Workgroup: sidrops
Published: 21 March 2024
Intended Status: Standards Track
Expires: 22 September 2024
Authors: J. Snijders Fastly
G. Huston APNIC

A profile for Signed Prefix Lists for Use in the Resource Public Key Infrastructure (RPKI)

Abstract

This document defines a "Signed Prefix List", a Cryptographic Message Syntax (CMS) protected content type for use with the Resource Public Key Infrastructure (RPKI) to carry the complete list of prefixes which an Autonomous System (the subject AS) may originate to all or any of its routing peers. The validation of a Signed Prefix List confirms that the holder of the subject AS produced the object, and that this list is a current, accurate and complete description of address prefixes that may be announced into the routing system originated by the subject AS.

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 22 September 2024.

document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

This document defines a "Signed Prefix List", a Cryptographic Message Syntax (CMS) [RFC5652] [RFC6268] protected content type to carry a list of IP address prefixes and an Autonomous System Number (the subject AS). The list of prefixes describes the maximal set of prefixes that the subject AS MAY announce to any of its routing peers. The content is signed by the holder of the RPKI private key associated with the subject AS.

RPKI Signed Prefix Lists allow other RPKI-validating routing entities to audit the collection of announcements that have the subject AS as the originating AS. Any prefixes originated by this AS not contained in a validated Signed Prefix List **SHOULD** be regarded as ineligible, but ultimately their consequent handling by the local routing entity that performed the audit function is a matter of local policy.

The intent of this object is to offer a RPKI-based successor to the [RFC2622] 'route-set' class objects used in Internet Routing Registries (IRRs). The semantics of the route-set and the Signed Prefix List are similair. The difference is that the RPKI signature allows a relying party of be assured of the currency and authenticity of the Signed Prefix List as a complete enumeration of all prefixes that may be announced as originating by the subject AS.

Signed Prefix List objects follow the Signed Object Template for the RPKI [RFC6488].

1.1. Requirements Language

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. The Signed Prefix List ContentType

The eContentType for a Prefix List is defined as id-ct-rpkiSignedPrefixList, with Object Identifier (OID) 1.2.840.113549.1.9.16.1.51.

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

3. The Signed Prefix List eContent

The content of a Signed Prefix List is a single ASN and a list of IP address prefixes. A Signed Prefix List is formally defined as follows:

```
RpkiSignedPrefixList-2024
  { iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs9(9) smime(16) mod(0)
    id-mod-rpkiSignedPrefixList-2024(TBD) }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS
  CONTENT-TYPE
  FROM CryptographicMessageSyntax-2010 -- in [RFC6268]
    { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
      pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) } ;

ct-rpkiSignedPrefixList CONTENT-TYPE ::=
  { TYPE RpkiSignedPrefixList
    IDENTIFIED BY id-ct-rpkiSignedPrefixList }

id-ct-rpkiSignedPrefixList OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) id-smime(16) id-ct(1) 51 }

RpkiSignedPrefixList ::= SEQUENCE {
  version [0]     INTEGER (0..MAX) DEFAULT 0,
  asID            INTEGER (1..4294967295),
  prefixBlocks    SEQUENCE (SIZE(0..2)) OF AddressFamilyAddressPrefixes

AddressFamilyAddressPrefixes ::= SEQUENCE {
  addressFamily   ADDRESS-FAMILY.&afi ({AddressFamilySet}),
  addressPrefixes ADDRESS-FAMILY.&Prefixes ({AddressFamilySet}{@addressF

ADDRESS-FAMILY ::= CLASS {
  &afi          OCTET STRING (SIZE(2)) UNIQUE,
  &Prefixes
} WITH SYNTAX { AFI &afi PREFIXES &Prefixes }

AddressFamilySet ADDRESS-FAMILY ::= { addressFamilyIPv4 | addressFamilyI

addressFamilyIPv4 ADDRESS-FAMILY ::= { AFI afi-IPv4 PREFIXES IPv4Prefixe
addressFamilyIPv6 ADDRESS-FAMILY ::= { AFI afi-IPv6 PREFIXES IPv6Prefixe

afi-IPv4 OCTET STRING ::= '0001'H
afi-IPv6 OCTET STRING ::= '0002'H

IPv4Prefixes ::= SEQUENCE (SIZE(1..MAX)) OF AddressPrefix{ub-IPv4}
IPv6Prefixes ::= SEQUENCE (SIZE(1..MAX)) OF AddressPrefix{ub-IPv6}

ub-IPv4 INTEGER ::= 32
ub-IPv6 INTEGER ::= 128
```

```
AddressPrefix {INTEGER: ub} ::= BIT STRING (SIZE(0..ub))

END
```

3.1. Version

The version number of the RpkiSignedPrefixList **MUST** be 0.

