Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 8, 2019

G. Michaelson
G. Huston
T. Harrison
APNIC
T. Bruijnzeels
M. Hoffmann
opennetlabs

March 7, 2019

A profile for Resource Tagged Attestations (RTAs) draft-michaelson-rpki-rta-01

Abstract

This document defines a Cryptographic Message Syntax (CMS) profile for a general purpose Resource Tagged Attestation (RTA), for use with the Resource Public Key Infrastructure (RPKI). The objective is to allow an attestation, in the form of an arbitrary digital object, to be signed "with resources", and for validation to provide an outcome of "valid with resources". The profile is intended to provide for the signing of an attestation with an arbitrary set of resources.

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 https://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 September 8, 2019.

Copyright Notice

Table of Contents

1. Introduction

This document defines a Cryptographic Message Syntax (CMS) [RFC5652] profile for a general purpose Resource Tagged Attestation (RTA), for use with the Resource Public Key Infrastructure (RPKI) [RFC6480]. An RTA allows an arbitrary digital object to be signed "with resources," and for validation of the digital signature to provide an outcome of "valid with resources." The profile is intended to provide for the signing of a arbitrary attestation with a set of resources by the duly delegated resource holder(s).

The RTA makes use of the template for RPKI Digitally Signed Objects [RFC6488], which defines a CMS wrapper for the RTA content, as well as a generic validation procedure for RPKI signed objects. However, this specification does not comply to the profile in [RFC6488] in all

respects. This document describes the areas of difference to the template profile, the ASN.1 syntax for the RTA eContent, and the additional steps required to validate RTAs (in addition to the validation steps specified in [RFC6488].

An RTA is a detached signature CMS model, it leverages concepts documented in [RFC8358] and [RFC5485]. Text from these RFCs has been repurposed removing references to internet-drafts and RFCs since this is a general detached signature signing model.

2. Conventions Used In This Document

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 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

3. RTA Profile

An RTA conforms to the template for RPKI Digitally Signed Objects [RFC6488], with the exception that in order to allow for arbitrary resource sets to be used to sign an RTA, it may be necessary to use multiple signatures to sign an RTA.

The differences between this RTA profile and the profile specified by the RPKI Digitally Signed Object template are as follows:

- Section 2.1 of [RFC6488] specifies a single SignerInfo object. An RTA MAY contain more than one SignerInfo object.

- Section 2.1.4, and Section 3 of [RFC6488] specify that the certificates field contains a single EE certificate. The certificates field of an RTA contains precisely the same number of EE certificates as there are SignerInfo objects in the RTA, where each EE certificate is needed to validate the signature in each SignerInfo. In addition, the certificates field MAY contain a collection of CA certificates that would allow a RP to validate the EE certificates.

- Section 2.1.5 of [RFC6488] specifies that the crls field be omitted. For RTAs the crls field MUST contain the current CRL for each CA certificate that has been included in the certificates field of the RTA.

- Section 3 of [RFC6488] describes the signed object validation checks that are to be performed by a Relying Party. Additional

validation checks for an RTA are required, as described in section 5 of this profile.

4. The RTA ContentType

The ContentType for an RTA is defined as resourceTaggedAttestation, and has the numerical value of 1.2.840.113549.1.9.16.1.36

This OID MUST appear both within the eContentType in the encapContentInfo object as well as the ContentType signed attribute in the signerInfo object (see [RFC6488]).

