Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 3, 2012

R. Gagliano
Cisco Systems
S. Kent
BBN Technologies
S. Turner
IECA, Inc.

August 2, 2011

Algorithm Agility Procedure for RPKI. draft-ietf-sidr-algorithm-agility-03

Abstract

This document specifies the process that Certification Authorities (CAs) and Relying Parties (RP) participating in the Resource Public Key Infrastructure (RPKI) will need to follow to transition to a new (and probably cryptographically stronger) algorithm set. The process is expected to be completed in a time scale of months or years. Consequently, no emergency transition is specified. The transition procedure defined in this document supports only a top-down migration (parent migrates before children).

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 February 3, 2012.

Copyright Notice

Table of Contents

1. Requirements notation

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].

2. Introduction

The RPKI must accommodate transitions between the public keys used by CAs. Transitions of this sort are usually termed "key rollover". Planned key rollover will occur at regular intervals throughout the life of the RPKI, as each CA changes its public keys, in a non-coordinated fashion. (By non-coordinated we mean that the time at which each CA elects to change its keys is locally determined, not coordinated across the RPKI.) Moreover, because a key change might be necessitated by suspected private key compromise, one can never assume coordination of these events among all of the CAs in the RPKI. In an emergency key rollover, the old certificate is revoked and a new certificate with a new key is issued. The mechanisms to perform a key rollover in RPKI (either planned or in an emergency), while maintaining the same algorithm suite, are covered in [I-D.ietf-sidr-keyroll].

This document describes the mechanism to perform a key rollover in RPKI due to the migration to a new signature algorithm suite. A signature algorithm suite encompasses both a signature algorithm (with a specified key size range) and a one-way hash algorithm. It is anticipated that the RPKI will require the adoption of updated key sizes and/or different algorithm suites over time. This document treats the adoption of a new hash algorithm while retaining the current signature algorithm as equivalent to an algorithm migration, and requires the CA to change its key. Migration to a new algorithm suite will be required in order to maintain an acceptable level of cryptographic security and protect the integrity of certificates, CRLs and signed objects in the RPKI. All of the data structures in the RPKI explicitly identify the signature and hash algorithms being used. However, experience has demonstrated that the ability to represent algorithm IDs is not sufficient to enable migration to new algorithm suites (algorithm agility). One also must ensure that protocols, infrastructure elements, and operational procedures also accommodate migration from one algorithm suite to another. Algorithm migration is expected to be very infrequent, but it also will require support of a "current" and "next" suite for a prolonged interval, probably several years.

This document defines how entities in the RPKI execute (planned) CA key rollover when the algorithm suite changes. The description covers actions by CAs, repository operators, and RPs. It describes the behavior required of both CAs and RPs to make such key changes work in the RPKI context, including how the RPKI repository system is used to support key rollover.

This document does not specify any algorithm suite.

This document does not specify any algorithm suite per se. The RPKI Certificate Policy (CP) [I-D.ietf-sidr-cp] mandates the use of the algorithms defined in [I-D.ietf-sidr-rpki-algs] by CAs and RPs. When an algorithm transition is initiated, [I-D.ietf-sidr-rpki-algs] will be updated (as defined in Section 4.1 of this document) redefining the required algorithm(s) for compliant RPKI CAs and RPs under the CP.

3. Terminology

This document assumes 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 "A Profile for Resource Certificate Repository Structure" [I-D.ietf-sidr-repos-struct]. Additional terms and conventions use din examples are provided below.

Algorithm migration A planned transition from one signature and hash algorithm to a new signature and hash algorithm.

Algorithm Suite A The "current" algorithm suite used for hashing and signing, in examples in this document

Algorithm Suite B The "next" algorithm suite used for hashing and signing, used in examples in this document

Algorithm Suite C The "old" algorithm suite used for hashing and signing, used in examples in this document

CA X The CA that issued CA Y's certificate (i.e., CA Y's parent), used in examples this document.

CA Y The CA that is changing keys and/or algorithm suites, used in examples this document

CA Z A CA that is a "child" of CA Y, used in examples this document

Certificate re-issuance (unilateral) A CA MAY reissue a certificate to a subordinate Subject without the involvement of the Subject. The public key, resource extensions, and most other fields are copied from the current Subject certificate into the next Subject certificate. The Issuer name MAY change, if necessary to reflect the Subject name in the CA certificate under which the reissued certificate will be validated. The validity interval also MAY be changed. This action is defined as a unilateral certificate re-issuance.

Non-Leaf CA A CA that issues certificates to other CAs is a non-leaf CA.

Leaf CA A leaf CA is a CA that issues only EE certs.

PoP (proof of possession) Execution of a protocol that demonstrates to an issuer that a subject requesting a certificate possesses the private key corresponding to the public key in the certificate submitted by the subject.

Signed Product Set (or Set) A collection of certificates, signed objects, a CRL and a manifest that are associated by virtue of being verifiable under the same parent CA certificate

4. Key Rollover steps for algorithm migration

The "current" RPKI algorithm suite (Suite A) is defined in the RPKI's CP document, by reference to [I-D.ietf-sidr-rpki-algs]. When a migration of the RPKI algorithm suite is needed, the first step MUST be an update of the [I-D.ietf-sidr-rpki-algs] document that will include all the information described in Section 4.3.

4.1. Milestones definition

CA Ready Algorithm B Date - After this date, all (non-leaf) CAs MUST be ready to process a request from a child CA to issue a certificate under the Algorithm B suite.

CA Go Algorithm B Date - After this date, all (non-leaf) CAs MUST have re-issued all of its signed product set under the Algorithm B suite.

RP Ready Algorithm B Date - After this date, all RPs MUST be prepared to process signed material issued under the Algorithm B suite.

Twilight Algorithm B - After this date, a CA MAY cease issuing signed products under the Algorithm A suite. Also, after this date, a RP MAY cease to validate signed materials issued under the Algorithm A suite.

End Of Life (EOL) Algorithm A - After this date every CA MUST NOT generate certificates, CRLs, or other RPKI signed objects under the Algorithm A suite. Also, after this date, no RP SHOULD accept as valid any certificate, CRL or signed object using the Algorithm A suite.

4.2. Process overview

The migration process described in this document involves a series of steps that MUST be executed in chronological order by CAs and RPs. The only milestone at which both CAs and RPs take action at the same moment is the "EOL Algorithm A" date. Due to the decentralized nature of the RPKI infrastructure, it is expected that the process will take several months or even years.

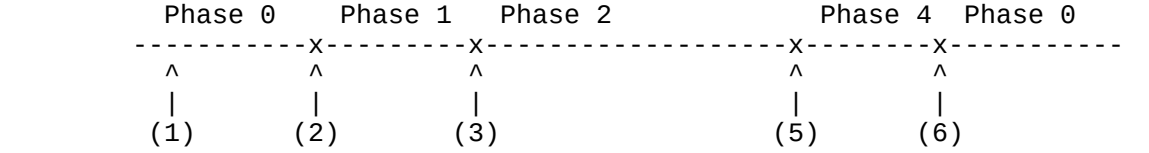
In order to facilitate the transition, CAs will start issuing certificates using the Algorithm B in a hierarchical top-down order. In our example, CA Y will issue certificates using the Algorithm B suite only after CA X has started to do so (CA Y Ready Algorithm B Date > CA X Ready Algorithm B Date). This ordered transition avoids issuance of "mixed" suite certificates, e.g., a CA certificate signed

using Suite A, containing a key from Suite B. In the RPKI, a CA MUST NOT sign a CA certificate carrying a subject key that corresponds to an algorithm suite that differs from the one used to sign the certificate.

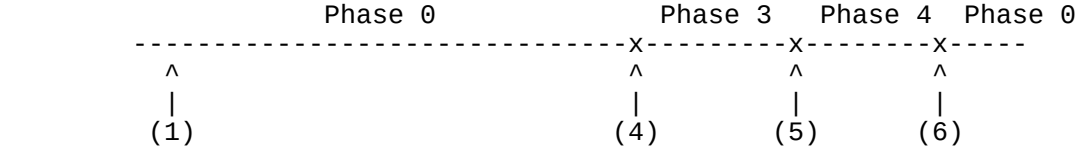
The algorithm agility model described here does not prohibit a CA issuing an EE certificate with a subject public key from a different algorithm suite, if that certificate is not used to verify repository objects. This exception to the mixed algorithm suite certificate rule is allowed because an EE certificate that is not used to verify repository objects does not interfere with the ability of RPs to download and verify repository content. Nonetheless, every CA in the RPKI is required to perform a Proof of Possession (PoP) check for the subject public key when issuing a certificate. In general a subject cannot assume that a CA is capable of supporting a different algorithm. However, if the subject is closely affiliated with the CA, it is reasonable to assume that there are ways for the subject to know whether the CA can support a request to issue an EE certificate containing a specific, different public key algorithm. This document does not specify how a subject can determine whether a CA is capable of issuing a mixed suite EE certificate, because it anticipates that such certificates will be issued only in contexts where the subject and CA are sufficiently closely affiliated (for example, an ISP issuing certificates to devices that it manages).

