SIDR
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2009

G. Huston
G. Michaelson
R. Loomans
APNIC

October 26, 2008

A Profile for X.509 PKIX Resource Certificates
draft-ietf-sidr-res-certs-14

Status of this Memo

Abstract

This document defines a standard profile for X.509 certificates for the purposes of supporting validation of assertions of "right-of-use" of an Internet Number Resource (IP Addresses and Autonomous System Numbers). This profile is used to convey the issuer's authorization of the subject to be regarded as the current holder of a "right-of-use" of the IP addresses and AS numbers that are described in the issued certificate.

Table of Contents

1. Introduction 4
1.1. Terminology 5
2. Describing Resources in Certificates 5
3. Resource Certificate Fields 6
3.1. Version 6
3.2. Serial number 6
3.3. Signature Algorithm 7
3.4. Issuer 7
3.5. Subject 7
3.6. Valid From 7
3.7. Valid To 8
3.8. Subject Public Key Info 8
3.9. Resource Certificate Version 3 Extension Fields 8
3.9.1. Basic Constraints 9
3.9.2. Subject Key Identifier 9
3.9.3. Authority Key Identifier 9
3.9.4. Key Usage 10
3.9.5. CRL Distribution Points 10
3.9.6. Authority Information Access 11
3.9.7. Subject Information Access 11
3.9.8. Certificate Policies 13
3.9.9. IP Resources 13
3.9.10. AS Resources 13
4. Resource Certificate Revocation List Profile 14
4.1. Version 14
4.2. Issuer Name 14
4.3. This Update 14
4.4. Next Update 14
4.5. Signature 15
4.6. Revoked Certificate List 15
4.6.1. Serial Number 15
4.6.2. Revocation Date 15
4.7. CRL Extensions 15
4.7.1. Authority Key Identifier 15
4.7.2. CRL Number 16
5. Resource Certificate Request Profile 16
5.1. PCKS#10 Profile 16
5.1.1. PKCS#10 Resource Certificate Request Template Fields 16
5.2. CRMF Profile 17
5.2.1. CRMF Resource Certificate Request Template Fields . . 18

5.2.2. Resource Certificate Request Control Fields 19
5.3. Certificate Extension Attributes in Certificate Requests 19
6. Resource Certificate Validation 21
6.1. Resource Extension Validation 22

6.2. Resource Certification Path Validation 23
6.3. Trust Anchors for Resource Certificates 24
6.3.1. Distribution Format of Default Trust Anchor Material 25
7. Design Notes 28
8. Security Considerations 31
9. IANA Considerations 31
10. Acknowledgements 31
11. References 32
11.1. Normative References 32
11.2. Informative References 32
Appendix A. Example Resource Certificate 33
Appendix B. Example Certificate Revocation List 35
Appendix C. Cryptographic Message Syntax Profile for RPKI Trust Anchor Material 37
C.1. Signed-Data ContentType 37
C.1.1. encapContentInfo 38
C.1.2. signerInfos 39
C.2. RTA Validation 41
Authors' Addresses 42
Intellectual Property and Copyright Statements 44

1. Introduction

This document defines a standard profile for X.509 certificates [X.509] for use in the context of certification of IP Addresses and AS Numbers. Such certificates are termed here "Resource Certificates." Resource Certificates are X.509 certificates that conform to the PKIX profile [RFC5280], and also conform to the constraints specified in this profile. Resource Certificates attest that the issuer has granted the subject a "right-of-use" for a listed set of IP addresses and Autonomous System numbers.

A Resource Certificate describes an action by a certificate issuer that binds a list of IP Address blocks and AS Numbers to the subject of the issued certificate. The binding is identified by the association of the subject's private key with the subject's public key contained in the Resource Certificate, as signed by the private key of the certificate's issuer.

In the context of the public Internet, and the use of public number resources within this context, it is intended that Resource Certificates are used in a manner that is explicitly aligned to the public number resource distribution function. Specifically, when a number resource is allocated or assigned by a number registry to an entity, this allocation is described by an associated Resource Certificate. This certificate is issued by the number registry, and the subject public key that is certified by the issuer corresponds to the public part of a key pair for which the private key is associated with the entity who is the recipient of the number assignment or allocation. A critical extension to the certificate enumerates the

IP Resources that were allocated or assigned by the issuer to the entity. In the context of the public number distribution function, this corresponds to a hierarchical PKI structure, where Resource Certificates are issued in only one 'direction' and there is a unique path of certificates from a certification authority operating at the apex of a resource distribution hierarchy to a valid certificate.

Validation of a Resource Certificate in such a hierarchical PKI can be undertaken by establishing a valid issuer-subject certificate chain from a certificate issued by a trust anchor certification authority to the certificate [RFC4158], with the additional constraint of ensuring that each subject's listed resources are fully encompassed by those of the issuer at each step in the issuer-subject certificate chain. Validation therefore logically corresponds to validation of an associated set of assignment or allocation actions of IP number resources.

While this profile describes the structure of a default Trust Anchor for this PKI, Relying Parties (RPs) in this PKI are free to select

the trust anchors upon which they rely, and thus the PKI as viewed by RPs need not match the public resource allocation hierarchy as described here.

Resource Certificates may be used in the context of the operation of secure inter-domain routing protocols to convey a right-of-use of an IP number resource that is being passed within the routing protocol, allowing relying parties to verify legitimacy and correctness of routing information. Related use contexts include validation of Internet Routing Registry objects, validation of routing requests, and detection of potential unauthorized use of IP addresses.

This profile defines those fields that are used in a Resource Certificate that MUST be present for the certificate to be valid. Relying Parties SHOULD check that a Resource Certificate conforms to this profile as a requisite for validation of a Resource Certificate.

1.1. Terminology

It is assumed that the reader is familiar with the terms and concepts described in "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [RFC5280], "X.509

Extensions for IP Addresses and AS Identifiers" [RFC3779], "Internet Protocol" [RFC0791], "Internet Protocol Version 6 (IPv6) Addressing Architecture" [RFC4291], "Internet Registry IP Allocation Guidelines" [RFC2050], and related regional Internet registry address management policy documents.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Describing Resources in Certificates

The framework for describing an association between the subject of a certificate and the resources currently under the subject's control is described in [RFC3779].

There are three aspects of this resource extension that are noted in this profile:

1. RFC 3779 notes that a resource extension SHOULD be a CRITICAL extension to the X.509 Certificate. This Resource Certificate profile further specifies that the use of this certificate extension MUST be used in all Resource Certificates and MUST be marked as CRITICAL.

2. RFC 3779 defines a sorted canonical form of describing a resource set, with maximal spanning ranges and maximal spanning prefix masks as appropriate. All valid certificates in this profile MUST use this sorted canonical form of resource description in the resource extension field.

3. A test of the resource extension in the context of certificate validity includes the condition that the resources described in the immediate parent CA certificate in the PKI (the certificate where this certificate's issuer is the subject) has a resource set (called here the "issuer's resource set") that MUST encompass the resource set of the issued certificate. In this context "encompass" allows for the issuer's resource set to be the same as, or a strict superset of, any subject's resource set.

Certificate validation entails the construction of a sequence of valid certificates in an issuer-subject chain (where the subject field of one certificate appears as the issuer in the next certificate in the sequence) from a trust anchor to the certificate being validated. Moreover, the resource extensions in this certificate sequence from the first CA under the trust anchor to the certificate being validated form a sequence of encompassing relationships in terms of the resources described in the resource extension.

3. Resource Certificate Fields

A Resource Certificate is a valid X.509 v3 public key certificate, consistent with the PKIX profile [RFC5280], containing the fields listed in this section. Unless specifically noted as being OPTIONAL, all the fields listed here MUST be present, and any other field MUST NOT appear in a conforming Resource Certificate. Where a field value is specified here this value MUST be used in conforming Resource Certificates.

3.1. Version

Resource Certificates are X.509 Version 3 certificates. This field MUST be present, and the Version MUST be 3 (i.e. the value of this field is 2).

3.2. Serial number

The serial number value is a positive integer that is unique per Issuer.

3.3. Signature Algorithm

This field describes the algorithm used to compute the signature on this certificate. This profile specifies a default of SHA-256 with RSA (sha256WithRSAEncryption), and allows for the use of SHA-384 or SHA-512. Accordingly, the value for this field MUST be one of the OID values { pkcs-1 11 }, { pkcs-1 12 } or { pkcs-1 13 } [RFC4055].

