Individual Submission
Internet-Draft
Intended status: BCP
Expires: November 9, 2010

G. Huston
G. Michaelson
APNIC

May 8, 2010

# A Profile for AS Adjacency Attestation Objects
## draft-huston-sidr-aao-profile-03.txt

## Abstract

This document describes a profile for AS Adjacency Attestation Objects (AAOs). An AAO is a digitally signed object that provides a means of verifying that an AS holder has made an attestation that it has a inter-domain routing adjacency with one or more other AS's, with the associated inference that this AS is prepared to announce or receive routes with these adjacent AS's in the inter-domain domain environment.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 9, 2010.

## Copyright Notice

Table of Contents

## 1. Introduction

The primary purpose of the Internet IP Address and AS Number Resource Public Key Infrastructure (RPKI) system [ID.ietf-sidr-arch] is to improve routing security. As part of this security framework, a mechanism is defined here to allow entities to verify that an AS holder attests that is adjacent to one or more other AS's, with the inference that it is prepared to announce routes to these adjacent AS's in the inter-domain routing environment. An AS Adjacency Attestation Object (AAO) provides this function.

An AAO is a digitally signed object that makes use of Cryptographic Message Syntax (CMS) [RFC5652] as a standard encapsulation format. CMS was chosen to take advantage of existing open source software available for processing messages in this format.

The AAO is an attestation, made and issued by the local AS holder, that the local AS is an inter-domain routing peer with each of the AS's that are enumerated in an associated AS list contained in the AAO. An AAO is a two part structure, containing the local AS and a list of adjacent AS's. The AAO is signed by a an End Entity (EE) Resource Certificate that has the local AS as the value of its [RFC3779] AS number resource extension.

### 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], and BGP-4 [RFC4271]

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. Semantic Interpretation of an AAO

An AAO is an attestation on the part of a AS holder that it supports currently active inter-domain routing adjacencies to each of the AS's listed in the AAO. The AAO does not list any prefixes that may be announced to the adjacent AS's either directly or indirectly. The AAO also does not list any local routing policies that have been applied to the routes that are advertised across this adjacency, nor any routing policies that may be applied to routes that are learned from this adjacency.





The AAO is intended to provide "closure" with respect to interpretation of the AAO by relying parties, to the extent that if a valid AAO exists for a local AS, then from the perspective of that local AS all adjacencies with those AS's listed in the valid AAO can be regarded as "valid" and any other adjacency from the perspective of the local AS can be regarded as potentially "invalid". In other words an AAO is an attestation of adjacency with the AS's listed in the AAO and an implicit attestation of the denial of adjacency with all other AS's.

Where an AS holder has published two or more valid AAO's, the set of "valid" adjacent AS's refers to the union of the lists of adjacent AS's and all other AS's can be regarded as "invalid" from the perspective of the local AS.

A relying party may infer from a valid AAO that the signing AS holder may have the intent to advertise route objects across this inter-AS routing adjacency, and may be prepared to learn route objects that are passed to it from the adjacent AS. The AAO does not described which routes may be announced across a corresponding inter-AS routing adjacency.

It is noted that an AAO is an asymmetric assertion, where one AS is asserting that an inter-domain routing adjacency with another AS exists. It should also be noted that this assertion is not explicitly acknowledged by the remote AS in the context of a single issued AAO. Relying parties may elect to place greater levels of confidence in the existence of an inter-domain routing adjacency when both AS's have signed and published AAO objects that contain mutual

references.

It is also noted that there is a subtle distinction that could be drawn here between the appropriate semantic interpretation a pair of unilateral assertions of adjacency using two AAOs and a combined assertion of adjacency where both AS's sign a single attestation of the existence of an inter-domain routing adjacency between these AS's. Such a combined approach, using a single assertion with two digital signatures, is not defined in this document.

## 3. Basic Format

Using CMS syntax, an AAO 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 AAO is a signed-data object, it uses the corresponding OID, 1.2.840.113549.1.7.2. [RFC5652]

### 3.1. Signed-Data Content Type

According to the CMS standard, 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
```

### 3.1.1. version

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

### 3.1.2. digestAlgorithms

The digestAlgorithms set contains the OIDs of the digest algorithm(s) used in signing the encapsulated content. This set MUST conform to the RPKI Algorithms and Key Size Profile specification [ID.sidr-rpki-algs].

### 3.1.3. 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
```





#### 3.1.3.1. eContentType

The ContentType for a AAO is defined as id-ct-ASAdjancyAttest and has the numerical value of 1.2.840.113549.1.9.16.1.32.

```
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-ASAdjacencyAttest OBJECT IDENTIFIER ::= { id-ct 32 }
```

#### 3.1.3.2. eContent

The content of an AAO identifies one or more AS's that the signing AS

holder is attesting the existence of a routing adjacency.

The AAO contains no routing policy qualifications, nor does it reference any address prefixes that may be announced or received within the context of any routing adjacency.

An AAO is defined as:

```
id-ct-ASAdjacencyAttest ::= SEQUENCE {
  version [0] INTEGER DEFAULT 0,
  ASIdentifiers       ::= SEQUENCE OF ASIdOrRange,
  localASNum ASId}

ASIdOrRange          ::= CHOICE {
  id                   ASId,
  range                ASRange }

ASRange              ::= SEQUENCE {
  min                  ASId,
  max                  ASId }

ASId                 ::= INTEGER
```

#### 3.1.3.2.1. version

The version number of the ASAdjacencyAttestation MUST be 0.