The following figure gives an overview of the process:

Process for RPKI CAs:



Process for RPKI RPs:



(1) RPKI's algorithm document updated.
(2) CA Ready Algorithm B Date
(3) CA Go Algorithm B Date
(4) RP Ready Algorithm B Date
(5) Twilight Date

(6) End Of Live (EOL) Date

4.3. Phase 0

Phase 0 is the initial phase of the process, throughout this phase the algorithm suite A is the only supported algorithm suite in RPKI.

The first milestone, which will initiate the migration process, is updating the [I-D.ietf-sidr-rpki-algs] document with the following definitions for the RPKI:

- Algorithm Suite A
- Algorithm Suite B
- CA Ready Algorithm B Date
- CA Go Algorithm B Date
- RP Ready Algorithm B Date
- Twilight Date
- EOL Date

All Dates MUST be represented using the local UTC date-time format specified in [RFC3339].

As an example, during Phase 0, CAs X, Y and Z are required to generate signed product sets using only the Algorithm Suite A. Also, RPs are required to validate signed product sets issued using only Algorithm Suite A.

```
CA X-Certificate-Algorithm-Suite-A (Cert-XA)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A
```

Note: Cert-XA represent the certificate for CA X, that is signed using the algorithm suite A.

4.4. Phase 1

Phase 1 starts at the CA Ready Algorithm B Date. During Phase 1, all (non-leaf) CAs MUST be ready to process a request from a child CA to issue or revoke a certificate using the Algorithm B suite.

As the transition will happen using a (hierarchic) top-down model, a child CA will be able to issue certificates using the Algorithm B suite only after its parent CA has issued its own. The RPKI provisioning protocol can identify if a parent CA is capable of issuing certificates using the Algorithm Suite B, and can identify the corresponding algorithm suite in each Certificate Signing Request (see Section 5).

The following figure shows the status of repository entries for the three example CAs during this Phase. Two distinct certificate chains are maintained and CA Z has not yet requested any material using the Algorithm B suite.

```
CA X-Certificate-Algorithm-Suite-A (Cert-XA)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A

CA X-Certificate-Algorithm-Suite-B (Cert-XB)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-B (Cert-YB)
                |-> CA-Y-CRL-Algorithm-Suite-B (CRL-YB)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-B
        |-> CA-X-CRL-Algorithm-Suite-B (CRL-XB)
        |-> CA-X-Signed-Objects-Algorithm-Suite-B
```

4.5. Phase 2

Phase 2 starts at the CA Go Algorithm B Date. At the start of this phase, all signed product sets MUST be available using both Algorithm Suite A and Algorithm Suite B. During this phase, RPs MUST be prepared to validate sets issued using Algorithm Suite A and MAY be prepared to validate sets issued using the Algorithm Suite B.

An RP that validates all signed product sets using both Algorithm Suite A or Algorithm Suite B, SHOULD expect the same results.

However, an object that validates using either Algorithm Suite A or Algorithm Suite B MUST be considered valid. A detailed analysis on the validation of multiple instance of signed objects is included in Section 6.

The following figure shows the status of the repository entries for the three example CAs throughout this phase, where all signed objects are available using both algorithm suites.

```
CA X-Certificate-Algorithm-Suite-A (Cert-XA)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A

CA X-Certificate-Algorithm-Suite-B (Cert-XB)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-B (Cert-YB)
                |-> CA-Z-Certificate-Algorithm-Suite-B (Cert-ZB)
                        |-> CA-Z-CRL-Algorithm-Suite-B (CRL-ZB)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-B
                |-> CA-Y-CRL-Algorithm-Suite-B (CRL-YB)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-B
        |-> CA-X-CRL-Algorithm-Suite-B (CRL-XB)
        |-> CA-X-Signed-Objects-Algorithm-Suite-B
```

4.6. Phase 3

Phase 3 starts at the RP Ready Algorithm B Date. During this phase, all signed product sets are available using both algorithm suites and all RPs MUST be able to validate them using either suite. An object that validates using either Algorithm Suite A or Algorithm Suite B MUST be considered as valid. It is RECOMMENDED that RPs utilize only Suite B for validation throughout this phase, in preparation for Phase 4.

There are no changes to the CA behavior throughout this phase.

4.7. Phase 4

Phase 4 starts at the Algorithm A Twilight Date. At that date, the Algorithm A is labeled as "old" and the Algorithm B is labeled as "current":

```
Before Twilight                 -->     After Twilight

Algorithm Suite A ("current")   -->     Algorithm Suite C ("old")
Algorithm Suite B ("new")       -->     Algorithm Suite A ("current")
```

During this phase, all signed product sets MUST be issued using Algorithm Suite A (formerly B) and MAY be issued using Algorithm Suite C (formerly A). All signed products sets issued using Suite A MUST be published at their corresponding publication points, but signed products sets issued using Suite C MAY be published at their corresponding publication points. Also, every RP MUST validate signed product sets using Suite A but also MAY validate signed product sets using Suite C.

The following figure describe a possible status for the repositories of the example CAs. In this case, CA Z no longer issues signed products using the Algorithm Suite C.

```
CA X-Certificate-Algorithm-Suite-C (Cert-XC)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-C (Cert-YC)
                |-> CA-Y-CRL-Algorithm-Suite-C (CRL-YC)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-C
        |-> CA-X-CRL-Algorithm-Suite-C (CRL-XC)
        |-> CA-X-Signed-Objects-Algorithm-Suite-C

CA X-Certificate-Algorithm-Suite-A (Cert-XA)
        |
        |-> CA-Y-Certificate-Algorithm-Suite-A (Cert-YA)
                |-> CA-Z-Certificate-Algorithm-Suite-A (Cert-ZA)
                        |-> CA-Z-CRL-Algorithm-Suite-A (CRL-ZA)
                        |-> CA-Z-Signed-Objects-Algorithm-Suite-A
                |-> CA-Y-CRL-Algorithm-Suite-A (CRL-YA)
                |-> CA-Y-Signed-Objects-Algorithm-Suite-A
        |-> CA-X-CRL-Algorithm-Suite-A (CRL-XA)
        |-> CA-X-Signed-Objects-Algorithm-Suite-A
```

4.8. Return to Phase 0

Phase 0 starts at the EOL Algorithm Date. At this phase, ALL signed product sets using Algorithm Suite C MUST be considered invalid. CAs MUST neither issue nor publish signed products using Algorithm Suite C.

This phase closes the loop as Algorithm Suite A is the only required algorithm suite in RPKI.

5. Multi Algorithm support in the RPKI provisioning protocol

The migration described in this document is a top-down process, where two synchronization issues need to be solved between child and parent CAs:

- A child CA needs to identify which algorithm suites are supported by its parent CA
- A child CA needs to identify which algorithm suite should be used to sign a Certificate Signing Request (CSR)

The RPKI provisioning protocol [I-D.ietf-sidr-rescerts-provisioning] supports multiple algorithms suites by implementing a different resource classes for each suite. Several different resource classes also may use the same algorithm suite for different resource sets.

A child CA that wants to identify which algorithm suites are supported by its parent CA MUST perform the following tasks:

1. Establish a provisioning protocol session with its parent CA
2. Perform a "list" command as described in Section 3.3.1 of [I-D.ietf-sidr-rescerts-provisioning]
3. From the Payload in the "list response" resource class, extract the "issuer's certificate" for each class. The Algorithm Suite for each class will match the Algorithm Suite used to issue the corresponding "issuer's certificate".

A child CA that wants to specify an Algorithm Suite to its parent CA (e.g., in a certificate request) MUST perform the following tasks:

1. Perform the tasks to identify the resource class for each Algorithm Suite supported by its parent CA (as above).
2. Identify the corresponding resource class in the appropriate provisioning protocol command (e.g. "issue" or "revoke")

Upon receipt of a certificate request from a child CA, a parent CA will verify the PoP of the private key. If a child CA requests issuing a certificate using an algorithm suite that does not match a resource class, the PoP validation will fail and the request will not be performed.

6. Validation of multiple instance of signed products

During Phases 1,2,3 and 4, two algorithm suites will be valid simultaneously in RPKI. In this section, we describe the RP behavior when validating instances of the same signed product but signed with different algorithm suites. As a general rule, the validation of signed products using different algorithm suites are independent and the RP MUST NOT keep any relationship between the different hierarchies.

During Phase 1 two (corresponding) files for an object MAY be available for each signed product, one signed under Algorithm Suite A and one under Algorithm Suite B. When an RP validates these signed products, if either instance of an object validates, the product is accepted. A failure to validate one instance of a product, under either algorithm Suite MUST NOT cause the RP to reject the other instance of the product. Because both instances of such products MUST contain the same resources, relying on either instance will yield the same outcome.

During Phases 2 and 3 of this process, two (corresponding) instances of all signed products MUST be available to RPs. As in Phase 1, when an RP validates these signed products, if either instance validates, the product is accepted. A failure to validate one instance of a product, under either algorithm Suite MUST NOT cause the RP to reject the other instance of the product. Also, as above, if only one instance of a signed product can be validated, subordinate products issued under the other (non-validated) algorithm suite cannot be used, and thus SHOULD NOT be processed (or even retrieved).

During Phase 4 two (corresponding) files for an object MAY be available for each signed product, one signed under Algorithm Suite A and one under Algorithm Suite C. When an RP validates these signed products, if either instance of an object validates, the product is accepted. A failure to validate one instance of a product, under either algorithm Suite MUST NOT cause the RP to reject the other instance of the product. Because both instances of such products MUST contain the same resources, relying on either instance will yield the same outcome.