3.4. Issuer

This field identifies the entity that has signed and issued the certificate. The value of this field is a valid X.501 name. Conventions are imposed on Issuer names used in resource certificates, as described in [ID.sidr-arch].

If the certificate is a subordinate certificate issued by virtue of the "cA" bit set in the immediate superior certificate, then the issuer name MUST correspond to the subject name as contained in the immediate superior certificate.

3.5. Subject

This field identifies the entity to whom the resource has been allocated / assigned. The value of this field is a valid X.501 name. As noted above, conventions are imposed on Subject names used in resource certificates, as described in [ID.sidr-arch].

In this profile the subject name is determined by the issuer, and each distinct subordinate CA and EE certified by the issuer MUST be identified using a subject name that is unique per issuer.

In this context "distinct" is defined as an entity and a given public key. An issuer SHOULD use a different subject name if the subject entity or the subject entity's key pair has changed.

3.6. Valid From

The starting time at which point the certificate is valid. In this profile the "Valid From" time SHOULD be no earlier than the time of certificate generation. As per Section 4.1.2.5 of [RFC5280], Certification Authorities (CAs) conforming to this profile MUST always encode the certificate's "Valid From" date through the year 2049 as UTCTime, and dates in 2050 or later MUST be encoded as GeneralizedTime. These two time formats are defined in [RFC5280].

In this profile, it is valid for a certificate to have a value for this field that pre-dates the same field value in any superior certificate. Relying Parties should not attempt to infer from this

time information a certificate was valid at a time in the past, or

will be valid at a time in the future, as the validity of a certificate refers to validity at the current time.

3.7. Valid To

The Valid To time is the date and time at which point in time the certificate's validity ends. It represents the anticipated lifetime of the resource allocation / assignment arrangement between the issuer and the subject. As per Section 4.1.2.5 of [RFC5280], CAs conforming to this profile MUST always encode the certificate's "Valid To" date through the year 2049 as UTCTime, and dates in 2050 or later MUST be encoded as GeneralizedTime. These two time formats are defined in [RFC5280].

As noted above, it is valid for a certificate to have a value for this field that post-dates the same field value in any superior certificate. The same caveats apply to Relying Party's assumptions relating to the certificate's validity at any time other than the current time,

While a CA is typically advised against issuing a certificate with a validity interval that exceeds the validity interval of the CA's certificate that will be used to validate the issued certificate, in the context of this profile, it is anticipated that a CA may have valid grounds to issue a certificate with a validity interval that exceeds the validity interval of its certificate.

3.8. Subject Public Key Info

This field specifies the subject's public key and the algorithm with which the key is used. The public key algorithm MUST be RSA, and, accordingly, the OID for the public key algorithm is 1.2.840.113549.1.1.1. The key size MUST be a minimum size of 2048 bits.

It is noted that larger key sizes are computationally expensive for both the CA and relying parties, indicating that care should be taken when deciding to use larger than the minimum key size noted above.

3.9. Resource Certificate Version 3 Extension Fields

As noted in Section 4.2 of [RFC5280], each extension in a certificate is designated as either critical or non-critical. A certificate-using system MUST reject the certificate if it encounters a critical extension it does not recognize; however, a non-critical extension MAY be ignored if it is not recognized [RFC5280].

The following X.509 V3 extensions MUST be present in a conforming Resource Certificate, except where explicitly noted otherwise.

3.9.1. Basic Constraints

The basic constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate.

The issuer determines whether the "cA" boolean is set. If this bit is set, then it indicates that the subject is allowed to issue resources certificates within this overall framework (i.e. the subject is a CA).

The Path Length Constraint is not specified in this profile and MUST NOT be present.

The Basic Constraints extension field is a critical extension in the Resource Certificate profile, and MUST be present when the subject is a CA, and MUST NOT be present otherwise.

3.9.2. Subject Key Identifier

The subject key identifier extension provides a means of identifying certificates that contain a particular public key. To facilitate certification path construction, this extension MUST appear in all Resource Certificates. This extension is non-critical.

The value of the subject key identifier MUST be the value placed in the key identifier field of the Authority Key Identifier extension of all certificates issued by this subject.

The Key Identifier used here is the 160-bit SHA-1 hash of the value of the DER-encoded ASN.1 bit string of the subject public key, as described in Section 4.2.1.2 of [RFC5280].

3.9.3. Authority Key Identifier

The authority key identifier extension provides a means of identifying certificates that are signed by the issuer's private key, by providing a hash value of the issuer's public key. To facilitate path construction, this extension MUST appear in all Resource Certificates. The keyIdentifier MUST be present in all Resource Certificates, with the exception of a CA who issues a "self-signed" certificate. The authorityCertIssuer and authorityCertSerialNumber fields MUST NOT be present. This extension is non-critical.

The Key Identifier used here is the 160-bit SHA-1 hash of the value

of the DER-encoded ASN.1 bit string of the issuer's public key, as described in Section 4.2.1.1 of [RFC5280].

3.9.4. Key Usage

This describes the purpose of the certificate. This is a critical extension, and it MUST be present.

In certificates issued to Certification Authorities only the keyCertSign and CRLSign bits are set to TRUE and these MUST be the only bits set to TRUE.

In end-entity certificates the digitalSignature bit MUST be set to TRUE and MUST be the only bit set to TRUE.

3.9.5. CRL Distribution Points

This field (CRLDP) identifies the location(s) of the CRL(s) associated with certificates issued by this Issuer. This profile uses the URI form of object identification. The preferred URI access mechanism is a single RSYNC URI ("rsync://") [rsync] that references a single inclusive CRL for each issuer.

In this profile the certificate issuer is also the CRL issuer, implying at the CRLIssuer field MUST be omitted, and the distributionPoint field MUST be present. The Reasons field MUST be omitted.

The distributionPoint MUST contain general names, and MUST NOT contain a nameRelativeToCRLIssuer. The type of the general name MUST be of type URI.

In this profile, the scope of the CRL is specified to be all certificates issued by this CA issuer.

The sequence of distributionPoint values MUST contain only a single DistributionPointName set. The DistributionPointName set MAY contain more than one URI value. An RSYNC URI MUST be present in the DistributionPointName set, and reference the most recent instance of

this issuer's certificate revocation list. Other access form URIs MAY be used in addition to the RSYNC URI.

This extension MUST be present and it is non-critical. There is one exception, namely where a CA distributes its public key in the form of a "self-signed" certificate, the CRLDP MUST be omitted.

3.9.6. Authority Information Access

This extension (AIA) identifies the point of publication of the certificate that is issued by the issuer's immediate superior CA, where this certificate's issuer is the subject. In this profile a single reference object to publication location of the immediate superior certificate MUST be used, except in the case where a CA distributes its public key in the form of a "self-signed" certificate, in which case the AIA field SHOULD be omitted.

This profile uses a URI form of object identification. The preferred URI access mechanisms is "rsync", and an RSYNC URI MUST be specified with an accessMethod value of id-ad-caIssuers. The URI MUST reference the point of publication of the certificate where this issuer is the subject (the issuer's immediate superior certificate). Other access method URIs referencing the same object MAY also be included in the value sequence of this extension.

When an Issuer re-issues a CA certificate, the subordinate certificates need to reference this new certificate via the AIA field. In order to avoid the situation where a certificate re-issuance necessarily implies a requirement to re-issue all subordinate certificates, CA Certificate issuers SHOULD use a persistent URL name scheme for issued certificates. This implies that re-issued certificates overwrite previously issued certificates to the same subject in the publication repository, and use the same publication name as previously issued certificates. In this way subordinate certificates can maintain a constant AIA field value and need not be re-issued due solely to a re-issue of the superior certificate. The issuers' policy with respect to the persistence of name objects of issued certificates MUST be specified in the Issuer's Certification Practice Statement.

This extension is non-critical.

3.9.7. Subject Information Access

This extension (SIA) identifies the location of information and services relating to the subject of the certificate in which the SIA extension appears. Where the Subject is a CA in this profile, this information and service collection will include all current valid certificates that have been issued by this subject that are signed with the subject's corresponding private key.

