Trust Anchor Management Requirements

Carl Wallace

cwallace@cygnacom.com

Background

- Initial work was done for the TAM BOF held during Chicago meeting last summer
- BOF did not yield a new working group
  - Work was moved to PKIX
  - New PKIX charter has been established

Working group comments

1. Targets for management
2. TA terminology
3. Types of associated data
4. Document organization

Targets for management

- Three targets have been suggested:
  - Individual TAs within a trust store
    - Focus of draft
  - Entire trust store
    - Suggested by Denis
  - Validation policies
    - Suggested by Denis

TA Terminology

- The TA definition in the draft essentially includes a fifth item under the 3280 statement of what a trust anchor includes:
  - (5) optionally, associated data used to constrain the types of information for which the trust anchor is authoritative
- Denis prefers TAAD to TA for this

Types of associated data

- Additional types
  - Revocation status checking mechanisms and parameters
- Nature of association
  - Per TA vs. Per group of TAs

Document organization

- Draft history
  - Initial draft submitted for TAM BOF,
  - Initial PKIX draft before Vancouver meeting (same content as last TAM BOF version)
  - -01 submitted in February (minor edits vs. -00)
- Content will be re-factored into a requirements draft shortly after IETF71
  - Requirements presently in security considerations will be moved into the body of the draft
  - Requirement description and rationale will be presented

Distilled Requirements

- Provide transport independence and applicability to session-oriented and store-and-forward contexts
- Enable a trust anchor manager to:
  - Discover trust stores
  - Report trust store contents
  - Add trust anchors to a trust store
  - Remove trust anchors from a trust store
  - Replace entire trust store (new requirement)
- Enable generation of messages intended for:
  - All stores that recognize TA manager
  - A group of stores (or groups of stores)
  - An individual store

Distilled requirements (cont.)

- Enable secure transfer of control of trust store management responsibility from one TA manager to another
  - Rekey is one example
- Support RFC 3280 certification path validation
- Enable usage of trust anchors for purposes other than certification path validation
  - Include a key identifier in trust anchor content to enable CMS-based applications
- Enable management of trust anchors that do not serve as trust anchors for certification path validation

Distilled requirements (cont.)

- Support management of trust anchors represented as self-signed certificates or as a distinguished name and public key information
- Enable authentication of device that produced a report listing the contents of a trust anchor store
  - Enable replay detection for TA store reports
- Enable the representation of constraints that influence certification path validation or otherwise establish the scope of usage of the trust anchor public key
  - Enable delegation of privileges
  - Limit trust anchor managers to a particular scope

Distilled requirements (cont.)

- Enable confirmation of TA mgmt. message integrity
- Enable authentication of TA mgmt. message originator and confirmation of authorization to originate TA mgmt. messages
- Reduce reliance on out-of-band trust mechanisms
- Enable replay detection without requiring a reliable source of time
- Support recovery from compromise of trust anchor private key

Comparison of ValidationPolicy and TrustAnchorInfo

```
ValidationPolicy ::= SEQUENCE {
  validationPolRef          ValidationPolRef,
  validationAlg         [0] ValidationAlg OPTIONAL,
  userPolicySet         [1] SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL,
  inhibitPolicyMapping  [2] BOOLEAN OPTIONAL,
  requireExplicitPolicy [3] BOOLEAN OPTIONAL,
  inhibitAnyPolicy      [4] BOOLEAN OPTIONAL,
  trustAnchors          [5] TrustAnchors OPTIONAL,
  keyUsages             [6] SEQUENCE OF KeyUsage OPTIONAL,
  extendedKeyUsages     [7] SEQUENCE OF KeyPurposeId OPTIONAL,
  specifiedKeyUsages    [8] SEQUENCE OF KeyPurposeId OPTIONAL }
```

```
TrustAnchorInfo ::= SEQUENCE {
   version   [0] TAMPVersion DEFAULT v2,
   pubKey    PublicKeyInfo,
   keyId     KeyIdentifier,
   taType    TrustAnchorType,
   taTitle   TrustAnchorTitle OPTIONAL,
   certPath  CertPathControls OPTIONAL }
```

```
CertPathControls ::= SEQUENCE {
  taName            Name,
  selfSigned        [0] Certificate
                        OPTIONAL,
  policySet         [1] CertificatePolicies
                        OPTIONAL,
  policyFlags       [2] CertPolicyFlags
                        OPTIONAL,
  clearanceConstr   [3]
              CAClearanceConstraints OPTIONAL,
  nameConstr        [4] NameConstraints
                        OPTIONAL }
```

Comparison of ValidationPolicy and TrustAnchorInfo

- ValidationPolicy associates data with groups of TAs vs. per TA
- Mainly common information, differences include:
  - ValidationPolicy has key usages
  - TrustAnchorInfo has name constraints, Apex information, CMS content constraints, key identifier, friendly name
- TrustAnchorInfo meets several requirements not met by ValidationPolicy, including
  - Representation of TA not used for path validation
  - Recovery from compromise
  - Self-signed or DN/key representation

Questions?