3.2. asID

The Autonomous System Number contained here **MUST** be a contained within the set of AS Identifier resources listed by the EE certificate carried in the CMS certificates field.

3.3. prefixBlocks

This field contains a SEQUENCE of AddressFamilyAddressPrefixes. The AddressFamilyAddressPrefixes elements **MUST** be ordered in ascending order by numeric value of the addressFamily field.

3.3.1. Element AddressFamilyAddressPrefixes

This field contains a SEQUENCE which contains one instance of addressFamily and one instance of addressPrefixes.

3.3.1.1. addressFamily

This field contains an OCTET STRING which is either '0001'H (IPv4) or '0002'H (IPv6).

3.3.1.2. addressPrefixes

This field contains a SEQUENCE of parameterized AddressPrefix instances.

3.3.1.3. Element AddressPrefix

This element is length bounded through the Information Object Class ADDRESS-FAMILY. The type is a BIT STRING, see Section 2.2.3.8 of [RFC3779] for more information on encoding IP prefixes.

3.3.2. Canonical form for prefixBlocks

As the data structure described by the SignedPrefixList Section 3 module allows for many different ways to represent the same set of IP address prefix information, a canonical form is defined such that every set of address prefixes has a unique representation. To produce and verify this canonical form, the process described in this section **MUST** be used to ensure information elements are unique with respect to one another and sorted in ascending order. This

canonicalization procedure builds upon the canonicalization procedure specified in section 2.2.3.6 of [RFC3779] and Section 4.3.3 of [I-D.ietf-sidrops-rfc6482bis].

To semantically compare, sort, and deduplicate the contents of the prefixBlocks field, each AddressPrefix element is mapped to an abstract data element composed of three integer values:

afi The AFI value appearing in the addressFamily field of the containing addressPrefixes as an integer.

addr The first IP address of the IP prefix appearing in the AddressPrefix field, as a 32-bit (IPv4) or 128-bit (IPv6) integer value.

plen The prefix length of the IP prefix appearing in the AddressPrefix address field as an integer value.

Thus, the equality or relative order of two AddressPrefix elements can be tested by comparing their abstract representations.

3.3.2.1. Comparator

The set of prefixBlocks is totally ordered. The order of two prefixBlocks is determined by the first non-equal comparison in the following list.

1. Data elements with a lower afi value precede data elements with a higher afi value.

2. Data elements with a lower addr value precede data elements with a higher addr value.

3. Data elements with a lower plen value precede data elements with a higher plen value.

Data elements for which all three values compare equal are duplicates of one another.

4. Semantics of Signed Prefix List

The IP address prefixes listed in a Signed Prefix List object are an enumeration of prefixes that may be announced as originating from the AS identified by the asID (the subject AS) if the object can be validated by the RPKI (Section 5). The object does not implicitly permit a more-specific prefix subsumed by a listed IP address prefix to be originated by this AS. For any such more-specific prefix to be permitted by the Signed Prefix List object, it must be explicitly listed in the list of IP address prefixes.

5. Signed Prefix List Validation

To validate a Signed Prefix List, the RP **MUST** perform all the validation checks specified in [RFC6488]. In addition, the RP **MUST** perform the following validation steps:

1. The contents of the CMS eContent field **MUST** conform to all the constraints described in Section 3.

2. The Autonomous System Identifier Delegation extension [RFC3779] **MUST** be present in the EE certificate contained in the CMS certificates field.

3. The AS identifier present in the RpkiSignedPrefixList eContent 'asID' field **MUST** be contained in the AS Identifiers present in the certificate extension.

4. The Autonomous System Identifier Delegation extension **MUST NOT** contain "inherit" elements.

5. The IP Address Delegation Extension [RFC3779] is not used in Signed Prefix List, and **MUST NOT** be present in the EE certificate.

6. Operational Considerations

Multiple valid Signed Prefix List objects which contain the same asID could exist. In such cases, the union of address prefix members of the collection of Signed Prefrix list objects forms the complete set of members. It is **RECOMMENDED** that a CA maintains a single Signed Prefix List for a given asID.