This profile uses a URI form of location identification. The preferred URI access mechanism is "rsync", and an RSYNC URI MUST be specified, with an access method value of id-ad-caRepository when the subject of the certificate is a CA. The RSYNC URI MUST reference an

object collection rather than an individual object and MUST use a trailing '/' in the URI.

Other access method URIs that reference the same location MAY also be included in the value sequence of this extension. The ordering of URIs in this sequence reflect the subject's relative preferences for access methods, with the first method in the sequence being the most preferred.

This extension MUST be present when the subject is a CA, and is non-critical.

For End Entity (EE) certificates, where the subject is not a CA, this extension MAY be present, and is non-critical. If present, it either references the location where objects signed by the private key associated with the EE certificate can be accessed, or, in the case of single-use EE certificates it references the location of the single object that has been signed by the corresponding private key.

When the subject is an End Entity, and it publishes objects signed with the matching private key in a repository, the directory where these signed objects is published is referenced the id-ad-signedObjectRepository OID.

```
id-ad OBJECT IDENTIFIER ::= { id-pkix 48 }
```

```
id-ad-signedObjectRepository OBJECT IDENTIFIER ::= { id-ad 9 }
```

When the subject is an End Entity, and it publishes a single object signed with the matching private key, the location where this signed object is published is referenced the id-ad-signedObject OID.

```
id-ad-signedObject OBJECT IDENTIFIER ::= { id-ad 11 }
```

This profile requires the use of repository publication manifests [ID.sidr-manifests] to list all signed objects that are deposited in the repository publication point associated with a CA or an EE. The publication point of the manifest for a CA or EE is placed in the SIA extension of the CA or EE certificate. This profile uses a URI form of manifest identification for the accessLocation. The preferred URI access mechanisms is "rsync", and an RSYNC URI MUST be specified. Other accessDescription fields may exist with this id-ad-Manifest accessMethod, where the accessLocation value indicates alternate URI access mechanisms for the same manifest object.

```
id-ad-rpkiManifest  OBJECT IDENTIFIER ::= { id-ad 10 }
```

CA certificates MUST include in the SIA an accessMethod OID of id-ad-

rpkiManifest, where the associated accessLocation refers to the subject's published manifest object as an object URL.

When an EE certificate is intended for use in verifying multiple objects, EE certificate MUST include in the SIA an access method OID of id-ad-rpkiManifest, where the associated access location refers to the publication point of the objects that are verified using this EE certificate.

When an EE certificate is used to verify a single published object, the EE certificate MUST include in the SIA an access method OID of id-ad-signedObject, where the associated access location refers to the publication point of the single object that is verified using this EE certificate. In this case, the SIA MUST NOT include the access method OID of id-ad-rpkiManifest.

3.9.8. Certificate Policies

This extension MUST reference the Resource Certificate Policy, using the OID Policy Identifier value of "1.3.6.1.5.5.7.14.2". This field MUST be present and MUST contain only this value for Resource Certificates.

No PolicyQualifiers are defined for use with this policy and thus none must be included in this extension.

This extension MUST be present and it is critical.

3.9.9. IP Resources

This extension contains the list of IP address resources as per [RFC3779]. The value may specify the "inherit" element for a particular AFI value. In the context of resource certificates describing public number resources for use in the public Internet, the SAFI value MUST NOT be used. All Resource Certificates MUST include an IP Resources extension, an AS Resources extension, or both extensions.

This extension, if present, MUST be marked critical.

3.9.10. AS Resources

This extension contains the list of AS number resources as per [RFC3779], or may specify the "inherit" element. RDI values are NOT supported in this profile and MUST NOT be used. All Resource Certificates MUST include an IP Resources extension, an AS Resources extension, or both extensions.

This extension, if present, MUST be marked critical.

4. Resource Certificate Revocation List Profile

Each CA MUST issue a version 2 Certificate Revocation List (CRL), consistent with [RFC5280]. The CRL issuer is the CA, and no indirect CRLs are supported in this profile.

An entry MUST NOT be removed from the CRL until it appears on one regularly scheduled CRL issued beyond the revoked certificate's

validity period, as required in [RFC5280].

This profile does not allow issuance of Delta CRLs.

The scope of the CRL MUST be "all certificates issued by this CA". The contents of the CRL are a list of all non-expired certificates that have been revoked by the CA.

No CRL fields other than those listed here are permitted in CRLs issued under this profile. Unless otherwise indicated, these fields MUST be present in the CRL. Where two or more CRLs issued by a single CA with the same scope, the CRL with the highest value of the "CRL Number" field supersedes all other CRLs issued by this CA.

4.1. Version

Resource Certificate Revocation Lists are Version 2 certificates (the integer value of this field is 1).

4.2. Issuer Name

The value of this field is the X.501 name of the issuing CA who is also the signer of the CRL, and is identical to the Issuer name in the Resource Certificates that are issued by this issuer.

4.3. This Update

This field contains the date and time that this CRL was issued. The value of this field MUST be encoded as UTCTime for dates through the year 2049, and MUST be encoded as GeneralizedTime for dates in the year 2050 or later.

4.4. Next Update

This is the date and time by which the next CRL SHOULD be issued. The value of this field MUST be encoded as UTCTime for dates through the year 2049, and MUST be encoded as GeneralizedTime for dates in

the year 2050 or later.

4.5. Signature

This field contains the algorithm used to sign this CRL. This profile specifies a default of SHA-256 with RSA (sha256WithRSAEncryption), and allows for the use of SHA-384 or SHA-512.

It is noted that larger key sizes are computationally expensive for both the CRL Issuer and relying parties, indicating that care should be taken when deciding to use larger than the default key size.

4.6. Revoked Certificate List

When there are no revoked certificates, then the revoked certificate list MUST be absent.

For each revoked resource certificate only the following fields MUST be present. No CRL entry extensions are supported in this profile, and CRL entry extensions MUST NOT be present in a CRL.

4.6.1. Serial Number

The serial number of the revoked certificate.

4.6.2. Revocation Date

The time the certificate was revoked. This time MUST NOT be a future date (i.e., a date later than ThisUpdate). The value of this field MUST be encoded as UTCTime for dates through the year 2049, and MUST be encoded as GeneralizedTime for dates in the year 2050 or later.

4.7. CRL Extensions

The X.509 v2 CRL format allows extensions to be placed in a CRL. The following extensions are supported in this profile, and MUST be present in a CRL.

4.7.1. Authority Key Identifier

The authority key identifier extension provides a means of identifying the public key corresponding to the private key used to sign a CRL. Conforming CRL issuers MUST use the key identifier method. The syntax for this CRL extension is defined in section 4.2.1.1 of [RFC5280].

This extension is non-critical.

4.7.2. CRL Number

The CRL Number extension conveys a monotonically increasing sequence number of positive integers for a given CA and scope. This extension allows users to easily determine when a particular CRL supersedes another CRL. The highest CRL Number value supersedes all other CRLs issued by the CA with the same scope.

This extension is non-critical.

5. Resource Certificate Request Profile

A resource certificate request MAY use either of PKCS#10 or Certificate Request Message Format (CRMF). A CA Issuer MUST support PKCS#10 and a CA Issuer may, with mutual consent of the subject, support CRMF.

5.1. PCKS#10 Profile

This profile refines the specification in [RFC2986], as it relates to Resource Certificates. A Certificate Request Message object, formatted according to PKCS#10, is passed to a CA as the initial step in issuing a certificate.

This request may be conveyed to the CA via a Registration Authority (RA), acting under the direction of a Subject.

With the exception of the public key related fields, the CA is permitted to alter any requested field when issuing a corresponding certificate.

5.1.1. PKCS#10 Resource Certificate Request Template Fields

This profile applies the following additional constraints to fields that may appear in a CertificationRequestInfo:

Version
: This field is mandatory and MUST have the value 0.

Subject
: This field is optional. If present, the value of this field SHOULD be empty, in which case the issuer MUST generate a subject name that is unique in the context of certificates issued by this issuer. If the value of this field is non-empty, then the CA MAY consider the value of this field as the

subject's suggested subject name, but the CA is NOT bound to

honor this suggestion, as the subject name MUST be unique per subordinate CA and EE in certificates issued by this issuer.

