Overview of draft-ietf-sidr-roa-00.txt

Steve Kent

BBN Technologies

Presentation Outline

- Route origination security
- Proposed ROA design
- Design rationale
- Proposed ROA format
- ROA and NLRI matching
- Questions

Route Origination Security

- One goal of this PKI is to enable ISPs to verify route origination assertions in BGP UPDATE messages
- To support this goal, each address space holder needs to digitally sign one or more objects that identify each AS authorized to advertise routes on behalf of the address space holder
- We call the object a route origination authorization (ROA)
- An address space holder issues a distinct ROA to each ISP he wants to advertise all or a portion of his address space
- Since each ISP is an address space holder, it would sign one or more ROAs (one per AS number) authorizing itself to advertise the addresses it holds

ROA Design

- A ROA has three major data elements, encapsulated in a CMS signed data object
  - A version number
  - One of more address prefixes, corresponding to the NLRI that the ROA signer authorizes for origination by one or more ISPs, enumerated below
  - An AS number of an ISP authorized to originate routes to the above list of prefixes
- A ROA is valid only if the (EE) certificate used to verify its signature is valid, and if the address space extension in the certificate exactly matches the prefixes in the ROA
- We use the CMS format to represent a signed ROA, as this format is supported in open source software

ROA Design Rationale

- We assume a 1-1 match between ROAs and the EE certificates used to verify them
  - This makes it easier to manage revocation of ROAs
  - It avoids the need to put a validity interval in a ROA
  - Can discard private key after signing the ROA
- Just one AS per ROA (Randy), to keep the ROA design simple, and to enforce the notion that each ROA authorizes only one ISP (Geoff)
- Include the prefix(es) in the ROA to make it easier for a human user to understand (Randy)
- We use the prefix representation from RFC 3779 to make matching against the certificate easy

ROA Format (1/2)

```
RouteOriginAttestation ::= SEQUENCE {
  version [0] INTEGER DEFAULT 0,
  -- this is the ROA version #
  asID   ASID,
  ipAddrBlocks ROAIPAddrBlocks }

ASID ::= INTEGER
-- this is the AS number

ROAIPAddrBlocks ::= SEQUENCE of ROAIPAddressFamily

ROAIPAddressFamily ::= SEQUENCE {
  addressFamily OCTET STRING (SIZE (2..3)),
  addressesOrRanges SEQUENCE OF IPAddressOrRange }
-- Only two address families: IPv4 and IPv6
```

ROA Format (2/2)

```
IPAddressOrRange ::= CHOICE {
        addressPrefix IPAddress,
        addressRange  IPAddressRange }

      IPAddressRange ::= SEQUENCE {
        min IPAddress,
        max IPAddress }

      IPAddress ::= BIT STRING
```

This syntax is taken from RFC 3779

Profiled CMS Format (1/3)

```
SignedData ::= SEQUENCE {
    version CMSVersion,
    -- version number is 3
    digestAlgorithms DigestAlgorithmIdentifiers,
    -- for a ROA, just one, SHA-256
    encapContentInfo EncapsulatedContentInfo,
    certificates [0] IMPLICIT CertificateSet OPTIONAL,
    -- only one certificate for a ROA, and only if
        transmitted (vs. stored in a repository)
    signerInfos SignerInfos }

  DigestAlgorithmIdentifiers ::=
              SET OF DigestAlgorithmIdentifier
              -- for a ROA, just one alg identifier

  SignerInfos ::= SET OF SignerInfo
  -- for a ROA, just one entry
```

Profiled CMS Format (2/3)

```
  EncapsulatedContentInfo ::= SEQUENCE {
     eContentType ContentType,
     eContent [0] EXPLICIT OCTET STRING OPTIONAL }

     ContentType ::= OBJECT IDENTIFIER


id-smime OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) pkcs(1) pkcs9(9) 16 }

  id-ct OBJECT INDENTIFIER ::= {id-smime 1}

  routeOriginAttestion OBJECT IDENTIFIER ::= {id-ct 24}
  -- this is the OID for a ROA
```