5. The RTA eContent

The content of an RTA indicates that an arbitrary digital object has been signed "with resources". An RTA is formally defined as:

```
ResourceTaggedAttestationDefinitions DEFINITIONS ::=
BEGIN

  -- definition from rfc3029
 id-ct OBJECT IDENTIFIER ::= { iso(1) member-body(2)
            us(840) rsadsi(113549) pkcs(1) pkcs-9(9) id-smime(16) 1 }

 id-ct-resourceTaggedAttestation OBJECT IDENTIFIER ::=
            { id-ct(1) 36 }

  ResourceTaggedAttestation ::= SEQUENCE {
      version  [0]          INTEGER DEFAULT 0,
      subjectKeyIdentifers  SubjectKeys,
      resources             ResourceBlock,
      digestAlgorithm       AlgorithmIdentifer,
      messageDigest         OCTET STRING }

   SubjectKeys         ::= SET SIZE (1..MAX) OF SubjectKeyIdentifier
          -- defined in RFC5280

   ResourceBlock       ::= SEQUENCE {
     asID         [0]       AsList OPTIONAL,
     ipAddrBlocks [1]       IPList OPTIONAL }
         -- at least one of asID or ipAddrBlocks must be present

   AsList              ::= SEQUENCE (SIZE(1..MAX)) OF ASIdOrRange
   ASIdOrRange         ::= CHOICE {
      id                    ASId,
      range                 ASRange }

   ASRange             ::= SEQUENCE {
```

```
         min                 ASId,
         max                 ASId }

      ASId               ::= INTEGER

      IPList             ::= SEQUENCE (SIZE(1..MAX)) OF IPAddressFamily

      IPAddressFamily    ::= SEQUENCE {    -- AFI & optional SAFI --
         addressFamily       OCTET STRING (SIZE (2..3)),
         addressesOrRanges   SEQUENCE OF IPAddressOrRange }

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

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

      IPAddress          ::= BIT STRING

      -- imported from [@!RFC5280]
      AlgorithmIdentifer ::= SEQUENCE {
         algorithm           OBJECT IDENTIFIER,
         parameters          ANY DEFINED BY algorithm OPTIONAL }
   END
```

Note that this content appears as the eContent within the encapContentInfo (see [RFC6488]).

[TODO: this needs some work. The AttestationSet is from prior pre-00 state. What is this referring to?]

Note that AttestationSet is a SET OF EncapsulatedContentInfo from [RFC5485]

5.1. version

The version number of the ResourceTaggedAttestation MUST be 0.

5.2. subjectKeyIdentifiers

The subjectKeyIdentifiers MUST be the set of SubjectKeyIdentifier values contained in each of the EE certificates carried in the CMS certificates field.

5.3. resources

The resources contained here are the resources used to tag the attestation, and MUST match the set of resources listed by the set of EE certificates carried in the CMS certificates field.

The ordering of resources is defined in [RFC3779].

5.4. digestAlgorithm

The digest algorithm used to create the message digest of the attested digital object. This algorithm MUST be a hashing algorithm defined in [RFC7935].

5.5. messageDigest

The message digest of the attested digital object using the algorithm specified in the digestAlgorithm field.

5.6. attestations

The SET OF EncapsulatedContentInfo [RFC5485] which form the individual digital signatures, made by each signing party. For each instance in the set, one of the subjectKeyIdentifiers MUST identify a certificate which can validate the signature. This means that there will be an instance of a SignedData and SignerInfo for that subjectKeyIdentifier (SignerInfo.sid)

The eContentType is id-ct-anyContentType, which refers to the ASN.1 ANY octet sequence.

6. RTA Validation

To validate an RTA the relying party MUST perform all the validation checks specified in [RFC6488] as well as the following additional RTA-specific validation steps.

- Canonicalization of the attested object MUST be performed.
- The message digest of the attested object using the digest algorithm specified in the the digestAlgorithm field MUST be calculated and MUST match the value given in the messageDigest field of the RTA content.
- The signature verification process defined section 5.6 of [RFC5652] MUST be performed for all public keys referenced in each SignerInfo of the CMS. If any signature cannot be verified, then the RTA MUST NOT be validated. This process includes CRL checks

which may require fetching from the CRLDP of any certificate without an embedded CRL in the CMS which is current.

[TODO more text needed about CRL/CRLDP and handling expired CRLS]

- The set of public keys contained in the subjectKeyIdentifers of the RTA MUST exactly match the set of subjectKeyIdentifiers contained in the set of SignerInfo objects of the CMS object.