SubjectPublicKeyInfo
This field specifies the subject's public key and the algorithm with which the key is used. The public key algorithm MUST be RSA, and the OID for the algorithm is 1.2.840.113549.1.1.1. This field also includes a bit-string representation of the entity's public key. For the RSA public-key algorithm the bit string contains the DER encoding of a value of PKCS #1 type RSAPublicKey.

Attributes
[RFC2986] defines the attributes field as key-value pairs where the key is an OID and the value's structure depends on the key.

The only attribute used in this profile is the ExtensionRequest attribute as defined in [RFC2985]. This attribute contains X509v3 Certificate Extensions. The profile for extensions in certificate requests is specified in Section 5.3.

This profile applies the following additional constraints to fields that MAY appear in a CertificationRequest Object:

signatureAlgorithm
This profile specifies a default of SHA-256 with RSA (sha256WithRSAEncryption), and allows for the use of SHA-384 or SHA-512. Accordingly, the value for this field MUST be one of the OID values { pkcs-1 11 }, { pkcs-1 12 } or { pkcs-1 13 } [RFC4055].

It is noted that larger key sizes are computationally expensive for both the CA and relying parties, indicating that care should be taken when deciding to use larger than the default key size.

5.2. CRMF Profile

This profile refines the Certificate Request Message Format (CRMF)

specification in [RFC4211], as it relates to Resource Certificates. A Certificate Request Message object, formatted according to the CRMF, is passed to a CA as the initial step in issuing a certificate.

This request MAY be conveyed to the CA via a Registration Authority (RA), acting under the direction of a subject.

With the exception of the public key related fields, the CA is

permitted to alter any requested field when issuing a corresponding certificate.

5.2.1. CRMF Resource Certificate Request Template Fields

This profile applies the following additional constraints to fields that may appear in a Certificate Request Template:

Version
: This field MAY be absent, or MAY specify the request of a Version 3 Certificate. It SHOULD be omitted.

SerialNumber
: As per [RFC4211], this field is assigned by the CA and MUST be omitted in this profile.

SigningAlgorithm
: As per [RFC4211], this field is assigned by the CA and MUST be omitted in this profile.

Issuer
: This field is assigned by the CA and MUST be omitted in this profile.

Validity
: This field MAY be omitted. If omitted, the CA will issue a Certificate with Validity dates as determined by the CA. If specified, then the CA MAY override the requested values with dates as determined by the CA.

Subject
: This field is optional. If present, the value of this field

SHOULD be empty, in which case the issuer MUST generate a subject name that is unique in the context of certificates issued by this issuer. If the value of this field is non-empty, then the CA MAY consider the value of this field as the subject's suggested subject name, but the CA is NOT bound to honor this suggestion, as the subject name MUST be unique per issuer in certificates issued by this issuer.

PublicKey
This field MUST be present.

extensions
This attribute contains X509v3 Certificate Extensions. The profile for extensions in certificate requests is specified in Section 5.3.

5.2.2. Resource Certificate Request Control Fields

The following control fields are supported in this profile:

Authenticator Control
It is noted that the intended model of authentication of the subject is a long term one, and the advice as offered in [RFC4211] is that the Authenticator Control field be used.

5.3. Certificate Extension Attributes in Certificate Requests

The following extensions MAY appear in a PKCS#10 or CRMF Certificate Request. Any other extensions MUST NOT appear in a Certificate Request. This profile places the following additional constraints on these extensions.:

BasicConstraints
If this is omitted then the CA will issue an end entity

certificate with the BasicConstraints extension not present in the issued certificate.

The Path Length Constraint is not supported in this Resource Certificate Profile, and this field MUST be omitted in this profile.

The CA MAY honor the SubjectType CA bit set to on. If this bit is set, then it indicates that the Subject is allowed to issue resource certificates within this overall framework.

The CA MUST honor the SubjectType CA bit set to off (End Entity certificate request), in which case the corresponding end entity certificate will not contain a BasicConstraints extension.

SubjectKeyIdentifier
This field is assigned by the CA and MUST be omitted in this profile.

AuthorityKeyIdentifier
This field is assigned by the CA and MUST be omitted in this profile.

KeyUsage
The CA MAY honor KeyUsage extensions of keyCertSign and cRLSign if present, as long as this is consistent with the BasicConstraints SubjectType sub field, when specified.

SubjectInformationAccess
This field MUST be present when the subject is a CA, and the field value SHOULD be honored by the CA. If the CA is not able to honor the requested field value, then the CA MUST reject the Certificate Request.

This field (SIA) identifies the location of information and services relating to the subject of the certificate in which the SIA extension appears.

Where the subject is a CA in this profile, this information and service collection will include all current valid certificates that have been issued by this subject that are signed with the subject's corresponding private key.

This profile uses a URI form of location identification. An RSYNC URI MUST be specified, with an access method value of id-ad-caRepository when the subject of the certificate is a CA. The RSYNC URI MUST reference an object collection rather than an individual object and MUST use a trailing '/' in the URI. Other access method URIs that reference the same location MAY also be included in the value sequence of this extension. The ordering of URIs in this sequence reflect the subject's relative preferences for access methods, with the first method in the sequence being the most preferred by the Subject.

A request for a CA certificate MUST include in the SIA of the request the id-ad-caRepository access method, and also MUST include in the SIA of the request the accessMethod OID of id-ad-rpkiManifest, where the associated accessLocation refers to the subject's published manifest object as an object URL.

This field MAY be present when the subject is a EE. If it is present the field value SHOULD be honored by the CA. If the CA is not able to honor the requested field value, then the CA MUST reject the Certificate Request. If it is not present the CA SHOULD honor this request and omit the SIA from the issued certificate. If the CA is not able to honor the request to omit the SIA, then the CA MUST reject the Certificate Request.

When an EE certificate is intended for use in verifying multiple objects, the certificate request for the EE certificate MUST include in the SIA of the request an access method OID of id-ad-signedObjectRepository, and also MUST include in the SIA of the request an access method OID of id-ad-rpkiManifest, where the associated access location refers to the publication point of the objects that are verified using this EE certificate.

When an EE certificate is used to sign a single published object, the certificate request for the EE certificate MUST include in the SIA of the request an access method OID of id-

ad-signedObject, where the associated access location refers to the publication point of the single object that is verified using this EE certificate, and MUST NOT include an id-ad-rpkiManifest access method OID in the SIA of the request.

In the case when the EE certificate is to be used exclusively to sign one or more unpublished objects, such that the all signed objects will not be published in any RPKI repository, then the SIA SHOULD be omitted from the request.

CRLDistributionPoints
This field is assigned by the CA and MUST be omitted in this profile.

AuthorityInformationAccess
This field is assigned by the CA and MUST be omitted in this profile.

CertificatePolicies
This field is assigned by the CA and MUST be omitted in this profile.

With the exceptions of the publicKey field and the SubjectInformationAccess field, the CA is permitted to alter any requested field.

6. Resource Certificate Validation

This section describes the Resource Certificate validation procedure. This refines the generic procedure described in section 6 of [RFC5280].

To meet this goal, the path validation process verifies, among other things, that a prospective certification path (a sequence of n

certificates) satisfies the following conditions:

1. for all x in {1, ..., n-1}, the subject of certificate x is the issuer of certificate x+1;

2. certificate 1 is issued by a trust anchor (Note that a trust anchor is NOT a resource certificate in this context and thus does not contain RFC 3779 extensions.);

3. certificate n is the certificate to be validated; and

4. for all x in {1, ..., n}, the certificate is valid.

6.1. Resource Extension Validation

The IP resource extension definition [RFC3779] defines a critical extensions for Internet number resources. These are ASN.1 encoded representations of the IPv4 and IPv6 address range (either as a prefix/length, or start-end pair) and the AS number set.

Valid Resource Certificates MUST have a valid IP address and/or AS number resource extension. In order to validate a Resource Certificate the resource extension MUST also be validated. This validation process relies on definitions of comparison of resource sets:

more specific
: Given two IP address or AS number contiguous ranges, A and B, A is "more specific" than B if range B includes all IP addresses or AS numbers described by range A, and if range B is larger than range A.

equal
: Given two IP address or AS number contiguous ranges, A and B, A is "equal" to B if range A describes precisely the same collection of IP addresses or AS numbers as described by range B. The definition of "inheritance" in [RFC3779] is equivalent to this "equality" comparison.

encompass
: Given two IP address and AS number sets X and Y, X "encompasses" Y if, for every contiguous range of IP addresses or AS numbers elements in set Y, the range element is either more specific than or equal to a contiguous range element within the set X.

