EnvelopedData
                      PKIData                SignedData
                                              PKIData

Note: PKIResponse can be substituted for PKIData in the above figure.
```

Updates to draft

- Basic structural updates
- Requirements section

  To be moved to a distinct document?

  Please comment if there is a requirement missing.

  Be proactive: supply a solution with your requirement!
- Proof-of-Possession

Proof-of-Possession

“the CA is adequately convinced that the entity requesting a certificate for the public key Y, has access to the corresponding private key X” [CRMF]

Proof-of-Possession (POP) refers to a value that can be used to prove that the private key corresponding to a public key is in the possession and can be used by an end-entity.

Of the different types of POP defined in CMC, EST focuses on:

- Signature
  - Provides the required POP by a signature operation over **some data**.
- Attested
  - Trusted entity asserts that the POP has been proven

EST does not use ~~Direct,~~ ~~Publish~~, or ~~Indirect~~

EST Proof-of-Possession

- Signature: Add “some data” to the signed message (CRMF/PKCS#10) which is pertinent to the exchange, available to both EST client & server, and relatively easy to get to:

  TLS binding information similar to **tls-unique**

- tls-unique**-securerenegotiation**

  The first TLS Finished message sent in the **first** TLS handshake of the TLS connection. Secure Renegotiation is mandated.

  This is the same as tls-unique prior to renegotiation

- ‘Attested’ is the fallback case

  If tls-unique-sr is not valid then the client must be trusted to have already checked PoP

  Server has an authenticated client ID to determine trust

Prototype experience

- Addition of of ‘tls-unique’ to mod_ssl allows support of this binding method from simple CGI scripts running under Apache
- Use existing openssl ‘tls-unique’ API

  Call this before renegotiation occurs

  (e.g. SSL_get_finished / SSL_get_peer_finished)
- The simplest client implementations can use ‘attested’ as a fallback (not ideal)

```
curl $URL -s -d $PKCS10FILE -o $NEWCERT —E
$EXISTINGCERT —cacert $CACERT —u user:pwd
```

Questions & Answers

EST as a Working Group item?