7. Revocations

As the algorithm migration process mandates the maintenance of two parallel certificate hierarchies, revocations requests for each algorithm suite MUST be handled independently. A Child CA MUST request revocation of a certificate relative to a specific algorithm suite.

During phase 2 and phase 3, the two parallel certificate hierarchies are designed to carry identical information. Consequently, a child CA requesting the revocation of a certificate during these two phases MUST perform that request for both algorithm suites (A and B). A non-leaf CA is NOT required to verify that its child CAs comply with this requirement.

8. Key rollover

Key rollover (without algorithm changes) is effected independently for each algorithm suite and MUST follow the process described in [I-D.ietf-sidr-keyroll].

9. Repository structure

The two parallel hierarchies that will exist during the transition process SHOULD have independent publications points. The repository structures for each algorithm suite are described in [I-D.ietf-sidr-repos-struct].

10. IANA Considerations

No IANA requirements

11. Security Considerations

An algorithm transition in RPKI should be a very infrequent event and it requires wide community consensus. The events that may lead to an algorithm transition may be related to a weakness of the cryptographic strength of the algorithm suite in use by RPKI, which is normal to happen over time. The procedure described in this document will take months or years to complete an algorithm transition. During that time, the RPKI system will be vulnerable to any cryptographic weakness that may have triggered this procedure.

This document does not describe an emergency mechanism for algorithm migration. Due to the distributed nature of RPKI, and the very large number of CAs and RPs, the authors do not believe it is feasible to effect an emergency algorithm migration procedure.

If a CA does not complete its migration to the new algorithm suite as described in this document (after the EOL of the "old" algorithm suite), its signed product set will not longer be valid. Consequently, the RPKI may, at the end of Phase 4, have a smaller number of valid signed products than before starting the process. Conversely, a RP that does not follow this process will lose the ability to validate signed products issued under the new algorithm suite. The resulting incomplete view of routing info from the RPKI (as a result of a failure by CAs or RPs to complete the transition) could degrade routing in the public Internet.

12. Acknowledgements

The authors would like to acknowledge the work of the SIDR working group co-chairs (Sandra Murphy and Chris Morrow) as well as the contributions given by Geoff Huston, Arturo Servin and Brian Weis.

13. References

13.1. Normative References

[I-D.ietf-sidr-cp]
Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource PKI (RPKI", draft-ietf-sidr-cp-17 (work in progress), April 2011.

[I-D.ietf-sidr-keyroll]
Huston, G., Michaelson, G., and S. Kent, "CA Key Rollover in the RPKI", draft-ietf-sidr-keyroll-05 (work in progress), December 2010.

[I-D.ietf-sidr-repos-struct]
Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", draft-ietf-sidr-repos-struct-06 (work in progress), November 2010.

[I-D.ietf-sidr-res-certs]
Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", draft-ietf-sidr-res-certs-21 (work in progress), December 2010.

[I-D.ietf-sidr-rescerts-provisioning]
Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", draft-ietf-sidr-rescerts-provisioning-10 (work in progress), June 2011.

[I-D.ietf-sidr-rpki-algs]
Huston, G., "A Profile for Algorithms and Key Sizes for use in the Resource Public Key Infrastructure", draft-ietf-sidr-rpki-algs-04 (work in progress), November 2010.

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

[RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

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

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[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.

13.2. Informative References

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, February 2010.

Appendix A. Change Log

From 02 to 03:

1. Explicitely named than "mixed" certificates are not allowed for CA certs but may be possible for EE certs that are not used to validate repository objects.

From 01 to 02:

1. Add reference to Multi-Objects validation

2. EOL Data is the only milestone that RP and CA take actions "at the same time".

3. Updated references

4. Editorial

From 00 to 01:

1. Include text to clarify former Suites

2. Include text that documents that an RP that validates an object signed with either suites in Phase 2 MUST consider it as valid

From individual submission to WG item:

1. Change form "laisez faire" to "top-down"

2. Included Multi Algorithm support in the RPKI provisioning protocol

3. Included Validation of multiple instance of signed products

4. Included Revocations

5. Included Key rollover

6. Included Repository structure

7. Included Security Considerations

8. Included Acknowledgements

Authors' Addresses

Roque Gagliano
Cisco Systems
Avenue des Uttins 5
Rolle, 1180
Switzerland

Email: rogaglia@cisco.com

Stephen Kent
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
USA

Email: kent@bbn.com

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

Email: turners@ieca.com

Secure Inter-Domain Routing Working Group
Internet-Draft
Updates: [ID.sidr-rpki-algs]
Intended Status: Standards Track
Expires: April 26, 2012

S. Turner
IECA

October 24, 2011

BGP Algorithms, Key Formats, & Signature Formats
draft-ietf-sidr-bgpsec-algs-00

Abstract

This document specifies the algorithms, algorithms' parameters, asymmetric key formats, asymmetric key size and signature format used in BGPSEC (Border Gateway Protocol Security). This document updates the Profile for Algorithms and Key Sizes for use in the Resource Public Key Infrastructure (draft-ietf-sidr-rpki-algs).

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 April 26, 2012.

Copyright Notice

1. Introduction

This document specifies:

- o the digital signature algorithm and parameters;
- o the hash algorithm and parameters;
- o the public and private key formats; and,
- o the signature format

used by Resource Public Key Infrastructure (RPKI) Certification Authorities (CA), and BGPSEC (Border Gateway Protocol Security) speakers (i.e., routers). CAs use these algorithms when issuing BGPSEC Router Certificates [ID.bgpsec-pki-profiles] and CRLs [ID.sidr-res-cert-profile]. BGPSEC routers use these when requesting BGPSEC certificates [ID.bgpsec-pki-profiles], generating BGPSEC Update messages [ID.sidr-bgpsec-protocol], and verifying BGPSEC Update messages [ID.sidr-bgpsec-protocol].

This document is referenced by the BGPSEC specification [ID.bgpsec-protocol] and the profile for BGPSEC Router Certificates and Certification Requests [ID.bgpsec-pki-profiles]. Familiarity with these documents is assumed. Implementers are reminded, however, that, as noted in Section 2 of [ID.bgpsec-pki-profiles], the algorithms used to sign CA Certificates, BGPSEC Router Certificates, and CRLs are found in [ID.sidr-rpki-algs].

This document updates [ID.sidr-rpki-algs] to add support for a) a different algorithm for BGPSEC certificate requests, which are only issued by BGPSEC speakers; b) a different Subject Public Key Info format for BGPSEC certificates, which is needed for the specified BGPSEC signature algorithm; and, c) a different signature format for BGPSEC signatures, which is needed for the specified BGPSEC signature algorithm. The BGPSEC certificate are differentiated from other RPKI certificates by the use of the BGPSEC Extended Key Usage defined in [ID.bgpsec-pki-profiles].

1.1. Terminology

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 [RFC2119].

2. Algorithms

Four cryptographic algorithms are used to support BGPSEC:

- o The signature algorithm used when issuing BGPSEC certificates and CRLs, which would revoke BGPSEC certificates, MUST be as specified in [ID.sidr-rpki-algs].

- The signature algorithm used in certification requests and BGPSEC Update messages MUST be Elliptic Curve Digital Signature Algorithm (ECDSA) [DSS].

- The hashing algorithm used when issuing certificates and CRLs MUST be as specified in [ID.sidr-rpki-algs].

- The hashing algorithm use when generating certification requests and BGPSEC Update messages MUST be SHA-256 [SHS]. Hash algorithms are not identified by themselves in certificates, or BGPSEC Update messages instead they are combined with the digital signature algorithm (see below).

  NOTE: The exception to the above hashing algorithm is the use of SHA-1 [SHS] when CAs generate authority and subject key identifiers [ID.bgpsec-pki-profiles].

To support BGPSEC, the algorithms are identified as follows:

- In certificates and CRLs, an Object Identifier (OID) is used. The value and locations are as specified in section 2 of [ID.sidr-rpki-algs].

- In certification request, an OID is used. The ecdsa-with-SHA256 OID [RFC5480] MUST appear in the PKCS #10 signatureAlgorithm field [RFC4211] or in Certificate Request Message Format (CRMF) POPOSigningKey signature field [RFC2986].

- In BGPSEC Update messages, the ECDSA with SHA-256 Algorithm Suite Identifier from Section 7 is included in the Signature-Block List's Algorithm Suite Identifier field.

3. Asymmetric Key Format

The RSA key pairs used to compute signatures on CA certificates, BGPSEC Router Certificates, and CRLs are as specified in section 3 of [ID.sidr-rpki-algs]. The remainder of this section addresses key formats found in the BGPSEC router certificate requests and in BGPSEC Router Certificates.

The ECDSA key pairs used to compute signatures for certificate requests and BGPSEC Update messages MUST come from the P-256 curve [RFC5480]. The public key pair MUST use the uncompressed form.

3.1. Public Key Format

The Subject's public key is included in subjectPublicKeyInfo [RFC5280]. It has two sub-fields: algorithm and subjectPublicKey.

The values for the structures and their sub-structures follow:

- algorithm (which is an AlgorithmIdentifier type): The id-ecPublicKey OID MUST be used in the algorithm field, as specified in 2.1.1 of [RFC5480]. The value for the associated parameters MUST be secp256r1, as specified in 2.1.1.1 of [RFC5480].