If an AS holder publishes a Signed Prefix List, then relying parties **SHOULD** assume that the list is complete for that originating AS, and the presence of any route with the same AS as the originating AS and an address prefix that is not included in the Signed Prefix List implies that the route has been propagated within the routing system without the permission of the originating AS.

The construction of an 'allowlist' for a given EBGP session using Signed Prefix List(s) compliments both best current practices [RFC7454] and the practice of rejecting RPKI-ROV-invalid BGP route announcements [RFC6811]. In other words, if a given BGP route is covered by a Signed Prefix List, but also is "Invalid" from a Route Origin Validation perspective, it is RECOMMENDED to reject the route announcement. Here the term "reject the route" is used in the sense of "consider the route ineligible for path selection" [RFC4271].

6.1. EE Certificates

The Certificate Authority (CA) **SHOULD** sign only one Signed Prefix List with each generated private key and **SHOULD** generate a new key pair for each new version of a Signed Prefix List object. The CA **MUST** generate a new End Entity (EE) certificate for each signing of a particular Signed Prefix List. An associated EE certificate used in this fashion is termed a "one-time-use" EE certificate (see Section 3 of [RFC6487]).

6.2. Object Filenames

A guideline for naming Signed Prefix List objects is that the file name chosen in the repository be a value derived from the public key of the EE certificate. One such method of generating a publication name is described in Section 2.1 of [RFC4387]; convert the 160-bit hash of an EE's public key value into a 27-character string using a modified form of Base64 encoding, with an additional modification as proposed in Section 5, table 2, of [RFC4648].

7. Security Considerations

Relying Parties are warned that the data in a Signed Prefix List is self-asserted by the AS holder. There is no implied authority in a Signed Prefix List that any IP prefix holder has granted the AS permission to originate a route for any of the listed prefixes. Such an authority is separately conveyed in the RPKI as a ROA.

While one-time-use EE certificates and their associated key pairs are supposed to be used in an ephemeral manner; CAs are not technically restricted from generating and signing multiple different objects with the same key pair, or using the same EE certificate for different objects. Any RPKI objects, including Signed Prefix List objects, that share the same EE certificate cannot be revoked individually.

8. IANA Considerations

8.1. SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)

IANA has temporarily allocated the following in the "SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1)" registry:

Decimal	Description	References
51	id-ct-rpkiSignedPrefixList	draft-ietf-sidrops-rpki-prefixlist

Table 1

8.2. RPKI Signed Objects

IANA is requested to register the following OID in the "RPKI Signed Objects" registry [RFC6488]:

Name	OID	Reference
Signed Prefix List	1.2.840.113549.1.9.16.1.51	draft-ietf-sidrops-rpki-prefixlist

Table 2

8.3. RPKI Repository Name Schemes

IANA is requested to add the Signed Prefix List file extension to the "RPKI Repository Name Schemes" registry [RFC6481] as follows:

Filename Extension	RPKI Object	Reference
.spl	Signed Prefix List	draft-ietf-sidrops-rpki-prefixlist

Table 3

8.4. SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)

IANA is requested to allocate the following in the "SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0)" registry:

Decimal	Description	References
TBD	id-mod-rpkiSignedPrefixList-2024	draft-ietf-sidrops-rpki-prefixlist

Table 4

8.5. Media Types

IANA is requested to register the media type "application/rpki-prefixlist" in the "Media Types" registry as follows:

8.5.1. Signed Prefix List Media Type

Type name: application
Subtype name: rpki-prefixlist
Required parameters: N/A
Optional parameters: N/A
Encoding considerations: binary
Security considerations: Carries a Signed Prefix List. This media type contains no active content. See Section 5 of draft-ietf-sidrops-rpki-prefixlist for further information.
Interoperability considerations: N/A
Published specification: draft-ietf-sidrops-rpki-prefixlist

Applications that use this media type: RPKI operators
Fragment identifier considerations: N/A
Additional information:

Content:
This media type is a signed object, as defined in [RFC6488], which contains a list of prefixes as defined in draft-ietf-sidrops-rpki-prefixlist.
Magic number(s): N/A
File extension(s): .spl
Macintosh file type code(s): N/A
Person & email address to contact for further information: Job Snijders (job@fastly.com)
Intended usage: COMMON
Restrictions on usage: N/A
Author: Job Snijders (job@fastly.com)
Change controller: IETF

9. References

9.1. Normative References

[I-D.ietf-sidrops-rfc6482bis] Snijders, J., Maddison, B., Lepinski, M., Kong, D., and S. Kent, "A Profile for Route Origin Authorizations (ROAs)", Work in Progress, Internet-Draft, draft-ietf-sidrops-rfc6482bis-09, 14 December 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-rfc6482bis-09>.