#### 3.1.3.2.2. ASIdentifiers

The ASIdentifiers element is a SEQUENCE containing AS numbers for which the localASnum AS is attesting the existence of a routing adjacency. Any pair of items in the asIdentifiers SEQUENCE MUST NOT overlap. Any contiguous series of AS identifiers MUST be combined





into a single range whenever possible. The AS identifiers in the asIdentifiers element MUST be sorted by increasing numeric value.

#### 3.1.3.2.2.1. ASIdOrRange

The ASIdOrRange type is a CHOICE of either a single integer (ASId) or a single sequence (ASRange).

#### 3.1.3.2.2.2. ASRange

The ASRange type is a SEQUENCE consisting of a min and a max element, and is used to specify a range of AS identifier values.

##### 3.1.3.2.2.2.1. min and max

The min and max elements have type ASId. The min element is used to specify the value of the minimum AS identifier in the range, and the max element specifies the value of the maximum AS identifier in the range.

#### 3.1.3.2.2.3. ASId

The ASId type is an INTEGER.

### 3.1.3.2.3. localASNum

The localASNum field contains the AS that is making the attestation of routing adjacency to each of the AS's listed in the ASIdentifiers element.

## 3.1.4. CertificateSet

The CertificateSet type is defined in section 10 of [RFC5652]

## 3.1.5. certificates

The certificates element MUST be included and MUST contain only the single EE resource certificate needed to validate this AAO.

## 3.1.6. crls

The crls element MUST be omitted.

## 3.1.7. 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 }
```

#### 3.1.7.1. version

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

#### 3.1.7.2. sid

The sid is defined as:

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

For a AAO, the sid MUST be a SubjectKeyIdentifier.

#### 3.1.7.3. digestAlgorithm

The digestAlgorithm MUST consist of the OID of a digest algorithm that conforms to the RPKI Algorithms and Key Size Profile specification [ID.sidr-rpki-algs].

#### 3.1.7.4. signedAttrs

The signedAttrs 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 signed 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 by the signer 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 AAO the attrValues must consist of only a single AttributeValue

#### 3.1.7.4.1. 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 AAO MUST be 1.2.840.113549.1.9.16.1.24 (matching the eContentType in the EncapsulatedContentInfo).

#### 3.1.7.4.2. 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 [RFC5652].

#### 3.1.7.4.3. 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 AAO (as specified in Section 4). 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.

#### 3.1.7.4.4. BinarySigningTimeAttribute

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 AAO (as specified in Section 3). 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.

#### 3.1.7.5. signatureAlgorithm

The signatureAlgorithm MUST consist of the OID of a signature algorithm that conforms RPKI Algorithms and Key Size Profile specification [ID.sidr-rpki-algs].

#### 3.1.7.6. signature

The signature value is defined as:

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

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

#### 3.1.7.7. unsignedAttrs

unsignedAttrs MUST be omitted.

## 4. AAO Validation

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

1. Verify that the AAO 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 certificates field in the SignedData object is present and contains an EE certificate whose Subject Key Identifier (SKI) matches the sid field of the SignerInfo object.

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

   e. The eContentType in the EncapsulatedContentInfo is id-ct-ADAdjacencyAttest (OID 1.2.840.113549.1.9.16.1.32)

   f. The version of the id-ct-ASAdjacencyAttest is 0.

   g. The version of the SignerInfo is 3.

   h. 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).

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

   j. The digestAlgorithm in the SignedData and SignerInfo objects as well as the signatureAlgorithm in the SignerInfo object conform to the RPKI Algorithms and Key Size Profile specification [ID.sidr-rpki-algs].

2. The public key in the EE certificate (contained within the AAO) can be used to successfully verify the signature on the AAO.

3. The EE certificate has an Autonomous System Identifier Delegation Extension [RFC3779] and that the Autonomous System Identifier in that extension exactly matches the Autonomous System Identifier in the localASNum element of the AAO.

4. The EE certificate is a valid end-entity certificate in the Resource PKI as specified by [ID.ietf-sidr-res-certs]. (in particular, there exists a valid certification path from a trust anchor to the EE certificate.)

## 5. Security Considerations

There is no assumption of confidentiality for the data in a AAO; it is anticipated that AAOs will be stored in public repositories that





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

The purpose of a AAO is to convey a unilateral statement of routing capability that an AS has the capability to announce route objects via a routing adjacency with another AS and has the capability to listen for route objects that are passed to it over a routing adjacency. This should not be interpreted as an authority, nor is a relying party justified in assuming that such a routing adjacency exists, nor that any valid routing announcements that are passed across this routing adjacency.

A relying party may be able to place greater confidence in the inferred existence of a routing adjacency in the case where both AS holders have issued current AAO objects that nominate each other as an adjacent AS.

The AAO object does not convey any information relating to route policies that may be applied to the adjacency by either party to a route adjacency, nor what prefixes may be advertised across that adjacency, nor any attributes that may be associated with such advertisements.

## 6. IANA Considerations

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

## 7. Acknowledgements

The authors would like to acknowledge the work of Matt Lepinski, Stephen Kent and Derrick Kong, whose work on the Route Origin Attestation Profile was used as the starting point for this document.

## 8. References

### 8.1. Normative References

[ID.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch (work in progress), March 2009.





[ID.ietf-sidr-res-certs]
Huston, G., Michaleson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", Internet Draft draft-ietf-sidr-res-certs, February 2009.

[ID.sidr-rpki-algs]
Huston, G., "A Profile for Algorithms and Key Sizes for use in the Resource Public Key Infrastructure", Work in progress: Internet Drafts draft-ietf-sidr-rpki-algs-00.txt, August 2009.

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

[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.

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

## 8.2. Informative References

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

## Authors' Addresses

Geoff Huston

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

George Michaelson

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