Secure Inter-Domain Routing (sidr)
Internet Draft
Expires: November 9, 2011
Intended Status: Proposed Standard

M. Lepinski
S. Kent
D. Kong
BBN Technologies

May 9, 2011

A Profile for Route Origin Authorizations (ROAs)
draft-ietf-sidr-roa-format-12.txt

Status of this Memo

Copyright Notice

Abstract

This document defines a standard profile for Route Origin Authorizations (ROAs). A ROA is a digitally signed object that provides a means of verifying that an IP address block holder has authorized an Autonomous System (AS) to originate routes to one or more prefixes within the address block.

Table of Contents

1. Introduction

The primary purpose of the Internet IP Address and Autonomous System (AS) Number Resource Public Key Infrastructure (RPKI) system is to improve routing security. (See [ARCH] for more information.) As part of this system, a mechanism is needed to allow entities to verify that an AS has been given permission by an IP address block holder to advertise routes to one or more prefixes within that block. A ROA provides this function.

The ROA makes use of the template for RPKI digitally signed objects [SIGNOBJ], which defines a Crytopgraphic Message Syntax (CMS) [RFC5652] wrapper for the ROA content as well as a generic validation procedure for RPKI signed objects. Therefore, to complete the specification of the ROA (see Section 4 of [SIGNOBJ]), this document defines:

1. The OID that identifies the signed object as being a ROA. (This OID appears within the eContentType in the encapContentInfo object as well as the ContentType signed attribute in the signerInfo object.)

2. The ASN.1 syntax for the ROA eContent. (This is the payload that specifies the AS being authorized to originate routes as well as the prefixes to which the AS may originate routes.)

3. An additional step required to validate ROAs (in addition to the validation steps specified in [SIGNOBJ]).

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] and "X.509 Extensions for IP Addresses and AS Identifiers" [RFC3779].

Additionally, this document makes use of the RPKI signed object profile [SIGNOBJ] and thus familiarity with that document is assumed. Note that the RPKI signed object profile makes use of certificates adhering to the RPKI resource certificate profile [RESCERT] and thus familiarly with this profile is also assumed.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

2. The ROA ContentType

The ContentType for a ROA is defined as routeOriginAuthz and has the numerical value of 1.2.840.113549.1.9.16.1.24.

This OID MUST appear both within the eContentType in the encapContentInfo object as well as the ContentType signed attribute in the signerInfo object (see [SIGNOBJ]).

3. The ROA eContent

The content of a ROA identifies a single AS that has been authorized by the address space holder to originate routes and a list of one or more IP address prefixes that will be advertised. If the address space holder needs to authorize multiple ASes to advertise the same

set of address prefixes, the holder issues multiple ROAs, one per AS number. A ROA is formally defined as:

```
RouteOriginAttestation ::= SEQUENCE {
   version [0] INTEGER DEFAULT 0,
   asID  ASID,
   ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF ROAIPAddressFamily }

ASID ::= INTEGER

ROAIPAddressFamily ::= SEQUENCE {
   addressFamily OCTET STRING (SIZE (2..3)),
   addresses SEQUENCE (SIZE (1..MAX)) OF ROAIPAddress }


ROAIPAddress ::= SEQUENCE {
   address IPAddress,
   maxLength INTEGER OPTIONAL }

IPAddress ::= BIT STRING
```

Note that this content appears as the eContent within the encapContentInfo (see [SIGNOBJ]).

3.1. version

The version number of the RouteOriginAttestation MUST be 0.

3.2. asID

The asID field contains the AS number that is authorized to originate routes to the given IP address prefixes.

3.3. ipAddrBlocks

The ipAddrBlocks field encodes the set of IP address prefixes to which the AS is authorized to originate routes. Note that the syntax here is more restrictive than that used in the IP Address Delegation extension defined in RFC 3779. That extension can represent arbitrary address ranges, whereas ROAs need to represent only prefixes.

Within the ROAIPAddressFamily structure, addressFamily contains the Address Family Identifier (AFI) of an IP address family. This specification only supports IPv4 and IPv6. Therefore, addressFamily MUST be either 0001 or 0002.

Within a ROAIPAddress structure, the addresses field represents prefixes as a sequence of type IPAddress. (See [RFC3779] for more details). If present, the maxLength MUST be an integer greater than or equal to the length of the accompanying prefix and less than or equal to the length (in bits) of an IP address in the address family (32 for IPv4 and 128 for IPv6). When present, the maxLength specifies the maximum length of IP address prefix that the AS is authorized to advertise. (For example, if the IP Address prefix is 203.0.113/24 and the maxLength is 26, the AS is authorized to advertise any more specific prefix having length at most 26. That is, in this example, the AS would be authorized to advertise 203.0.113/24, 203.0.113.128/25, or 203.0.113.0/25, but not 203.0.113.0/27.) When the maxLength is not present, the AS is only authorized to advertise exactly the prefix specified in the ROA.