- The set of resources contained in resources of the RTA MUST exactly match the set of resources contained in the set of EE certificates of the CMS object.

- The number of certificates in the CMS object MUST equal the number of signerInfo objects in the CMS, and the subjectKeyidentifiers in these certificates MUST match one and only one subjectkeyidentifier of a signerinfo object.

7. Need for Canonicalization

As in [RFC5485] and [RFC8358] there is a need for canonicalization.

The following text is based on section 4 of [RFC8358] with changes to remove references to internet-drafts and RFCs.

In general, the content is treated like a single octet string for the generation of the digital signature. Unfortunately, text and HTML files require canonicalization to avoid signature validation problems. The primary concern is the manner in which different operating systems indicate the end of a line of text. Some systems use a single new-line character, other systems use the combination of the carriage-return character followed by a line-feed character, and other systems use fixed-length records padded with space characters. For the digital signature to validate properly, a single convention must be employed.

7.1. ASCII, UTF-8, and HTML File Canonicalization

The canonicalization procedure follows the conventions used for text files in the File Transfer Protocol (FTP) [FTP]. Such files must be supported by FTP implementations, so code reuse seems likely.

The canonicalization procedure converts the data from its internal character representation to the standard 8-bit NVT-ASCII representation (see TELNET [TELNET]). In accordance with the NVT standard, the <CRLF> sequence MUST be used to denote the end of a line of text. Using the standard NVT-ASCII representation means that data MUST be interpreted as 8-bit bytes.

Trailing space characters MUST NOT appear on a line of text. That is, the space character must not be followed by the <CRLF> sequence.

Thus, a blank line is represented solely by the <CRLF> sequence.

The form-feed nonprintable character (0x0C) is expected.

Other non-printable characters, such as tab and backspace, are not expected, but they do occur. Non-printable or non-ASCII characters (ones outside the range 0x20 to 0x7E) MUST NOT be changed in any way not covered by the rules for end-of-line handling in the previous paragraph.

Trailing blank lines MUST NOT appear at the end of the file. That is, the file must not end with multiple consecutive <CRLF> sequences.

In some environments, a Byte Order Mark (BOM) (U+FEFF) is used at the beginning of a file to indicate that it contains non-ASCII characters. In UTF-8 or HTML files, a BOM at the beginning of the file is not considered to be part of the file content. One or more consecutive leading BOMs, if present, MUST NOT be processed by the digital signature algorithm.

Any end-of-file marker used by an operating system is not considered to be part of the file content. When present, such end-of-file markers MUST NOT be processed by the digital signature algorithm.

Note: This text file canonicalization procedure is consistent with the NVT-ASCII definition offered in Appendix B of RFC 5198 [UFNI].

7.2. XML File Canonicalization

Utilities that produce XML files are expected to follow the guidance provided by the World Wide Web Consortium (W3C) in Section 2.11 of [R20081126]. If this guidance is followed, no canonicalization is needed.

A robust signature generation process MAY perform canonicalization to ensure that the W3C guidance has been followed. This guidance says that a <LF> character MUST be used to denote the end of a line of text within an XML file. Therefore, any two-character <CRLF> sequence and any <CR> that is not followed by <LF> are to be translated to a single <LF> character.

7.3. No Canonicalization of Other File Formats

No canonicalization is needed for file formats currently used or planned other than ASCII, UTF-8, HTML, and XML files. Other file formats, including PDF [PDF], PostScript [PS], and EPUB [EPUB] are treated as a simple sequence of octets by the digital signature algorithm.

8. Standalone Use

An RTA MAY include the set of certificates and CRL which permit the RTA and the object which has been signed to be validated cryptographically given a set of applicable trust anchors. The set of certificates and CRLs must form a complete path from a trust anchor to each end-entity certificate used to sign.

No publication protocol is specified, or expected. RTA objects are standalone, and intended to be exchanged freely as attachments to email or lodged in the web, or other mechanisms.