Validation of a certificate's resource extension in the context of an ordered certificate sequence of {1,2, ... , n} where '1' is issued by a trust anchor and 'n' is the target certificate, and where the subject of certificate 'x' is the issuer of certificate 'x' + 1, implies that the resources described in certificate 'x' "encompass" the resources described in certificate 'x' + 1, and the resources described in the trust anchor information "encompass" the resources described in certificate 1.

6.2. Resource Certification Path Validation

Validation of signed resource data using a target resource certificate consists of assembling an ordered sequence (or 'Certification Path') of certificates ({1,2,...n} where '1' is a certificate that has been issued by a trust anchor, and 'n' is the target certificate) verifying that all of the following conditions hold:

1. The certificate can be verified using the Issuer's public key and the signature algorithm

2. The current time lies within the certificate's Validity From and To values.

3. The certificate contains all fields that MUST be present and contains field values as specified in this profile for all field values that MUST be present.

4. No field value that MUST NOT be present in this profile is present in the certificate.

5. The Issuer has not revoked the certificate by placing the certificate's serial number on the Issuer's current Certificate Revocation List, and the Certificate Revocation List is itself valid.

6. That the resource extension data is "encompassed" by the resource extension data contained in a valid certificate where this Issuer is the Subject (the previous certificate in the ordered sequence)

7. The Certification Path originates with a certificate issued by a trust anchor, and there exists a signing chain across the Certification Path where the Subject of Certificate x in the Certification Path matches the Issuer in Certificate x+1 in the Certification Path.

A certificate validation algorithm may perform these tests in any chosen order.

Certificates and CRLs used in this process may be found in a locally maintained cache, maintained by a regular synchronization across the distributed publication repository structure.

There exists the possibility of encountering certificate paths that are arbitrarily long, or attempting to generate paths with loops as means of creating a potential DOS attack on a relying party. Some further heuristics may be required to halt the certification path validation process in order to avoid some of the issues associated with attempts to validate such structures. It is suggested that implementations of Resource Certificate validation MAY halt with a validation failure if the certification path length exceeds a pre-determined configuration parameter.

6.3. Trust Anchors for Resource Certificates

The default trust model for the resource certificate PKI maps to the extant public resource allocation system, comprised of IANA, RIRs, NIRs (in some regions) and LIRs. This is a strict hierarchy, in that any number resource and a corresponding recipient entity has only one 'parent' issuing registry for that resource. Moreover, the issuing registry is not a direct or indirect subordinate recipient entity of the recipient entity in question (i.e., there are no loops in the model).

Nonetheless, as in any PKI, selection of one or more entities as trust anchor is a task undertaken by each relying party. The structure of the resource certificate profile admits the same variety of trust models as PKIX (and X.509) standards. There is only one additional caveat on the general applicability of trust models, namely that in forming a validation path to a CA immediately below a trust anchor, the sequence of certificates MUST preserve the resource extension validation property, as described in Section 6.1. Section 6.1.

The trust anchor information, describing a CA that serves as a trust anchor, includes the following:

1. the trust anchor name,

2. the public key algorithm,

3. the trust anchor's public key, and

4. optionally, parameters associated with the public key.

The trust anchor information may be provided to the path processing procedure in the form of a self-signed certificate.

6.3.1. Distribution Format of Default Trust Anchor Material

In the RPKI, the certificate framework corresponds to the hierarchies of the resource distribution function. In consideration of this, it is reasonable to nominate to relying parties a default set of trust anchors for the RPKI that correspond to the entities who operate at the upper levels of the associated resource allocation hierarchy. The corresponding nominated trust anchor CA(s) should therefore map, in some fashion, to apex point(s) of the hierarchical resource distribution structure.

The characteristics of a trust anchor framework for the RPKI includes the following considerations:

* The entity or entities that issue proposed trust anchor material for the RPKI should be as close as possible to the apex of the associated resource distribution hierarchy.

* Such trust anchor material SHOULD be long-lived. As it can be reasonably anticipated that default trust anchor material would be distributed with relying party validation software, the implication is that the distributed default trust anchor material SHOULD remain constant for extended time intervals.

* It is a poor trust model when any entity that issues putative trust anchor material claims to be authoritative for information or actions of which the entity has no direct

knowledge, nor is in possession of a current definitive record of such actions. Entities who propose themselves in a role of a trust anchor issuer SHOULD be able to point to corroborative material supporting the assertion that they are legitimate authorities for the information for which they are representing themselves as a trust anchor for relying parties.

An entity offering itself as a putative trust anchor for a part of the RPKI is required to regularly publish an RPKI CA certificate at a stable URL, and to publish this URL as distributed trust anchor material, as follows:

* The entity issues an RPKI self-signed "root" CA certificate that is used as the apex of an RPKI certificate validation tree. This certificate MUST meet all of the criteria established in Section 3 of this document for a self-signed RPKI certificate. This certificate MUST be reissued periodically, prior to its expiration, and MUST be reissued upon any change in the resource set that has been allocated to the entity operating this CA. The validity interval of this certificate SHOULD reflect the anticipated period of changes to the entity's resource set .

* The entity maintains a trust anchor key pair that is distinct from the key pair represented in the self-signed RPKI CA noted above.

* The entity issues a self-signed CA certificate [RFC5280] (that contains no RFC 3779 extension) where the subject public key in the certificate is the public key of the trust anchor and the certificate is signed using the corresponding private key of trust anchor key pair. This is called the TA CA certificate. This certificate MUST have the keyCertSign sign bit set in the key usage extension, and the CA flag set in the basic constraints extension, no AIA value and no CRLDP value. The validity period of this certificate should be very long, as is the norm for trust anchor material. The SIA of this certificate references a publication point where the CRL and the CMS structure defined below are published.

* The trust anchor (TA) issues an EE certificate (a TA EE certificate) with a validity period identical to the validity period of its RPKI self-signed "root" CA certificate. This EE certificate MUST have the digitalSignature bit set, and this MUST be the only bit set to TRUE. There is no BasicConstraints extension in this certificate. The validity period of this TA EE certificate SHOULD be aligned to the validity period of the RPKI self-signed "root" CA certificate.

* The TA CA regularly issues a CRL. The CRL issuance cycle SHOULD be shorter than the validity period for the RPKI self-signed "root" certificate.

* Each time the RPKI self-signed "root" certificate is re-issued, or prior to the expiration of the TA EE certificate, the TA generates a Cryptographic Message Syntax (CMS) [RFC3852] signed-data object, the payload of which is an RPKI self-signed "root" certificate. The object is CMS-signed with the private key of the TA EE certificate. The TA EE certificate is

included as a CMS signed attribute in the CMS object. The TA CA certificate and the associated CRL are not to be included in the CMS object. The format of the CMS object is specified in Appendix C. The CMS object is published at the location referenced in the SIA of the TA CA certificate.

* The entity publicly distributes the TA CA certificate as its trust anchor material, in an out-of-band fashion, e.g., as part of widely distributed relying party software.

Relying Parties can assemble the default trust anchor collection by using the TA CA certificate for each nominated trust anchor:

* The TA CA's CRL and CMS objects can be retrieved from the publication point referenced by the SIA in the TA CA certificate.

* The CRL can be verified against the TA CA certificate.

* The CMS signature can be verified using the included TA EE certificate together with the retrieved CRL and the self-signed

TA CA certificate.

* The relying party can then load the enclosed RPKI self-signed CA certificate as a trust anchor for RPKI validation for those resources described in the IP Resource extensions [RFC3779] of this RPKI certificate.

Relying Parties SHOULD perform this retrieval and validation operation at intervals no less frequent than the nextUpdate time of the published TA CA CRL, and SHOULD perform the retrieval operation prior to the expiration of the TA EE certificate, or upon revocation of the TA EE certificate that is used to verify the CMS object that holds the trust anchor's current RPKI self-signed CA certificate.