Note that a valid ROA may contain an IP Address prefix (within a ROAIPAddress element) that is encompassed by another IP Address prefix (within a separate ROAIPAddress element). For example, a ROA may contain the prefix 203.0.113/24 with maxLength 26, as well as the prefix 203.0.113.0/28 with maxLength 28. (Such a ROA would authorize the indicated AS to advertise any prefix beginning with 203.0.113 with length at least 24 and no greater than 26, as well as the specific prefix 203.0.113.0/28.) Additionally, a ROA MAY contain two ROAIPAddress elements where the IP Address prefix is identical in both cases. However, this is NOT RECOMMENDED as in such a case the ROAIPAddress with the shorter maxLength grants no additional privileges to the indicated AS and thus can be omitted without changing the meaning of the ROA.

4. ROA Validation

Before a relying party can use a ROA to validate a routing announcement, the relying party MUST first validate the ROA. To validate a ROA the relying party MUST perform all the validation checks specified in [SIGNOBJ] as well as the following additional ROA-specific validation step.

- The IP Address Delegation extension [RFC3779] is present in the End-Entity (EE) certificate (contained within the ROA) and each IP address prefix(es) in ROA is contained within the set of IP addresses specified by the EE certificate's IP address delegation extension.

5. Security Considerations

There is no assumption of confidentiality for the data in a ROA; it is anticipated that ROAs will be stored in repositories that are

accessible to all ISPs, and perhaps to all Internet users. There is no explicit authentication associated with a ROA, since the PKI used for ROA validation provides authorization but not authentication. Although the ROA is a signed, application layer object, there is no intent to convey non-repudiation via a ROA.

The purpose of a ROA is to convey authorization for an AS to originate a route to the prefix(es) in the ROA. Thus the integrity of a ROA MUST be established. The ROA specification makes use of the RPKI signed object format, thus all security considerations in [SIGNOBJ] also apply to ROAs. Additionally, the signed object profile uses the CMS signed message format for integrity, and thus ROA inherit all security considerations associated with that data structure.

The right of the ROA signer to authorize the target AS to originate routes to the prefix(es) is established through use of the address space and AS number PKI described in [ARCH]. Specifically one MUST verify the signature on the ROA using an X.509 certificate issued under this PKI, and check that the prefix(es) in the ROA match those in the address space extension in the certificate.

6. IANA Considerations

None.

7. Acknowledgments

The authors wish to thank Charles Gardiner and Russ Housley for their help and contributions. Additionally, the authors would like to thank Rob Austein, Roque Gagliano, Danny McPherson and Sam Weiler for their careful reviews and helpful comments.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5652] Housley, R., "Cryptographic Message Syntax", RFC 5652, September 2009.

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

[RFC5280] Cooper, D., et. al., "Internet X.509 Public Key Infrastructure and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RESCERT] Huston, G., Michaelson, G., and Loomans, R., "A Profile for X.509 PKIX Resource Certificates", draft-ietf-sidr-res-certs, November 2010.

[SIGNOBJ] Lepinski, M., Chi, A., and Kent, S., "Generic Signed Objects for the Resource Public Key Infrastructure", draft-ietf-sidr-signed-object, February 2011.

8.2. Informative References

[ARCH] Lepinski, M. and Kent, S., "An Infrastructure to Support Secure Internet Routing," draft-ietf-sidr-arch, February 2011.

APPENDIX A: ASN.1 Module

This normative appendix provides an ASN.1 module that specifies the ROA content in ASN.1 syntax.

```
RPKI-ROA { iso(1) member-body(2) us(840) rsadsi(113549)
   pkcs(1) pkcs9(9) smime(16) mod(0) 61 }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

RouteOriginAttestation ::= SEQUENCE {
   version [0] INTEGER DEFAULT 0,
   asID  ASID,
   ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF ROAIPAddressFamily }

ASID ::= INTEGER

ROAIPAddressFamily ::= SEQUENCE {
   addressFamily OCTET STRING (SIZE (2..3)),
   addresses SEQUENCE (SIZE (1..MAX)) OF ROAIPAddress }

ROAIPAddress ::= SEQUENCE {
   address IPAddress,
   maxLength INTEGER OPTIONAL }

IPAddress ::= BIT STRING

END
```

Authors' Addresses

Matt Lepinski
BBN Technologies
10 Moulton Street
Cambridge MA 02138

Email: mlepinski@bbn.com

Stephen Kent
BBN Technologies
10 Moulton Street
Cambridge MA 02138

Email: skent@bbn.com

Derrick Kong
BBN Technologies
10 Moulton Street
Cambridge MA 02138

Email: dkong@bbn.com