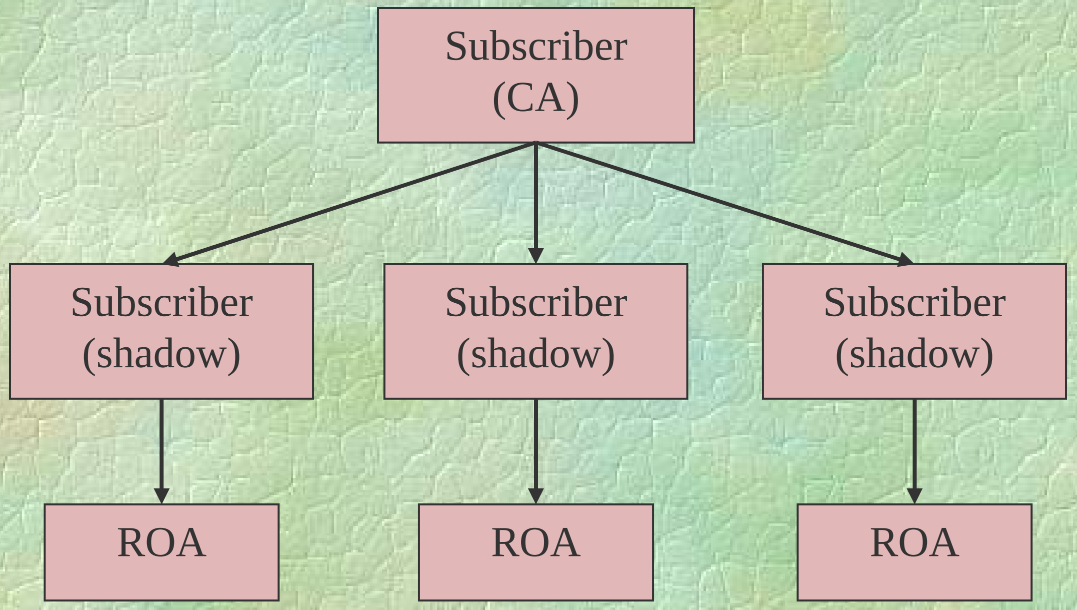
Profiled CMS Format (3/3)

```
SignerInfo ::= SEQUENCE {
      version CMSVersion,
          -- version number is 3
      sid SignerIdentifier,
          -- for a ROA, this is an SKI, a back-pointer
             to the (EE) certificate for verification
      digestAlgorithm DigestAlgorithmIdentifier,
          -- for a ROA, this is SHA-256
      signatureAlgorithm SignatureAlgorithmIdentifier,
          -- for a ROA, this is SHA-256 with RSA
      signature SignatureValue }
```

Generating a ROA

- An ISP or subscriber generates one ROA for each AS for which it wants to authorize route origination
  - If all prefixes of the resource holder are to be advertised by the same ISPs, and came from one source, then the entity signs one ROA containing all the prefixes
  - If the resource holder wants to authorize some ISPs to originate some prefixes, and other ISPs to originate other prefixes, the holder signs one ROA for each set of addresses to be originated by each ISP (AS)
  - If a resource holder has addresses from different sources, it MUST generate multiple ROAs, each restricted to one allocation source

Multiple ROA Example



Subscriber (CA)

Subscriber (shadow)

Subscriber (shadow)