- subjectPublicKey: ECPublicKey MUST be used to encode the certificate's subjectPublicKey field, as specified in Section 2.2 of [RFC5480].

3.2. Private Key Format

Local Policy determines private key format.

4. Signature Format

The structure for the certificate's and CRL's signature field MUST be as specified in Section 4 of [ID.sidr-rpki-algs]. The structure for the certification request's and BGPSEC Update message's signature field MUST be as specified in Section 2.2.3 of [RFC3279].

5. Additional Requirements

It is anticipated that BGPSEC will require the adoption of updated key sizes and a different set of signature and hash algorithms over time, in order to maintain an acceptable level of cryptographic security to protect the integrity of BGPSEC. This profile should be updated to specify such future requirements, when appropriate.

CAs and RPs SHOULD be capable of supporting a transition to allow for the phased introduction of additional encryption algorithms and key specifications, and also accommodate the orderly deprecation of previously specified algorithms and keys. Accordingly, CAs and RPs SHOULD be capable of supporting multiple RPKI algorithm and key profiles simultaneously within the scope of such anticipated transitions. The recommended procedures to implement such a transition of key sizes and algorithms is not specified in this document.

6. Security Considerations

The Security Considerations of [RFC3279], [RFC5480], [ID.sidr-rpki-algs], and [ID.bgpsec-pki-profiles] apply to certificates. The security considerations of [RFC3279], [ID.sidr-rpki-algs], [ID.bgpsec-pki-profiles] apply to certification requests. The security considerations of [RFC3279] and [ID.sidr-bgpsec-protocol] apply to BGPSEC Update messages. No new security are introduced as a

result of this specification.

7. IANA Considerations

The Internet Assigned Numbers Authority (IANA) is requested to define the "BGPSEC Algorithm Suite Registry" described below.

An algorithm suite consists of a digest algorithm and a signature algorithm. This specification creates an IANA registry of one-octet BGPSEC algorithm suite identifiers. Additionally, this document registers a single algorithm suite which uses the digest algorithm SHA-256 and the signature algorithm ECDSA on the P-256 curve [RFC5480].

BGPSEC Algorithm Suites Registry

Digest Algorithm	Signature Algorithm	Algorithm Suite Identifier	Specification Pointer
SHA-256	ECDSA P-256	TBD	RFC 5480

Future assignments are to be made using either the Standards Action process defined in [RFC5226], or the Early IANA Allocation process defined in [RFC4020]. Assignments consist of a digest algorithm name, signature algorithm name, and the algorithm suite identifier value.

10. Acknowledgements

The author wishes to thank Geoff Huston for producing [ID.sidr-rpki-algs], which this document is heavily based on. I'd also like to thank Roque Gagliano for his review and comments.

11. References

11.1. Normative References

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

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

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key

Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[RFC4020] Kompella, K. and A. Zinin, "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 4020, February 2005.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[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.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, March 2009.

[DSS] National Institute of Standards and Technology (NIST), FIPS Publication 186-3: Digital Signature Standard, June 2009.

[SHS] National Institute of Standards and Technology (NIST), "FIPS Publication 180-3: Secure Hash Standard", FIPS Publication 180-3, October 2008.

[ID.sidr-res-cert-profile] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", draft-ietf-sidr-res-certs, work-in-progress.

[ID.sidr-rpki-algs] Huston, G., "A Profile for Algorithms and Key Sizes for use in the Resource Public Key Infrastructure", draft-ietf-sidr-rpki-algs, work-in-progress.

[ID.sidr-bgpsec-protocol] Lepinski, M., "BGPSEC Protocol Specification", draft-ietf-sidr-bgpsec-protocol, work-in-progress.

[ID.bgpsec-pki-profiles] Reynolds, M. and S. Turner, "A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests", draft-ietf-sidr-bpgsec-pki-profiles, work-in-progress.

11.1. Informative References

None.

Authors' Addresses

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

EMail: turners@ieca.com

Network Working Group
Internet-Draft
Intended status: BCP
Expires: April 21, 2012

R. Bush
Internet Initiative Japan

October 19, 2011

BGPsec Operational Considerations
draft-ietf-sidr-bgpsec-ops-01

Abstract

Deployment of the BGPsec architecture and protocols has many operational considerations. This document attempts to collect and present them. It is expected to evolve as BGPsec is formalized and initially deployed.

Requirements Language

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 [RFC2119].

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 April 21, 2012.

Copyright Notice

Table of Contents

1. Introduction

BGPsec is a new protocol with many operational considerations. It is expected to be deployed incrementally over a number of years. As core BGPsec-capable routers may require large memory and crypto assist, it is thought that origin validation based on the RPKI will occur over the next two to five years and that BGPsec will start to deploy late in that window.

BGPsec relies on widespread propagation of the Resource Public Key Infrastructure (RPKI) [I-D.ietf-sidr-arch]. How the RPKI is distributed and maintained globally and within an operator's infrastructure may be different for BGPsec than for origin validation.

BGPsec need be spoken only by a AS's eBGP speaking, AKA border, routers, and is designed so that it can be used to protect announcements which are originated by small edge routers, and this has special operational considerations.

Different prefixes have different timing and replay protection considerations.

2. Suggested Reading

It is assumed that the reader understands BGP, [RFC4271], BGPsec, [I-D.lepinski-bgpsec-overview], the RPKI, see [I-D.ietf-sidr-arch], the RPKI Repository Structure, see [I-D.ietf-sidr-repos-struct], and ROAs, see [I-D.ietf-sidr-roa-format].

3. RPKI Distribution and Maintenance

The RPKI is a distributed database containing certificates, CRLs, manifests, ROAs, and Ghostbuster Records as described in [I-D.ietf-sidr-repos-struct]. Policies and considerations for RPKI object generation and maintenance are discussed elsewhere.

A local valid cache containing all RPKI data may be gathered from the global distributed database using the rsync protocol and a validation tool such as rcynic.

Validated caches may also be created and maintained from other validated caches. Network operators SHOULD take maximum advantage of this feature to minimize load on the global distributed RPKI database.

As RPKI-based origin validation relies on the availability of RPKI data, operators SHOULD locate caches close to routers that require these data and services. A router can peer with one or more nearby caches.

For redundancy, a router SHOULD peer with more than one cache at the same time. Peering with two or more, at least one local and others remote, is recommended.

If an operator trusts upstreams to carry their traffic, they SHOULD also trust the RPKI data those upstreams cache, and SHOULD peer with those caches. Note that this places an obligation on those upstreams to maintain fresh and reliable caches.

A transit provider or a network with peers SHOULD validate NLRI in announcements made by upstreams, downstreams, and peers. To minimize impact on the global RPKI, they SHOULD fetch from and then revalidate data from caches provided by their upstreams.

An environment where private address space is announced in eBGP the operator MAY have private RPKI objects which cover these private spaces. This will require a trust anchor created and owned by that environment, see [I-D.ietf-sidr-ltamgmt].

4. AS/Router Certificates

A site/operator MAY use a single certificate/key in all their routers, one certificate/key per router, or any granularity in between.

A large operator, concerned that a compromise of one router’s key would make many routers vulnerable, MAY accept a more complex certificate/key distribution burden to reduce this exposure.

On the other extreme, an edge site with one or two routers MAY use a single certificate/key.

Routers MAY be capable of generating their own keys and having their certificates signed and published in the RPKI by their NOC. This would mean that a router’s private key need never leave the router.

5. Within a Network

BGPsec is spoken by edge routers in a network, those which border other networks/ASs.

In a fully BGPsec enabled AS, Route Reflectors MUST have BGPsec enabled if and only if there are eBGP speakers in their client cone.

A BGPsec capable router MAY use the data it receives to influence local policy within its network, see Section 8. In deployment this policy should fit into the AS's existing policy, preferences, etc. This allows a network to incrementally deploy BGPsec capable border routers.

eBGP speakers which face more critical peers or up/downstreams would be candidates for the earliest deployment. Both securing one's own announcements and validating received announcements should be considered in partial deployment.

An eBGP listener MUST NOT trust non-BGPsec markings such as communities received across a trust boundary.

6. Considerations for Edge Sites

An edge site which does not provide transit and trusts its upstream(s) SHOULD only originate a signed prefix announcement and need not validate received announcements.

BGPsec protocol capability negotiation provides for a speaker signing the data it sends but being unable to accept signed data. Thus a smallish edge router may hold only its own signing key(s) and sign it's announcement but not receive signed announcements and therefore not need to deal with the majority of the RPKI.

As the vast majority (84%) of ASs are stubs, and they announce the majority of prefixes, this allows for simpler and cheaper early incremental deployment. It may also mean that edge sites concerned with routing security will be attracted to upstreams which support BGPsec.

7. Beaconing Considerations

The BGPsec protocol attempts to reduce exposure to replay attacks by allowing the route originator to sign an announcement with a validity period and re-announce well within that period.

This re-announcement is termed 'beaconing'. All timing values are, of course, jittered.

It is only the originator of an NLRI which signs the announcement with a validity period.

To reduce vulnerability to a lost beacon announcement, a router SHOULD beacon at a rate somewhat greater than half the signature validity period it uses.