The EE certificates generated and used to sign MAY omit the Subject Information Access (SIA) extension mandated by RFC 6487 as that extension requires an rsync URI for the accessLocation form and the RTA method does not require repository access via rsync.

An RTA and its associated EE certificates MAY appear on an RPKI Manifest and MAY be published in a repository.

9. IANA Considerations

IANA is entirely off the hook on this one.

10. Security Considerations

Security is explicitly a consideration in the whole of this draft.

The intent is to make testable digital signatures over data to associate the data with specific INR.

- If the private key of any RPKI certificate leaks, anyone could in theory make signatures.

- The applicability of the INR to the INR in the data is not specified. Validity is taken to mean the cryptographic validity of the certification chains, and associated signatures. The applicability of the specific RFC3779 resources to the signed data is out of scope.

- Given the lack of constraint on signed objects, there is no intention to have the signed object placed in a repository or appear on a manifest, or in any other way interfere with the operations of the distributed RPKI system. RTA objects themselves may appear in repositories, and are constrained in size to the ASN.1 encoded burden of the set of certificates which are sufficient to describe the RFC3779 resources associated with the signatures.

- By design, each signing party signs the RTA object discretely. Since the RTA object includes the set of subjectKeyIdentifiers there is partial closure over the question "who agrees to sign" since the object is only valid if the set of signing parties matches the list of expected signing keys. However, in principle a sub-set (down to one) of these signing parties can assert an RTA which specifies only that subset, or itself solely to sign, and make a valid RTA which cannot be disproved. Since the RTA can only refer to RFC3779 data which is within scope of the set of signers, the impact of this is to refine (narrow down) the relevant set of internet resources which can relate to the (detached) signed object. However, this places the burden of semantic validation of the meaning of those resources, contextually, on the consumer. Caveat Emptor.

11. Acknowledgments

Russ Housely advised informally on the use of CMS signed objects around 2012.

Russ's work on CMS signed internet drafts in [RFC8358] and [RFC5485] has been re-purposed here to apply to arbitrary signed objects, not just internet-drafts and text documents.

An early implementation of RTA was coded by Robert Loomans and Gary Kennedy at APNIC before 2011 which used simpler ASN.1 semantics to specify the signed object.

Jamie Gillespie (APNIC) provided valuable feedback and critique of the 00 draft.

12. Revision history

- 00 draft initial upload from older text, inclusion of CMS references.

- 01 draft explicit language for the lack of repository references, use of CRL, spellcheck nits.

13. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <https://www.rfc-editor.org/info/rfc3779>.

[RFC5485] Housley, R., "Digital Signatures on Internet-Draft Documents", RFC 5485, DOI 10.17487/RFC5485, March 2009, <https://www.rfc-editor.org/info/rfc5485>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

[RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <https://www.rfc-editor.org/info/rfc6488>.

[RFC7935] Huston, G. and G. Michaelson, Ed., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure", RFC 7935, DOI 10.17487/RFC7935, August 2016, <https://www.rfc-editor.org/info/rfc7935>.

[RFC8358] Housley, R., "Update to Digital Signatures on Internet-Draft Documents", RFC 8358, DOI 10.17487/RFC8358, March 2018, <https://www.rfc-editor.org/info/rfc8358>.

Authors' Addresses

George G. Michaelson
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane, QLD 4101
Australia

Email: ggm@apnic.net

Geoff Huston
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane, QLD 4101
Australia

Email: gih@apnic.net

Tom Harrison
Asia Pacific Network Information Centre
6 Cordelia St
South Brisbane, QLD 4101
Australia

Email: tomh@apnic.net

Tim Bruijnzeels
Open Netlabs B.V.
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: timb@opennetlabs.nl

Martin Hoffmann
Open Netlabs B.V.
Science Park 400
Amsterdam 1098 XH
The Netherlands

Email: martin@nlnetlabs.nl