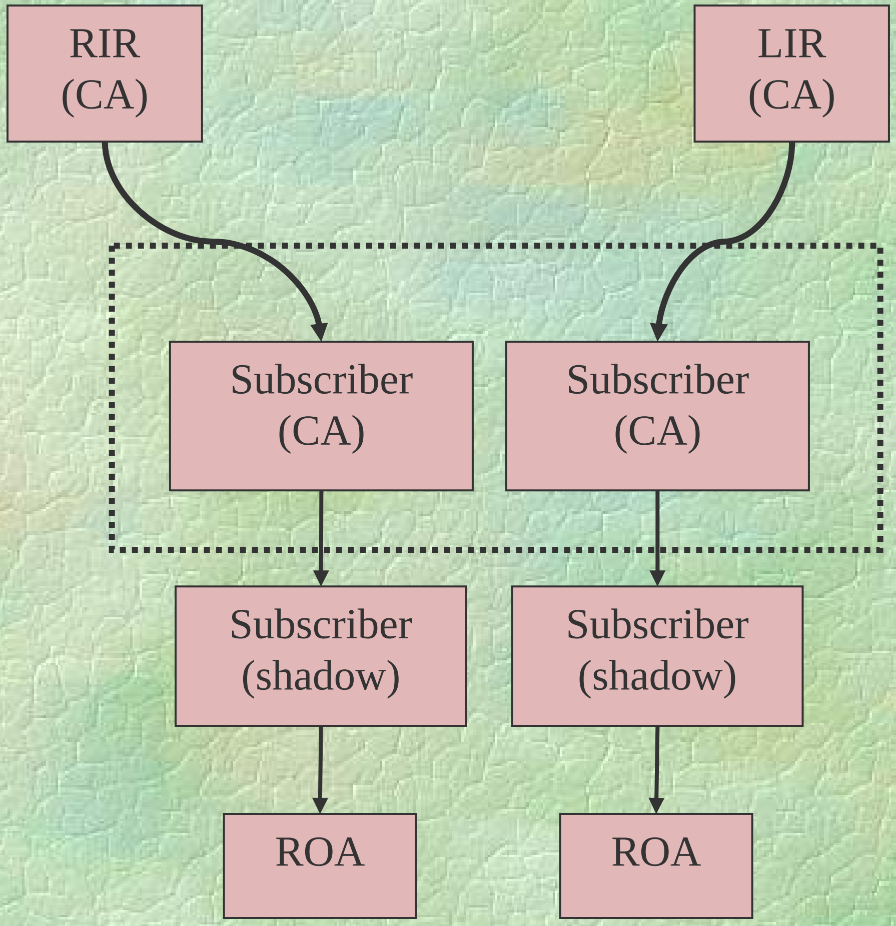
Subscriber (shadow)

ROA

ROA

ROA

Multi-source Allocation Example



ROAs and Prefix Matching (1/3)

- This I-D says that there MUST be an exact match between the prefix(es) in a ROA and those in the EE certificate used to verify it
- The I-D does **not** say how to match the addresses represented in a ROA to the NLRI in a BGP UPDATE
- Suggested answers include
  - Exact match
  - Exact or more specific
  - Changing the ROA to express a range of prefix lengths for determining a match
  - Changing the ROA to look like an RPSL declaration

ROAs and Prefix Matching (2/3)

- Some observations
  - We have two places at which to make prefix comparisons : ROA vs. EE certificate and ROA vs. NLRI
  - I think we should keep ROA/certificate matching simple
  - I think we should stick with ASN.1 in the ROA, not mix ASN.1 and text (canonicalization is a solved problem for the RFC 3779 expression of addresses/ranges)
  - A resource holder can create multiple ROAs if necessary to express authorization

ROA and prefix Matching (3/3)

- Observations from the SIDR WG meeting
  - A resource certificate and a ROA can express address ranges that are NOT CIDR prefixes, so one cannot mandate an exact match between NLRI (which must be a prefix) and a ROA in all cases
  - In such cases, one could issue two (or more) ROAs each of which is a CIDR prefix to cover the prefixes in a BGP advertisement; the ability to match multiple ROAs to an advertisement MUST be supported to accommodate multiple allocation sources anyway
  - One also could formulate an ASN.1 version of RPSL expressiveness for advertisement authorization