As beaconing places a load on the entire global routing system, careful thought MUST be given to any need to beacon frequently. This would be based on a conservative estimation of the vulnerability to a replay attack.

Beacon timing and signature validity periods SHOULD be as follows:

The Exemplary Citizen: Prefix originators who are not overly concerned about replay attacks might announce with a signature validity of multiple weeks and beacon one third of the validity period.

Normal Prefix: Most prefixes SHOULD announce with a signature validity of a week and beacon every three days.

Critical Prefix: Of course, we all think what we do is critical. But prefixes of top level DNS servers, and RPKI publication points are actually critical to large swaths of the Internet and are therefore tempting targets for replay attacks. It is suggested that the beaconing of these prefixes SHOULD be two to four hours, with a signature validity of six to twelve hours.

Note that this may incur route flap damping (RFD) with current default but deprecated RFD parameters, see [I-D.ymbk-rfd-usable].

8. Routing Policy

Unlike origin validation based on the RPKI, BGPsec marks a received announcement as Valid or Invalid, there is no NotFound state. How this is used in routing is up to the operator’s local policy. See [I-D.ietf-sidr-pfx-validate].

As BGPsec will be rolled out over years and does not allow for intermediate non-signing edge routers, coverage will be spotty for a long time. Hence a normal operator’s policy SHOULD NOT be overly strict, perhaps preferring valid announcements and giving very low preference, but still using, invalid announcements.

A BGPsec speaker validates signed paths at the eBGP edge.

Local policy on the eBGP edge MAY convey the validation state of a BGP signed path through normal local policy mechanisms, e.g. setting a BGP community, or modifying a metric value such as local-preference

or MED. Some MAY choose to use the large Local-Pref hammer. Others MAY choose to let AS-Path rule and set their internal metric, which comes after AS-Path in the BGP decision process.

Because of possible RPKI version skew, an AS Path which does not validate at router R0 might validate at R1. Therefore, signed paths that are invalid and yet propagated SHOULD have their signatures kept intact and should be signed if sent to external BGPsec speakers.

This implies that updates which a speaker judges to be invalid MAY be propagated to iBGP peers. Therefore, unless local policy ensures otherwise, a signed path learned via iBGP MAY be invalid. If needed, the validation state SHOULD be signaled by normal local policy mechanisms such as communities or metrics.

On the other hand, local policy on the eBGP edge might preclude iBGP or eBGP announcement of signed AS Paths which are invalid.

If a BGPsec speaker receives an unsigned path, it SHOULD perform origin validation per [I-D.ietf-sidr-pfx-validate].

If it is known that a BGPsec neighbor is not a transparent route server, and the router provides a knob to disallow a received pCount (prepend count, zero for transparent route servers) of zero, that knob SHOULD be applied.

9. Notes

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no 'fix' for this, it is the nature of distributed data with distributed caches.

Operators who manage certificates SHOULD have RPKI Ghostbuster Records (see [I-D.ietf-sidr-ghostbusters]), signed indirectly by End Entity certificates, for those certificates on which others' routing depends for certificate and/or ROA validation.

As a router must evaluate certificates and ROAs which are time dependent, routers' clocks MUST be correct to a tolerance of approximately an hour.

If a router has reason to believe its clock is seriouly incorrect, e.g. it has a time earlier than 2011, it SHOULD NOT attempt to validate incoming updates. It SHOULD defer validation until it believes it is within reasonable time tolerance.

Servers should provide time service, such as NTP [RFC5905], to client routers.

10. Security Considerations

BGPsec is all about security, routing security. The major security considerations for the protocol are described in [I-D.ietf-sidr-bgpsec-protocol].

11. IANA Considerations

This document has no IANA Considerations.

12. Acknowledgments

The author wishes to thank the BGPsec design team.

13. References

13.1. Normative References

[I-D.ietf-sidr-bgpsec-protocol]
Lepinski, M., "BGPSEC Protocol Specification", draft-ietf-sidr-bgpsec-protocol-00 (work in progress), June 2011.

[I-D.ietf-sidr-ghostbusters]
Bush, R., "The RPKI Ghostbusters Record", draft-ietf-sidr-ghostbusters-15 (work in progress), October 2011.

[I-D.ietf-sidr-roa-format]
Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", draft-ietf-sidr-roa-format-12 (work in progress), May 2011.

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

13.2. Informative References

[I-D.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support

Secure Internet Routing", draft-ietf-sidr-arch-13 (work in progress), May 2011.

[I-D.ietf-sidr-ltamgmt]
Reynolds, M. and S. Kent, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-02 (work in progress), June 2011.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-02 (work in progress), July 2011.

[I-D.ietf-sidr-repos-struct]
Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", draft-ietf-sidr-repos-struct-09 (work in progress), July 2011.

[I-D.lepinski-bgpsec-overview]
Lepinski, M. and S. Turner, "An Overview of BGPSEC", draft-lepinski-bgpsec-overview-00 (work in progress), March 2011.

[I-D.ymbk-rfd-usable]
Pelsser, C., Bush, R., Patel, K., Mohapatra, P., and O. Maennel, "Making Route Flap Damping Usable", draft-ymbk-rfd-usable-01 (work in progress), June 2011.

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

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

Author's Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Secure Inter-Domain Routing Working Group
Internet-Draft
Updates: [ID.sidr-res-cert-profile]
Intended Status: Standards Track
Expires: April 25, 2012

M. Reynolds
BBN
S. Turner
IECA

October 24, 2011

A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests
draft-ietf-sidr-bgpsec-pki-profiles-00

Abstract

This document defines a standard profile for X.509 certificates for the purposes of supporting validation of Autonomous System (AS) paths in the Border Gateway Protocol (BGP), as part of an extension to that protocol known as BGPSEC. BGP is a critical component for the proper operation of the Internet as a whole. The BGPSEC protocol is under development as a component to address the requirement to provide security for the BGP protocol. The goal of BGPSEC is to design a protocol for full AS path validation based on the use of strong cryptographic primitives. The end-entity (EE) certificates specified by this profile are issued under Resource Public Key Infrastructure (RPKI) Certification Authority (CA) certificates, containing the AS Identifier Delegation extension, to routers within the Autonomous System (AS). The certificate asserts that the router(s) holding the private key are authorized to send out secure route advertisements on behalf of the specified AS. This document also profiles the Certificate Revocation List (CRL), profiles the format of certification requests, and specifies Relying Party certificate path validation procedures. The document extends the RPKI; therefore, this documents updates the RPKI Resource Certificates Profile (draft-ietf-sidr-res-cert-profile).

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 April 26, 2012.

1. Introduction

This document defines a profile for X.509 end-entity (EE) certificates [RFC5280] for use in the context of certification of Autonomous System (AS) paths in the Border Gateway Protocol Security (BGPSEC) protocol. Such certificates are termed "BGPSEC Router Certificates". The holder of the private key associated with a BGPSEC Router Certificate is authorized to send secure route advertisements (BGPSEC UPDATEs) on behalf of the AS named in the certificate. That is, a router holding the private key may send to its BGP peers, route advertisements that contain the specified AS number as the last item in the AS PATH attribute. A key property that BGPSEC will provide is that every AS along the AS PATH can verify that the other ASes along the path have authorized the advertisement of the given route (to the next AS along the AS PATH).

This document is a profile of [ID.sidr-res-cert-profile], which is a profile of [RFC5280], and it updates [ID.sidr-res-cert-profile]. It establishes requirements imposed on a Resource Certificate that is used as a BGPSEC Router Certificate, i.e., it defines constraints for certificate fields and extensions for the certificate to be valid in this context. This document also profiles the Certificate Revocation List (CRL) and certification requests. Finally, this document specifies the Relying Party (RP) certificate path validation procedures.

1.1. Terminology

It is assumed that the reader is familiar with the terms and concepts described in "A Profile for X.509 PKIX Resource Certificates" [ID.sidr-res-cert-profile], "BGPSEC Protocol Specification" [ID.sidr-

bgpsec-protocol], "A Border Gateway Protocol 4 (BGP-4)" [RFC4271], "BGP Security Vulnerabilities Analysis" [RFC4272], "Considerations in Validating the Path in BGP" [RFC5123], and "Capability Advertisement with BGP-4" [RFC5492].

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 [RFC2119].

2. Describing Resources in Certificates

Figure 1 depicts some of the entities in the RPKI and some of the products generated by RPKI entities. IANA issues a Certification Authority (CA) to a Regional Internet Registries (RIR). The RIR, in turn, issues a CA certificate to an Internet Service Providers (ISP). The ISP in turn issues End-Entity (EE) Certificates to itself as well as CRLs. These certificates are referred to as "Resource Certificates", and are profiled in [ID.sidr-res-cert-profile]. The [ID.sidr-arch] envisioned using Resource Certificates to generate Manifests [ID.sidr-rpki-manifests] and Route Origin Authorizations (ROAs) [ID.sidr-rpki-roa-format]. ROAs and Manifests also include the Resource Certificates used to sign them.