If a trust anchor chooses to reissue its RPKI self- signed CA certificate before the expiration of that certificate, it MUST perform the follow actions: revise the nextUpdate time of the TA CA's CRL to reflect the issue date for the new TA EE certificate, issue a new TA EE certificate and a new CMS object with the new RPKI self-signed CA certificate, and revoke the old TA EE certificate at the nextUpdate time in the next issued CRL. This revocation will provide an indication to relying parties to perform the retrieval operation of the RPKI self-signed CA certificate at a time earlier than the normal update cycle time. .

7. Design Notes

The following notes provide some additional commentary on the considerations that lie behind some of the design choices that were made in the design of this certificate profile. These notes do not constitute a formal part of the profile specification, and the interpretation of key words as defined in RFC2119 are not applicable in this section of the document.

Certificate Extensions:
: This profile does not permit the use of any other critical or non-critical extensions. The rationale for this restriction is that the resource certificate profile is intended for a specific use, and in this context it is not seen as being

appropriate to be in the position of having certificates with additional non-critical extensions that relying parties may see as valid certificates without understanding the extensions, but were the relying party in a position to understand the extensions, would contradict or qualify in some way this original judgment of validity. This profile takes the position of minimalism over extensibility. The specific goal for the associated Resource Public Key Infrastructure to precisely match the IP number resource allocation structure through an aligned certificate structure that describes the allocation and its context within the number resource distribution hierarchy. The profile defines a resource certificate that is structured to meet these requirements.

Certification Authorities and Key Values:

This profile uses a definition of an instance of a CA as a combination of a named entity and a key pair. Within this definition a CA instance cannot rollover a key pair. However, the entity can generate a new instance of a CA with a new key pair and roll over all the signed subordinate products to the new CA.

This has a number of implications in terms of subject name management, CRL Scope and repository publication point management.

Subject Name:

For Subject Names the issuer should ensure that when an entity requests a certificate with a new key pair, the CA issues a certificate with a new subject name. One way to achieve this is for the issuer to use a mapping of the hash of the subject public key value into a character string for a CommonName that becomes the CA Subject Name.

CRL Scope:

For CRL Scope this profile specifies that a CA issues a single CRL sequence, and the scope of the CRL is all certificates issued by this CA. Because the CA instance is bound to a single key pair this implies that the CA's public key, the key used to validate the CA's CRL, and the key used to validate the certificates revoked by that

CRL are all the same.

Repository Publication Point:

The definition of a CA affects the design of the repository publication system. In order to minimize the amount of forced re-certification on key rollover events, a repository publication regime that uses the same repository publication point for all CA instances that refers to the same entity, but with different key values will minimize the extent of re-generation of certificates to only immediate subordinate certificates.

In order for two or more CA instances to share a single repository publication point there needs to be a regime of key management into OLD, CURRENT and FUTURE keys and a similar regime of OLD, CURRENT and FUTURE CAs. An OLD CA should regularly publish its CRL for as long as the OLD CA instance is still valid, and issue EE certificates as necessary to maintain a current manifest of all OLD CA published products, but it should not sign any other products. The CURRENT CA should publish its CRL, and should publish all subordinate products, as well as issuing EE certificates as necessary to maintain a current manifest of all CURRENT CA published products. FUTURE CAs should publish no products at all in the repository publication point. It would be consistent with this repository object name framework for the CRL and manifest to be published using object names derived from the hash of the public key value of the CA instance.

Key Rollover:

As a CA instance is associated with a single key pair, there are some considerations regarding the procedure that should be followed by an entity performing a key rollover function. The entity will need to create a new CA instance and then use this new CA instance to re-issue all subordinate products with the new CA instance.

To perform a key rollover operation the entity will need to:

1. Generate a NEW key pair.

2. Generate a certificate request with the NEW key pair and pass the request to the entity's issuer.

3. Request the entity's issuer to generate and publish a NEW CA certificate, with an issuer-selected subject name that is distinct from the subject name used in conjunction with the previous subject name value for this entity.

4. Mark the CURRENT CA as OLD and the NEW CA as CURRENT.

5. The CURRENT CA will generate new certificates for all existing subordinate CA and EE certificates, and publish those products in the same repository publication point and with the same repository publication point name as the previous OLD subordinate CA and EE certificates. The keys in these reissued certificates MUST not change.

6. Where the signing structure uses a packaging format that includes the EE certificate within the signed data, signed objects that included OLD EE certificates in their signed data will need to be re-signed using an EE certificate issued by the CURRENT CA. In the case where the OLD EE certificate is a "single use" EE certificate and the associate private key has been destroyed this will entail the generate of a new key pair, the issuing of an EE certificate by the CURRENT CA. In the case of a "multi-use" EE certificate, the EE certificate should issued using the CURRENT CA. The object, together with the issued EE certificate, should be signed with the associated private key, and published in the same repository publication point, using the same repository publication point name, as the previously signed object that it replaces (i.e. overwrite the old signed object).

7. Generate a certificate revocation request for the OLD CA certificate and pass it to the entity's issuer.

8. Remove all published OLD CA products and destroy the OLD private key.

Name Uniqueness:

This profile specifies that subject names must be unique per issuer, and does not specify that subject names must be globally unique.

Given that the Resource Certificate PKI is a distributed PKI, there is no inherent ability for Certification authorities to coordinate PKI-wide unique subject names. IANA and the RIRs SHOULD use multi-attribute, structured Subject names in their RPKI certificates. All other entities (NIRs, LIRs, etc.) MUST be issued certificates in which the Subject name contains a single relative distinguished name, consisting of a CommonName attribute. This restriction is motivated by the need to change the names of these CAs when key rollover occurs, and to minimize liability for issuers in the RPKI. Also, as the publication repository is distributed, and distinct entities use distinct repository publication points any potential ambiguity is resolved by the distinct publication point.

8. Security Considerations

The Security Considerations of [RFC5280] and [RFC3779]apply to Resource Certificates as defined by this profile, and their use.

A Resource Certificate PKI cannot in and of itself resolve any forms of ambiguity relating to uniqueness of assertions of rights of use in the event that two or more valid certificates encompass the same resource. If the issuance of resource certificates is aligned to the status of resource allocations and assignments then the information conveyed in a certificate is no better than the information in the allocation and assignment databases.

9. IANA Considerations

[Note to IANA, to be removed prior to publication: there are no IANA considerations stated in this document.]

10. Acknowledgements

The authors would like to acknowledge the valued contributions from Stephen Kent, Robert Kisteleki, Randy Bush, Russ Housley, Ricardo

Patara and Rob Austein in the preparation and subsequent review of this document. The document also reflects review comments received from Sean Turner and David Cooper.

11. References

11.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2050] Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D., and J. Postel, "INTERNET REGISTRY IP ALLOCATION GUIDELINES", BCP 12, RFC 2050, November 1996.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, June 2005.

[RFC4211] Schaad, J., "Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)", RFC 4211, September 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,

Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[X.509] ITU-T, "Recommendation X.509: The Directory - Authentication Framework", 2000.

11.2. Informative References

[ID.sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", Work in progress: Internet

Drafts draft-ietf-sidr-arch-03.txt, February 2008.

[ID.sidr-manifests]
Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure", Work in progress: Internet
Drafts draft-ietf-sidr-rpki-manifests-00.txt, January 2008.

[RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985, November 2000.

[RFC2986] Nystrom, M. and B. Kaliski, "PKCS #10: Certification Request Syntax Specification Version 1.7", RFC 2986, November 2000.

[RFC4158] Cooper, M., Dzambasow, Y., Hesse, P., Joseph, S., and R. Nicholas, "Internet X.509 Public Key Infrastructure: Certification Path Building", RFC 4158, September 2005.

[rsync] Tridgell, A., "rsync", April 2006, <http://samba.anu.edu.au/rsync/>.

