Resource Certificate Profile

Geoff Huston, George Michaelson, Rob Loomans

APNIC

IETF 67

Resource Certificate Profile

Background:

- This certificate is intended to express a “right-of-use relationship between the subject and an IP number resource set, as certified by the certificate’s issuer
- The certificate structure is intended to follow the allocation path – each party certifies their own allocation actions, so that the Issuer’s attestation regarding “right-of-use” mirrors the Issuer’s allocation actions of the number resource to a Subject
- The base profile is RFC3280 PKI Certificate Profile and RFC3779 IP Address extensions
- The proposed profile for Resource Certificates is in **draft-ietf-sidr-res-certs**
- This draft has been produced by an APNIC editing group, with input from a design team and this WG

draft-ietf-sidr-res-certs

- General constraints:
  - RFC3779 extensions are a CRITICAL extension and MUST be present, using a sorted canonical representation
  - An Issuer cannot certify more resources than the Issuer has in existing valid resource certificates
  - An Issuer cannot certify the same resource to 2 or more distinct Subjects

draft-ietf-sidr-res-certs

- **Certificate Fields**:

  Version = 3

  Serial Number = positive integer

  Signature Algorithm = SHA256 with RSA

  Subject Public Key Info = Minimum bit size of 1024 bits. Intended root certificates should use key size = 2048 bits

  Basic Constraints = CA ON for allocation certificates, CA = OFF for signing certificates

  Subject Key Identifier = 160 bit SHA-1 hash of the subject's public key

  Authority Key Identifier = 160 bit SHA-1 hash of the issuer's public key

  CRLDP =single CRL, with at least an RSYNC:: object URL

  AIA = *publication point of Issuer's immediate superior certificate (in the form of a PURL*), with at least an RSYNC:: object URI

  SIA = if a CA, publication point of all issued certificates, *or if an EE cert, the URL of the object signed with this EE Cert,* with at least an RSYNC:: directory URI

Draft-ietf-sidr-res-certs

- **Certificate Revocation List Fields**

  Scope = all certificates issued by this CA

  Version = 2

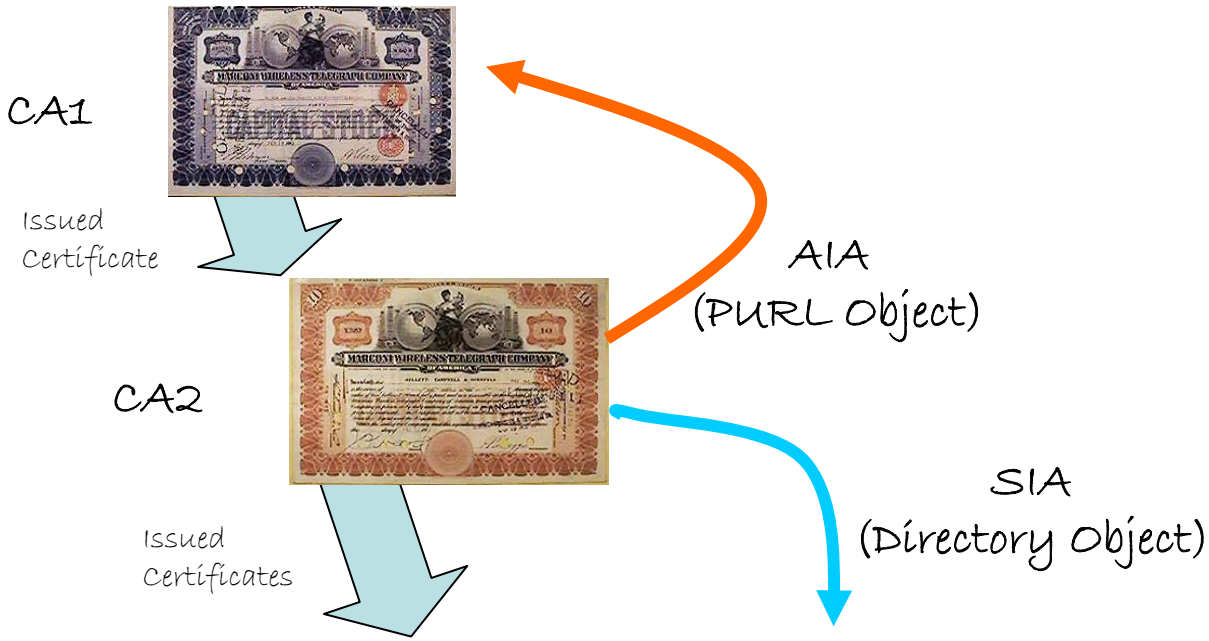
  Authority Key Identifier = 160 bit SHA-1 hash of the issuer's public key

  CRL Number = monotonically increasing integer

Current Activity

- The AIA points to the Issuer's immediate certificate
  - Define this as an object reference persistent URL (i.e persistent across re-issuance, but not against issuer re-key)

Certificate Pointers



Refinements to the Profile

- The AIA points to the Issuer's immediate certificate
  - Define this as an object reference persistent URL (i.e persistent across re-issuance, but not against issuer re-key)
- End Entity (no-CA) Certificates are used as one-off signing certificates
  - EE cert can be used for a single signing
  - Private key is destroyed after a single use
  - EE Cert SIA is a pointer to the object that has been signed with the corresponding private key
  - Signed object validity and resource attributes are controlled by the associated EE certificate(s)

End Entity Certificates

1. Generate Key Pair

2. Generate EE Cert of public key

3. Attach EE Cert to Document

4. Sign with Private Key

5. Destroy Private Key

6. Revoke Signature by revoking EE Cert

Refinements to the Profile

- The AIA points to the Issuer's immediate certificate
  - Define this as an object reference persistent URL (i.e persistent across re-issuance, but not against issuer re-key)
- End Entity (no-CA) Certificates are used as one-off signing certificates
  - EE cert can be used for a single signing
  - Private key is destroyed after a single use
  - EE Cert SIA is a pointer to the object that has been signed with the corresponding private key
  - Signed object validity and resource attributes are controlled by the associated EE certificate(s)
- Add the "Security Considerations" section text!

Review Comments

- Examples of Use of Resource Certificates?
- Example case of a subordinate certificate have a longer validity period than the superior certificate?
- Is the key size “SHOULD” a minimum or an absolute size?
- For Signature Algorithm should SHA-384 and SHA-512 be allowed options? Or should this be documented in a CP?
- Why specify RSYNC access as a “MUST” URI form? What is the normative language here?

Next Steps

- Generate an -03 version post IETF 67
- Request WG chair for Last Call on this document