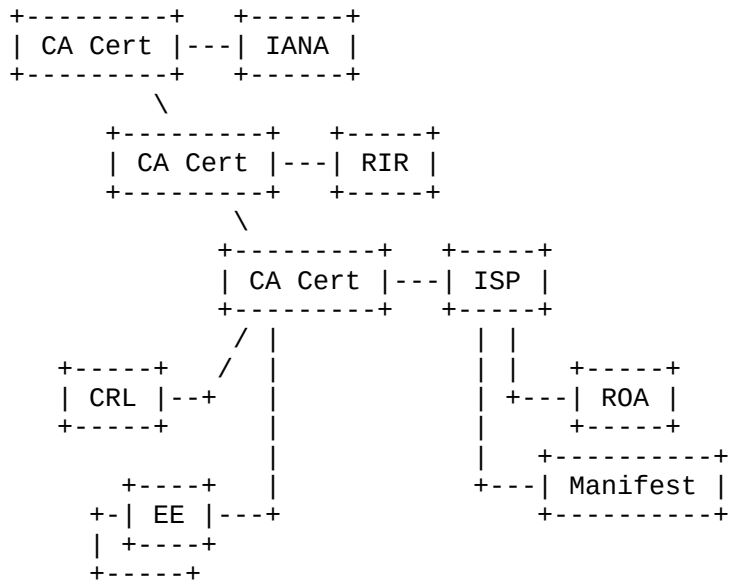


Figure 1

This document defines another type of Resource Certificate, which is

referred to as a "BGPSEC Router Certificate". The purpose of this certificate is explained in Section 1 and falls within the scope of appropriate uses defined within [ID.sidr-cp]. The issuance of BGPSEC Router Certificates has minimal impact on RPKI CAs because the RPKI CA certificate and CRL profile remain unchanged (i.e., they are as specified in [ID.sidr-res-cert-profile]). Further, the algorithms used to generate RPKI CA certificates that issue the BGPSEC Router Certificates and the CRLs necessary to check the validity of the BGPSEC Router Certificates remain unchanged (i.e., they are as specified in [ID.sidr-rpki-algs]). The only impact is that the RPKI CAs will need to be able to process a profiled certificate request (see Section 5) signed with algorithms found in [ID.turner-sidr-bgpsec-algs]. The use of BGPSEC Router Certificates in no way affects RPKI RPs that process Manifests and ROAs because the public key found in the BGPSEC Router Certificate is only ever used to verify the signature on the BGPSEC certificate request (only CAs process these), another BGPSEC Router Certificate (only BGPSEC routers process these), and the signature on a BGPSEC Update Message [ID.sidr-bgpsec-protocol] (only BGPSEC routers process these).

Only the differences between this profile and the profile in [ID.sidr-res-cert-profile] are listed. Note that BGPSEC Router Certificates are EE certificates and as such there is no impact on process described in [ID.sidr-algorithm-agility].

3. Updates to [ID.sidr-res-cert-profile]

3.1 BGPSEC Router Certificate Fields

A BGPSEC Router Certificate is a valid X.509 public key certificate, consistent with the PKIX profile [RFC5280], containing the fields listed in this section. This profile is also based on [ID.sidr-res-cert-profile] and only the differences between this profile and the profile in [ID.sidr-res-cert-profile] are listed.

3.1.1.1 Subject

This field identifies the router to which the certificate has been issued. Consistent with [ID.sidr-res-cert-profile], only two attributes are allowed in the Subject field: common name and serial number. Moreover, the only common name encoding options that are supported are printableString and UTF8String. For BGPSEC Router Certificates, it is RECOMMENDED that the common name attribute contain the literal string "ROUTER-" followed by the 32-bit AS Number [RFC3779] encoded as eight hexadecimal digits and that the serial number attribute contain the 32-bit BGP Identifier [RFC4271] (i.e., the router ID) encoded as eight hexadecimal digits. If the same certificate is issued to more than one router (hence the private key

is shared among these routers), the choice of the router ID used in this name is at the discretion of the Issuer. Note that router IDs are not guaranteed to be unique across the Internet, and thus the Subject name in a BGPSEC Router Certificate issued using this convention also is not guaranteed to be unique across different issuers. However, each certificate issued by an individual CA MUST contain a Subject name that is unique within that context.

3.1.2. Subject Public Key Info

Refer to section 3.1 of [ID.sidr-bgpsec-algs].

3.1.3. BGPSEC Router Certificate Version 3 Extension Fields

The following X.509 V3 extensions MUST be present (or MUST be absent, if so stated) in a conforming BGPSEC Router Certificate, except where explicitly noted otherwise. No other extensions are allowed in a conforming BGPSEC Router Certificate.

3.1.3.1. Extended Key Usage

BGPSEC Router Certificates MUST include the Extended Key Usage (EKU) extension. As specified, in [ID.sidr-res-cert-profile] this extension MUST be marked as non-critical. This document defines one EKU for BGPSEC Router Certificates:

```
id-kp OBJECT IDENTIFIER ::=
   { iso(1) identified-organization(3) dod(6) internet(1)
     security(5) mechanisms(5) pkix(7) TBD }

id-kp-bgpsec-router OBJECT IDENTIFIER ::= { id-kp TBD }
```

Relying Parties MUST require the extended key usage extension to be present in a BGPSEC Router Certificate. If multiple KeyPurposeId values are included, the relying parties need not recognize all of them, as long as the required KeyPurposeId value is present. BGPSEC RPs MUST reject certificates that do not contain the BGPSEC Router EKU even if they include the anyExtendedKeyUsage OID defined in [RFC5280].

3.1.3.2. Subject Information Access

This extension is not used in BGPSEC Router Certificates. It MUST be omitted.

3.1.3.3. IP Resources

This extension is not used in BGPSEC Router Certificates. It MUSt be

omitted.

3.1.3.4. AS Resources

Each BGPSEC Router Certificate MUST include the AS Resource Identifier Delegation extension, as specified in section 4.8.11 of [ID.sidr-res-cert-profile]. The AS Resource Identifier Delegation extension MUST include exactly one AS number, and the "inherit" element MUST NOT be specified.

3.2. BGPSEC Router Certificate Request Profile

Refer to section 6 of [ID.sidr-res-cert-profile]. The only differences between this profile and the profile in [ID.sidr-res-cert-profile] are:

- The ExtendedKeyUsage extension request MUST be included and the CA MUST honor the request;
- The SubjectPublicKeyInfo and PublicKey fields are specified in [ID.sidr-bgpsec-algs]; and,
- The request is signed with the algorithms specified in [ID.sidr-bgpsec-algs].

3.3. BGPSEC Router Certificate Validation

The validation procedure used for BGPSEC Router Certificates is identical to the validation procedure described in Section 7 of [ID.sidr-res-cert-profile] except that where "this specification" refers to [ID.sidr-res-cert-profile] in that profile in this profile "this specification" is this document.

The differences are as follows:

- BGPSEC Router Certificates MUST include the BGPSEC EKU defined in Section 3.9.5.
- BGPSEC Router Certificates MUST NOT include the SIA extension.
- BGPSEC Router Certificates MUST NOT include the IP Resource extension.
- BGPSEC Router Certificates MUST include the AS Resource Identifier Delegation extension.
- BGPSEC Router Certificate MUST include the "Subject Public Key Info" described in [ID.sidr-bgpsec-algs] as it updates [ID.sidr-

rpki-algs].

NOTE: The cryptographic algorithms used by BGPSEC routers are found in [ID.sidr-bgpsec-algs]. Currently, the algorithms specified in [ID.sidr-bgpsec-algs] and [ID.sidr-rpki-algs] are different. BGPSEC RPs will need to support algorithms that are needed to validate BGPSEC signatures as well as the algorithms that are needed to validate signatures on BGPSEC certificates, RPKI CA certificates, and RPKI CRLs.

4. Design Notes

The BGPSEC Router Certificate profile is based on the Resource Certificate profile as specified in [ID.sidr-res-cert-profile]. As a result, many of the design choices herein are a reflection of the design choices that were taken in that prior work. The reader is referred to [ID.sidr-res-cert-profile] for a fuller discussion of those choices.

5. Security Considerations

The Security Considerations of [ID.sidr-res-cert-profile] apply.

A bgpsec certificate will fail RPKI validation, as defined in [ID.sidr-res-cert-profile], because the algorithm suite is different. Consequently, a RP needs to identify the EKU before applying the correspondent validation.

A BGPSEC Router Certificate is an extension of the RPKI [ID.sidr-arch] to encompass routers. It is a building block of the larger BGPSEC security protocol used to validate signatures on BGPSEC Signature-Segment origination of Signed-Path segments [ID.sidr-bgpsec-protocol]. Thus its essential security function is the secure binding of an AS number to a public key, consistent with the RPKI allocation/assignment hierarchy.

6. IANA Considerations

None.

7. Acknowledgements

We would like to thanks Geoff Huston, George Michaelson, and Robert Loomans for their work on [ID.sidr-res-cert-profile], which this work is based on. In addition, the efforts of Steve Kent and Matt Lepinski were instrumental in preparing this work. Additionally, we'd like to thank Roque Gagliano, Sandra Murphy, and Geoff Huston for their reviews and 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.

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

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

[ID.sidr-res-cert-profile] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", draft-ietf-sidr-res-certs, work-in-progress.

[ID.sidr-rpki-algs] Huston, G., "The Profile for Algorithms and Key Sizes for use in the Resource Public Key Infrastructure", draft-ietf-sidr-rpki-algs, work-in-progress.