Appendix A. Example Resource Certificate

The following is an example Resource Certificate.

```
Certificate Name: hu9fdDBq60mrk7cPRuX2DYuXSRQ-3.cer

Data:
  Version: 3
  Serial: 3
  Signature Algorithm: Hash: SHA256, Encryption: RSA
  Issuer: CN=Demo Production APNIC CA - Not for real use,
    E=ca@apnic.net
  Validity:
    Not Before: Thu Jul 27 06:34:04 2006 GMT
    Not After: Fri Jul 27 06:34:04 2007 GMT
  Subject: CN=APNIC own-use network resources
  Subject Key Identifier:
    86:ef:5f:74:30:6a:eb:49:ab:93:b7:0f:46:e5:f6:0d:
    8b:97:49:14
  Subject Key Identifier g(SKI):
    hu9fdDBq60mrk7cPRuX2DYuXSRQ
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
```



```
      RSA Public Key: Modulus:
        c1:25:a1:b0:db:89:83:a0:fc:f1:c0:e4:7b:93:76:c1:
        59:b7:0d:ac:25:25:ed:88:ce:00:03:ea:99:1a:9a:2a:
        0e:10:2e:5f:c0:45:87:47:81:7b:1d:4d:44:aa:65:a3:
        f8:07:84:32:ea:04:70:27:05:2b:79:26:e6:e6:3a:cb:
        b2:9a:65:6c:c1:4e:d7:35:fb:f6:41:1e:8b:1c:b8:e4:
        5a:3a:d6:d0:7b:82:9a:23:03:f8:05:4c:68:42:67:fe:
        e7:45:d9:2c:a6:d1:b3:da:cf:ad:77:c5:80:d2:e3:1e:
        4d:e8:bf:a2:f2:44:10:b2:2f:61:bc:f4:89:31:54:7c:
        56:47:d5:b1:c3:48:26:95:93:c9:6f:70:14:4d:ac:a5:
        c2:8e:3d:1f:6d:f8:d4:93:9d:14:c7:15:c7:34:8e:ba:
        dd:70:b3:c2:2b:08:78:59:97:dd:e4:34:c7:d8:de:5c:
        f7:94:6f:95:59:ba:29:65:f5:98:15:8f:8e:57:59:5d:
        92:1f:64:2f:b5:3d:69:2e:69:83:c2:10:c6:aa:8e:03:
        d5:69:11:bd:0d:b5:d8:27:6c:74:2f:60:47:dd:2e:87:
        24:c2:36:68:2b:3c:fd:bd:22:57:a9:4d:e8:86:3c:27:
        03:ce:f0:03:2e:59:ce:05:a7:41:3f:2f:64:50:dd:e7
      RSA Public Key: Exponent: 65537
  Basic Constraints: CA: TRUE
  Subject Info Access:
    caRepository - rsync://repository.apnic.net/APNIC/
                           pvpjvwUeQix2e54X8fGbhmdYMo0/
```

```
                          q66IrWSGuBE7jqx8PAUHAlHCqRw/
                          hu9fdDBq60mrk7cPRuX2DYuXSRQ/
      Key Usage: keyCertSign, cRLSign
      CRL Distribution Points:
        rsync://repository.apnic.net/APNIC/
               pvpjvwUeQix2e54X8fGbhmdYMo0/
               q66IrWSGuBE7jqx8PAUHAlHCqRw/
               q66IrWSGuBE7jqx8PAUHAlHCqRw.crl
      Authority Info Access: caIssuers -
        rsync://repository.apnic.net/APNIC/
               pvpjvwUeQix2e54X8fGbhmdYMo0/
               q66IrWSGuBE7jqx8PAUHAlHCqRw.cer
      Authority Key Identifier: Key Identifier:
        ab:ae:88:ad:64:86:b8:11:3b:8e:ac:7c:3c:05:07:02:
        51:c2:a9:1c
      Authority Key Identifier: Key Identifier g(AKI):
        q66IrWSGuBE7jqx8PAUHAlHCqRw
      Certificate Policies: 1.3.6.1.5.5.7.14.2
      IPv4: 192.0.2.0/24,
      IPv6: 2001:DB8::/32
      ASNum: 4608, 4777, 9545, 18366-18370
      Signature:
        c5:e7:b2:f3:62:cb:e3:bc:50:1e:6b:90:13:19:f4:5b:
        4a:1c:1c:ab:b5:de:b1:a4:22:e0:28:f5:3b:d0:8c:59:
        0f:85:f2:06:a6:ae:22:e6:d0:99:fe:cb:eb:1d:6a:e2:
        a3:f1:a2:25:95:ec:a7:7d:96:35:dc:16:a7:2f:f5:b7:
```



```
        11:ba:97:05:57:5f:5d:07:5a:c8:19:c8:27:d3:f7:a3:
        92:66:cb:98:2d:e1:7f:a8:25:96:ab:af:ed:87:02:28:
        f5:ae:b6:e3:0c:f7:18:82:70:82:f4:76:54:06:b9:9f:
        e1:a5:f7:ae:72:dd:ee:f0:d4:d2:78:bb:61:73:cf:51:
        26:9f:ea:e8:20:49:06:ba:0c:ac:1d:f6:07:b8:63:a0:
        4d:3d:8e:12:84:3a:d0:ec:94:7e:02:db:d4:85:cf:12:
        5c:7b:12:1a:52:ab:3c:ba:00:f2:71:e7:f0:fd:b3:f4:
        81:e8:a7:cb:07:ca:3a:a4:24:fe:dc:bb:51:16:6a:28:
        33:40:a4:64:60:75:0e:c8:06:c8:5f:e5:98:be:16:a3:
        bc:19:e7:b3:4f:00:0a:8e:81:33:dd:4c:a0:fb:f5:1c:
        1f:1d:3f:b5:90:8b:ec:98:67:76:95:56:8a:94:45:54:
        52:3d:1c:69:4c:6f:8a:9f:09:ec:ef:b0:a9:bc:cf:9d
```

Appendix B. Example Certificate Revocation List

The following is an example Certificate Revocation List.


```
CRL Name: q66IrWSGuBE7jqx8PAUHAlHCqRw.crl

Data:
  Version: 2
  Signature Algorithm:
    Hash: SHA256, Encryption: RSA
  Issuer: CN=Demo Production APNIC CA - Not for real use,
    E=ca@apnic.net
```

```
This Update: Thu Jul 27 06:30:34 2006 GMT
Next Update: Fri Jul 28 06:30:34 2006 GMT
Authority Key Identifier: Key Identifier:
  ab:ae:88:ad:64:86:b8:11:3b:8e:ac:7c:3c:05:
  07:02:51:c2:a9:1c
Authority Key Identifier: Key Identifier g(AKI):
  q66IrWSGuBE7jqx8PAUHAlHCqRw
CRLNumber: 4
Revoked Certificates: 1
  Serial Number: 1
  Revocation Date: Mon Jul 17 05:10:19 2006 GMT
  Serial Number: 2
  Revocation Date: Mon Jul 17 05:12:25 2006 GMT
  Serial Number: 4
  Revocation Date: Mon Jul 17 05:40:39 2006 GMT
Signature:
  b2:5a:e8:7c:bd:a8:00:0f:03:1a:17:fd:40:2c:46:
  0e:d5:64:87:e7:e7:bc:10:7d:b6:3e:39:21:a9:12:
  f4:5a:d8:b8:d4:bd:57:1a:7d:2f:7c:0d:c6:4f:27:
  17:c8:0e:ae:8c:89:ff:00:f7:81:97:c3:a1:6a:0a:
  f7:d2:46:06:9a:d1:d5:4d:78:e1:b7:b0:58:4d:09:
  d6:7c:1e:a0:40:af:86:5d:8c:c9:48:f6:e6:20:2e:
  b9:b6:81:03:0b:51:ac:23:db:9f:c1:8e:d6:94:54:
  66:a5:68:52:ee:dd:0f:10:5d:21:b8:b8:19:ff:29:
  6f:51:2e:c8:74:5c:2a:d2:c5:fa:99:eb:c5:c2:a2:
  d0:96:fc:54:b3:ba:80:4b:92:7f:85:54:76:c9:12:
  cb:32:ea:1d:12:7b:f8:f9:a2:5c:a1:b1:06:8e:d8:
  c5:42:61:00:8c:f6:33:11:29:df:6e:b2:cc:c3:7c:
  d3:f3:0c:8d:5c:49:a5:fb:49:fd:e7:c4:73:68:0a:
  09:0e:6d:68:a9:06:52:3a:36:4f:19:47:83:59:da:
  02:5b:2a:d0:8a:7a:33:0a:d5:ce:be:b5:a2:7d:8d:
  59:a1:9d:ee:60:ce:77:3d:e1:86:9a:84:93:90:9f:
  34:a7:02:40:59:3a:a5:d1:18:fb:6f:fc:af:d4:02:
  d9
```


Appendix C. Cryptographic Message Syntax Profile for RPKI Trust Anchor

Material

Using the Cryptographic Message Syntax (CMS) [RFC3852], a RPKI Trust Anchor Object (RTA) is a type of signed-data object. The general format of a CMS object is:

```
ContentInfo ::= SEQUENCE {
  contentType ContentType,
  content [0] EXPLICIT ANY DEFINED BY contentType }

ContentType ::= OBJECT IDENTIFIER
```

As a RTA is a signed-data object, it uses the corresponding OID, 1.2.840.113549.1.7.2. [RFC3852].