[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>.

[RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, DOI 10.17487/RFC2622, June 1999, <https://www.rfc-editor.org/info/rfc2622>.

[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>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

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

[RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <https://www.rfc-editor.org/info/rfc6481>.

[RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <https://www.rfc-editor.org/info/rfc6487>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[RFC4387] Gutmann, P., Ed., "Internet X.509 Public Key Infrastructure Operational Protocols: Certificate Store Access via HTTP", RFC 4387, DOI 10.17487/RFC4387, February 2006, <https://www.rfc-editor.org/info/rfc4387>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC6268] Schaad, J. and S. Turner, "Additional New ASN.1 Modules for the Cryptographic Message Syntax (CMS) and the Public Key Infrastructure Using X.509 (PKIX)", RFC 6268, DOI 10.17487/RFC6268, July 2011, <https://www.rfc-editor.org/info/rfc6268>.

[RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <https://www.rfc-editor.org/info/rfc6811>.

[RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <https://www.rfc-editor.org/info/rfc7454>.

[rpki-client]
Jeker, C., Dzonsons, K., Buehler, T., and J. Snijders, "rpki-client", February 2024, <https://www.rpki-client.org/>.

Appendix A. Acknowledgements

The authors wish to thank Russ Housley, Theo Buehler, Sriram Kotikalapudi, and Ties de Kock for much appreciated feedback.

Appendix B. Example payloads

B.1. Example Signed Prefix List eContent Payload

Below an example of a DER-encoded Signed Prefix List eContent is provided with annotation following the '#' character.

```
$ cat << EOF | xxd -r -ps | openssl asn1parse -inform DER -i -dump
3081b102023cca3081aa307304020001306d03040043ddf5030400a5fee1
030506a5feff00030400c093a8030400c22047030400c63a03030401cc02
1e030400d11800030400d11801030400d11803030402d11804030403d118
08030400d11808030400d11809030404d11810030405d11820030406d118
40030407d11880303304020002302d03070120010418144e030700200106
7c208c030700200107fbfd040307002607fae002450307002a0eb2400000
EOF
    0:d=0  hl=3 l= 177 cons: SEQUENCE
    3:d=1  hl=2 l=   2 prim:  INTEGER       :3CCA  # AS 15562
    7:d=1  hl=3 l= 170 cons:  SEQUENCE
   10:d=2  hl=2 l= 115 cons:   SEQUENCE
   12:d=3  hl=2 l=   2 prim:    OCTET STRING
      0000 - 00 01                                  # AFI IPv4
   16:d=3  hl=2 l= 109 cons:    SEQUENCE
   18:d=4  hl=2 l=   4 prim:     BIT STRING
      0000 - 00 43 dd f5                            # 67.221.245.0/24
   24:d=4  hl=2 l=   4 prim:     BIT STRING
      0000 - 00 a5 fe e1                            # 165.254.225.0/24
   30:d=4  hl=2 l=   5 prim:     BIT STRING
      0000 - 06 a5 fe ff                            # 165.254.255.0/26
      0005 - <SPACES/NULS>
... snip ...
  127:d=2  hl=2 l=  51 cons:   SEQUENCE
  129:d=3  hl=2 l=   2 prim:    OCTET STRING
      0000 - 00 02                                  # AFI IPv6
  133:d=3  hl=2 l=  45 cons:    SEQUENCE
  135:d=4  hl=2 l=   7 prim:     BIT STRING
      0000 - 01 20 01 04 18 14 4e                   # 2001:418:144e::/47
  144:d=4  hl=2 l=   7 prim:     BIT STRING
      0000 - 00 20 01 06 7c 20 8c                   # 2001:67c:208c::/48
... snip ...
```

Appendix C. Implementation status

This section is to be removed before publishing as an RFC.

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available

implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

*Example .spl files were created by Job Snijders.

*A validator implementation [rpki-client], written in C was provided by Job Snijders from Fastly.

Authors' Addresses

Job Snijders
Fastly
Amsterdam
Netherlands

Email: job@fastly.com

Geoff Huston
APNIC
6 Cordelia St
South Brisbane QLD 4101
Australia

Email: gih@apnic.net