[ID.sidr-bgpsec-algs] Reynolds, M. and S. Turner, "BGP Algorithms, Key Formats, & Signature Formats", draft-ietf-sidr-bgpsec-algs, work-in-progress.

8.2. Informative References

[RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, January 2006.

[RFC5123] White, R. and B. Akyol, "Considerations in Validating the Path in BGP", RFC 5123, February 2008.

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, February 2009.

[ID.sidr-cp] Kent, S., Kong, D., Seo, K., and R., Watro, "Certificate Policy (CP) for the Resource PKI (RPKI)", draft-ietf-sidr-cp, work-in-progress.

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

[ID.sidr-rpki-roa-format] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", draft-ietf-sidr-roa-format, work-in-progress

[ID.sidr-rpki-manifests] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure", draft-ietf-sidr-rpki-manifests, work-in-progress.

[ID.sidr-algorithm-agility] Gagliano, R., Kent, S., and S. Turner, "Algorithm Agility Procedure for RPKI", draft-ietf-sidr-algorithm-agility, work-in-progress.

[ID.sidr-bgpsec-protocol] Lepinksi, M., "BGPSEC Protocol Specification", draft-ietf-sidr-bgpsec-protocol, work-in-progress.

Appendix A. Example BGPSEC Router Certificate

Appendix B. Example BGPSEC Router Certificate Request

Appendix C. Change Log

Please delete this section prior to publication.

C.1 Changes from turner-bgpsec-pki-profiles-02 to sidr-bgpsec-pki-profiles-00

Added this change log.

Amplified that a BGPSEC RP will need to support both the algorithms in [ID.sidr-bgpsec-algs] for BGPSEC and the algorithms in [ID.sidr-rpki-algs] for certificates and CRLs.

Changed the name of AS Resource extension to AS Resource Identifier Delegation to match what’s in RFC 3779.

C.2 Changes from turner-bgpsec-pki-profiles -01 to -02

Added text in Section 2 to indicate that there’s no impact on the procedures defined in [ID.sidr-algorithm-agility].

Added a security consideration to let implementers know the BGPSEC certificates will not pass RPKI validation [ID.sidr-res-cert-profile] and that keying off the EKU will help tremendously.

C.3 Changes from turner-bgpsec-pki-profiles -00 to -01

Corrected Section 2 to indicate that CA certificates are also RPKI certificates.

Removed sections and text that was already in [ID.sidr-res-cert-profile]. This will make it easier for reviewers to figure out what is different.

Modified Section 6 to use 2119-language.

Removed requirement from Section 6 to check that the AS # in the certificate is the last number in the AS path information of each BGP UPDATE message. Moved to [ID.sidr-bgpsec-protocol].

Authors' Addresses

Mark Reynolds
Raytheon BBN Technologies Corp.
10 Moulton St.
Cambridge, MA 02138

Email: mreynold@bbn.com

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031
USA

EMail: turners@ieca.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

M. Lepinski, Ed.
BBN

October 31, 2011

BGPSEC Protocol Specification
draft-ietf-sidr-bgpsec-protocol-01

Abstract

This document describes BGPSEC, an extension to the Border Gateway Protocol (BGP) that provides security for the AS-PATH attribute in BGP update messages. BGPSEC is implemented via a new optional non-transitive BGP path attribute that carries a digital signature produced by each autonomous system on the AS-PATH.

Requirements Language

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 [4].

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 May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document describes BGPSEC, a mechanism for providing path security for Border Gateway Protocol (BGP) [1] route advertisements. That is, a BGP speaker who receives a valid BGPSEC update has cryptographic assurance that the advertised route has the following two properties:

1. The route was originated by an AS that has been explicitly authorized by the holder of the IP address prefix to originate route advertisements for that prefix.

2. Every AS listed in the AS_Path attribute of the update explicitly authorized the advertisement of the route to the subsequent AS in the AS_Path.

This document specifies a new optional (non-transitive) BGP path attribute, BGPSEC_Path_Signatures. It also describes how a BGPSEC-compliant BGP speaker (referred to hereafter as a BGPSEC speaker) can generate, propagate, and validate BGP update messages containing this attribute to obtain the above assurances.

BGPSEC relies on the Resource Public Key Infrastructure (RPKI) certificates that attest to the allocation of AS number and IP address resources. (For more information on the RPKI, see [7] and the documents referenced therein.) Any BGPSEC speaker who wishes to send BGP update messages to external peers (eBGP) containing the BGPSEC_Path_Signatures must have an RPKI end-entity certificate (as well as the associated private signing key) corresponding to the BGPSEC speaker’s AS number. Note, however, that a BGPSEC speaker does not require such a certificate in order to validate update messages containing the BGPSEC_Path_Signatures attribute.

2. BGPSEC Negotiation

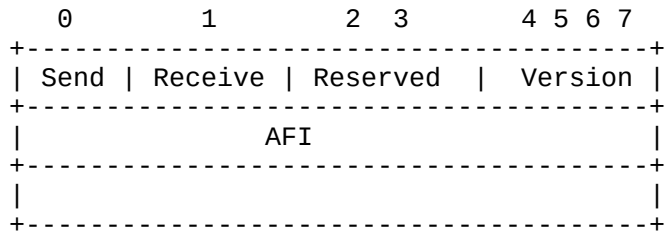
This document defines a new BGP capability [3]that allows a BGP speaker to advertise to its neighbors the ability to send and/or receive BGPSEC update messages (i.e., update messages containing the BGPSEC_Path_Signatures attribute).

This capability has capability code : TBD

The capability length for this capability MUST be set to 3.

The three octets of the capability value are specified as follows.

Capability Value:



The high order bit (bit 0) of the first octet is set to 1 to indicate that the sender is able to send BGPSEC update messages, and is set to zero otherwise. The next highest order bit (bit 1) of this octet is set to 1 to indicate that the sender is able to receive BGPSEC update messages, and is set to zero otherwise. The next two bits of the capability value (bits 2 and 3) are reserved for future use.

The four low order bits (4, 5, 6 and 7) of the first octet indicate the version of BGPSEC for which the BGP speaker is advertising support. This document defines only BGPSEC version 0 (all four bits set to zero). Other versions of BGPSEC may be defined in future documents. A BGPSEC speaker MAY advertise support for multiple versions of BGPSEC by including multiple versions of the BGPSEC capability in its BGP OPEN message.

If there does not exist at least one version of BGPSEC that is supported by both peers in a BGP session, then the use of BGPSEC has not been negotiated. (That is, in such a case, messages containing the BGPSEC_Path_Signatures MUST NOT be sent.)

If version 0 is the only version of BGPSEC for which both peers (in a BGP session) advertise support, then the use of BGPSEC has been negotiated and the BGPSEC peers MUST adhere to the specification of BGPSEC provided in this document. (If there are multiple versions of BGPSEC which are supported by both peers, then the behavior of those peers is outside the scope of this document.)

The second two octets contain the 16-bit Address Family Identifier (AFI) which indicates the address family for which the BGPSEC speaker is advertising support for BGPSEC. This document only specifies BGPSEC for use with two address families, IPv4 and IPv6. BGPSEC for use with other address families may be specified in future documents. Note that if the BGPSEC speaker wishes to use BGPSEC with two different address families (i.e., IPv4 and IPv6) over the same BGP session, then the speaker must include two instances of this capability (one for each address family) in the BGP OPEN message. Also note that a BGPSEC speaker SHOULD NOT advertise the capability

of BGPSEC support for IPv6 unless it has also advertised support for IPv6 [2].

By indicating support for receiving BGPSEC update messages, a BGP speaker is, in particular, indicating that the following are true:

- The BGP speaker understands the BGPSEC_Path_Signatures attribute (see Section 3).
- The BGP speaker supports 4-byte AS numbers (see RFC 4893).

Note that BGPSEC update messages can be quite large, therefore any BGPSEC speaker announcing the capability to receive BGPSEC messages SHOULD also announce support for the capability to receive BGP extended messages [5].

A BGP speaker MUST NOT send an update message containing the BGPSEC_Path_Signatures attribute within a given BGP session unless both of the following are true:

- The BGP speaker indicated support for sending BGPSEC update messages in its open message.
- The peer of the BGP speaker indicated support for receiving BGPSEC update messages in its open message.

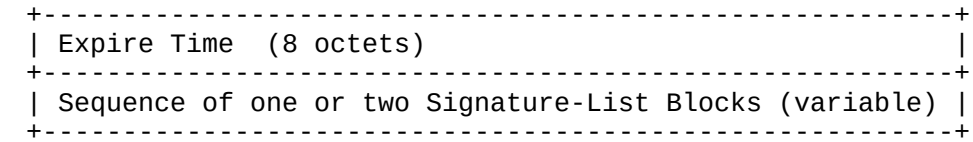
3. The BGPSEC_Path_Signatures Attribute

The BGPSEC_Path_Signatures attribute is a new optional (non-transitive) BGP path attribute.