C.1. Signed-Data ContentType

According to the CMS specification, the signed-data content type shall have ASN.1 type SignedData:

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

DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

SignerInfos ::= SET OF SignerInfo
```

The elements of the signed-data content type are as follows:

version
: The version is the syntax version number. It MUST be 3, corresponding to the signerInfo structure having version number 3.

digestAlgorithms
: The digestAlgorithms set MUST include only SHA-256, the OID for which is 2.16.840.1.101.3.4.2.1. [RFC4055]. It MUST NOT contain any other algorithms.

encapContentInfo
This element is defined in Appendix C.1.1.

certificates
The certificates element MUST be included and MUST contain only the single PKI EE certificate needed to validate this CMS Object. The CertificateSet type is defined in section 10 of [RFC3852]

crls
The crls element MUST be omitted.

signerInfos
This element is defined in Appendix C.1.2.

C.1.1. encapContentInfo

encapContentInfo is the signed content, consisting of a content type identifier and the content itself.

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

ContentType ::= OBJECT IDENTIFIER
```

The elements of this signed content type are as follows:

eContentType
The ContentType for an RTA is defined as id-ct-RPKITrustAnchor and has the numerical value of 1.2.840.113549.1.9.16.1.33.

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

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

id-ct-RPKITrustAnchor OBJECT IDENTIFIER ::= { id-ct 33 }
```

eContent
The content of an RTA is an RPKI self-signed CA certificate. It is formally defined as:

```
id-ct-RPKITrustAnchor ::= Certificate
```

The definition of Certificate is taken from [X.509].

C.1.2. signerInfos

SignerInfo is defined under CMS as:

```
SignerInfo ::= SEQUENCE {
  version CMSVersion,
  sid SignerIdentifier,
  digestAlgorithm DigestAlgorithmIdentifier,
  signedAttrs [0] IMPLICIT SignedAttributes OPTIONAL,
  signatureAlgorithm SignatureAlgorithmIdentifier,
  signature SignatureValue,
  unsignedAttrs [1] IMPLICIT UnsignedAttributes OPTIONAL }
```

The content of the SignerInfo element are as follows:

version
: The version number MUST be 3, corresponding with the choice of SubjectKeyIdentifier for the sid.

sid
: The sid is defined as:

```
SignerIdentifier ::= CHOICE {
    issuerAndSerialNumber IssuerAndSerialNumber,
    subjectKeyIdentifier [0] SubjectKeyIdentifier }
```

: For a RTA, the sid MUST be a SubjectKeyIdentifier.

digestAlgorithm
: The digestAlgorithm MUST be SHA-256, the OID for which is 2.16.840.1.101.3.4.2.1. [RFC4055]

signedAttrs
: The signedAttrs element is defined as:

```
SignedAttributes ::= SET SIZE (1..MAX) OF Attribute

Attribute ::= SEQUENCE {
  attrType OBJECT IDENTIFIER,
```

```
    attrValues SET OF AttributeValue }

  AttributeValue ::= ANY
```

The signedAttr element MUST be present and MUST include the content-type and message-digest attributes. The signer MAY also include the signing-time signed attribute, the binary-signing-time signed attribute, or both signed attributes. Other signed attributes that are deemed appropriate MAY also

be included. The intent is to allow additional signed attributes to be included if a future need is identified. This does not cause an interoperability concern because unrecognized signed attributes are ignored by the relying party.

The signedAttr MUST include only a single instance of any particular attribute. Additionally, even though the syntax allows for a SET OF AttributeValue, in a RTA the attrValues must consist of only a single AttributeValue.

ContentType Attribute
: The ContentType attribute MUST be present. The attrType OID for the ContentType attribute is 1.2.840.113549.1.9.3.

  The attrValues for the ContentType attribute in a RTA MUST be 1.2.840.113549.1.9.16.1.24 (matching the eContentType in the EncapsulatedContentInfo).

MessageDigest Attribute
: The MessageDigest attribute MUST be present. The attrType OID for the MessageDigest Attribute is 1.2.840.113549.1.9.4.

  The attrValues for the MessageDigest attribute contains the output of the digest algorithm applied to the content being signed, as specified in Section 11.1 of [RFC3852].

SigningTime Attribute

The SigningTime attribute MAY be present. If it is present it MUST be ignored by the relying party. The presence of absence of the SigningTime attribute in no way affects the validation of the RTA. The attrType OID for the SigningTime attribute is 1.2.840.113549.1.9.5.

The attrValues for the SigningTime attribute is defined as:

```
SigningTime ::= Time

Time ::= CHOICE {
   utcTime UTCTime,
   generalizedTime GeneralizedTime }
```


The Time element specifies the time, based on the local system clock, at which the digital signature was applied to the content.

BinarySigningTime Attribute

The BinarySigningTime attribute MAY be present. If it is present it MUST be ignored by the relying party. The presence of absence of the BinarySigningTime attribute in no way affects the validation of the RTA. The attrType OID for the SigningTime attribute is 1.2.840.113549.1.9.16.2.46.

The attrValues for the SigningTime attribute is defined as:

```
BinarySigningTime ::= BinaryTime

BinaryTime ::= INTEGER (0..MAX)
```

The BinaryTime element specifies the time, based on the local system clock, at which the digital signature was applied to the content.

signatureAlgorithm

The signatureAlgorithm MUST be RSA (rsaEncryption), the OID for which is 1.2.840.113549.1.1.1.q

signature
The signature value is defined as:

```
SignatureValue ::= OCTET STRING
```

The signature characteristics are defined by the digest and signature algorithms.

unsignedAttrs
unsignedAttrs MUST be omitted.

C.2. RTA Validation

Before a relying party can use an RTA, the relying party must first validate the RTA by performing the following steps.

1. Verify that the RTA syntax complies with this specification. In particular, verify the following:

a. The contentType of the CMS object is SignedData (OID 1.2.840.113549.1.7.2).

b. The version of the SignedData object is 3.

c. The digestAlgorithm in the SignedData object is SHA-256 (OID 2.16.840.1.101.3.4.2.1).

d. The certificates field in the SignedData object is present and contains a single EE certificate whose Subject Key Identifier (SKI) matches the sid field of the SignerInfo object.

e. The crls field in the SignedData object is omitted.

f. The eContentType in the EncapsulatedContentInfo is id-ct-RPKITrustAnchor (OID 1.2.840.113549.1.9.16.1.[TBD])

g. The version of the SignerInfo is 3.

h. The digestAlgorithm in the SignerInfo object is SHA-256 (OID 2.16.840.1.101.3.4.2.1).

i. The signatureAlgorithm in the SignerInfo object is RSA (OID 1.2.840.113549.1.1.1).

j. The signedAttrs field in the SignerInfo object is present and contains both the ContentType attribute (OID 1.2.840.113549.1.9.3) and the MessageDigest attribute (OID 1.2.840.113549.1.9.4).

k. The unsignedAttrs field in the SignerInfo object is omitted.

2. Use the public key in the EE certificate to verify the signature on the RTA.

3. Verify that the EE certificate is a valid end-entity certificate in the Trust Anchor PKI by validating that the PKI CA certificate issued this EE certificate, and the PKI CA's CRL has not revoked the EE certificate, and that the PKI CA's CRL is valid.

Authors' Addresses

Geoff Huston
Asia Pacific Network Information Centre

Email: gih@apnic.net
URI: http://www.apnic.net

George Michaelson
Asia Pacific Network Information Centre

Email: ggm@apnic.net
URI: http://www.apnic.net

Robert Loomans
Asia Pacific Network Information Centre

Email: robertl@apnic.net
URI: http://www.apnic.net

Full Copyright Statement

Intellectual Property