This document registers a new attribute type code for this attribute : TBD

The BGPSEC_Path_Signatures attribute has the following structure:

BGPSEC_Path_Signatures Attribute

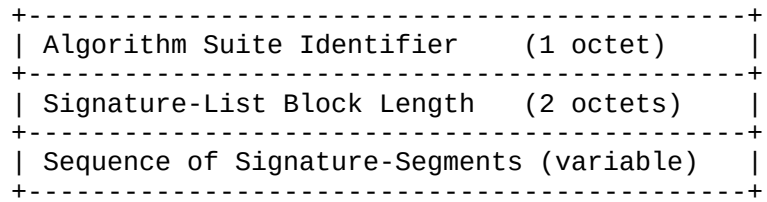


Expire Time contains a binary representation of a time as an unsigned integer number of (non-leap) seconds that have elapsed since midnight

UTC January 1, 1970. The Expire Time indicates the latest point in time that the route advertised in the update message can possibly be considered valid (see Section 5 for details on validity of BGPSEC update messages).

The BGPSEC_Path_Signatures attribute will contain one or two Signature-List Blocks, each of which corresponds to a different algorithm suite. Each of the Signature-List Blocks will contain a signature segment for each AS in the AS Path attribute. In the most common case, the BGPSEC_Path_Signatures attribute will contain only a single Signature-List Block. However, in order to enable a transition from an old algorithm suite to a new algorithm suite, it will be necessary to include two Signature-List Blocks (one for the old algorithm suite and one for the new algorithm suite) during the transition period.

Signature-List Block

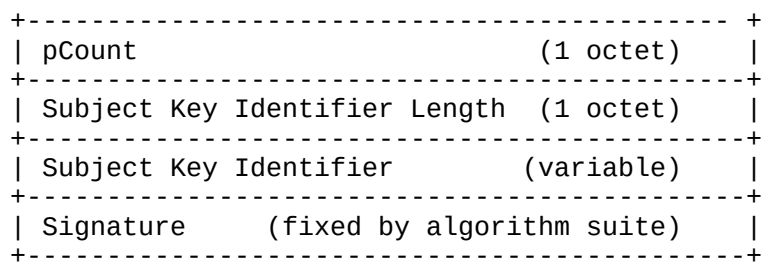


An algorithm suite consists of a digest algorithm and a signature algorithm. This version of BGPSEC only supports signature algorithms that produce a signatures of fixed length. Future registrations of algorithm suites for BGPSEC must specify the length of signatures produced by the algorithm suite. This specification creates an IANA registry of one-octet BGPSEC algorithm suite identifiers (see Section 8).

The Signature-List Block Length is the total number of octets in all Signature-Segments (i.e., the total size of the variable-length portion of the Signature-List block.)

A Signature-Segment has the following structure:

Signature Segments



The pCount field contains an unsigned integer indicating the number of repetitions of the associated autonomous system number that the signature covers. This field enables a BGPSEC speaker to mimic the semantics of adding multiple copies of their AS to the AS-PATH without requiring the speaker to generate multiple signatures.

The Subject Key Identifier Length contains the size (in octets) of the value in the Subject Key Identifier field of the Signature-Segment. The Subject Key Identifier contains the value in the Subject Key Identifier extension of the RPKI end-entity certificate that is used to verify the signature (see Section 5 for details on validity of BGPSEC update messages).

The Signature contains a digital signature that protects the NLRI, the AS_Path and the BGPSEC_Path_Signatures attribute (see Sections 4 and 5 for details on generating and verifying this signature, respectively). The length of the Signature field is a function of the algorithm suite for a given Signature-List Block. The specification for each BGPSEC algorithm suite must provide the length of signatures constructed using the given algorithm suite.

4. Generating a BGPSEC Update

Sections 4.1 and 4.2 cover two cases in which a BGPSEC speaker may generate an update message containing the BGPSEC_Path_Signatures attribute. The first case is that in which the BGPSEC speaker originates a new route advertisement (Section 4.1). That is, the BGPSEC speaker is constructing an update message in which the only AS to appear in the AS Path attribute is the speaker’s own AS (normally appears once but may appear multiple times if AS prepending is applied). The second case is that in which the BGPSEC speaker receives a route advertisement from a peer and then decides to propagate the route advertisement to an external (eBGP) peer (Section 4.2). That is, the BGPSEC speaker has received a BGPSEC update

message and is constructing a new update message for the same NLRI in which the AS Path attribute will contain AS number(s) other than the speaker's own AS.

In the remaining case where the BGPSEC speaker is sending the update message to an internal (iBGP) peer, the BGPSEC speaker populates the BGPSEC_Path_Signatures attribute by copying the BGPSEC_Path_Signatures attribute from the received update message. That is, the BGPSEC_Path_Signatures attribute is copied verbatim. Note that in the case that a BGPSEC speaker chooses to forward to an iBGP peer a BGPSEC update message that has not been successfully validated (see Section 5), the BGPSEC_Path_Signatures attribute SHOULD NOT be removed. (See Section 7 for the security ramifications of removing BGPSEC signatures.)

The information protected by the signature on a BGPSEC update message includes the AS number of the peer to whom the update message is being sent. Therefore, if a BGPSEC speaker wishes to send a BGPSEC update to multiple BGP peers, it MUST generate a separate BGPSEC update message for each unique peer AS to which the update message is sent.

A BGPSEC update message MUST advertise a route to only a single NLRI. This is because a BGPSEC speaker receiving an update message with multiple NLRI is unable to construct a valid BGPSEC update message (i.e., valid path signatures) containing a subset of the NLRI in the received update. If a BGPSEC speaker wishes to advertise routes to multiple NLRI, then it MUST generate a separate BGPSEC update message for each NLRI.

Note that in order to create or add a new signature to a Signature-List Block for a given algorithm suite, the BGPSEC speaker must possess a private key suitable for generating signatures for this algorithm suite. Additionally, this private key must correspond to the public key in a valid Resource PKI end-entity certificate whose AS number resource extension includes the BGPSEC speaker's AS number. Note also new signatures are only added to a BGPSEC update message when a BGPSEC speaker is generating an update message to send to an external peer (i.e., when the AS number of the peer is not equal to the BGPSEC speaker's own AS number). Therefore, a BGPSEC speaker who only sends BGPSEC update messages to peers within its own AS, it does not need to possess any private signature keys.

4.1. Originating a New BGPSEC Update

In an update message that originates a new route advertisement (i.e., an update whose AS_Path contains a single AS number), the BGPSEC speaker creates one Signature-List Block for each algorithm suite

that will be used. Typically, a BGPSEC speaker will use only a single algorithm suite. However, to ensure backwards compatibility during a period of transition from a 'current' algorithm suite to a 'new' algorithm suite, it will be necessary to originate update messages containing Signature-List Blocks for both the 'current' and the 'new' algorithm suites (see Section 6.1).

The Resource PKI enables the legitimate holder of IP address prefix(es) to issue a signed object, called a Route Origination Authorization (ROA), that authorizes a given AS to originate routes to a given set of prefixes (see [6]). Note that validation of a BGPSEC update message will fail (i.e., the validation algorithm, specified in Section 5.1, returns 'Not Good') unless there exists a valid ROA authorizing the first AS in the AS PATH attribute to originate routes to the prefix being advertised. Therefore, a BGPSEC speaker SHOULD NOT originate a BGPSEC update advertising a route for a given prefix unless there exists a valid ROA authorizing the BGPSEC speaker's AS to originate routes to this prefix.

The Expire Time field is set to specify a time at which the route advertisement specified in the update message will cease to be valid. Once the Expire Time has been reached, all BGPSEC speakers who have received the advertisement will treat it as invalid. The purpose of this field is to protect the BGPSEC speaker against attacks in which a malicious BGPSEC peer either replays a stale update message, or else fails to propagate the withdrawal for a prefix.

It is therefore necessary for the originating BGPSEC speaker to issue a new BGPSEC update, for the given prefix, prior to reaching the Expire Time. Setting appropriate values for Expire Time and for the rate at which new updates are sent out for a given prefix is an operational choice that involves trade offs between the window of replay protection versus network and processing load. Therefore, these settings are discussed in more detail in BGPSEC Operational Considerations document [9].

When originating a new route advertisement, each Signature-List Block MUST consist of a single Signature-Segment. The following describes how the BGPSEC speaker populates the fields of the Signature-List Block (see Section 3 for more information on the syntax of Signature-List Blocks).

The pCount field is typically set to the value 1. However, a BGPSEC speaker may set the pCount field to a value greater than 1. Setting the pCount field to a value greater than one has the same semantics as repeating an AS number multiple times in the AS-PATH of a non-BGPSEC update message (e.g., for traffic engineering purposes). However, even when the pCount field is set to a value greater than 1,

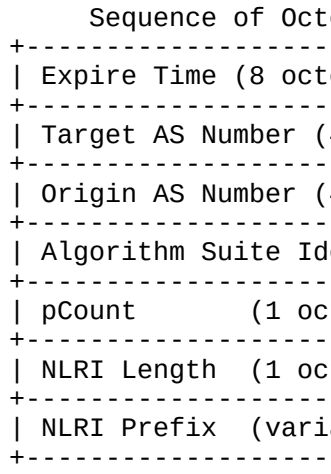
the BGPSEC speaker still only places a single copy of its AS number in the AS-PATH attribute. This is because the BGPSEC validation algorithm (see Section 5) requires a one-to-one correspondence between signatures and AS numbers in the AS-PATH. That is, setting a pCount value greater than 1 achieves the same semantics as repetition, but requires the generation of only a single signatures. Whereas a BGPSEC update message with actual repetition in the AS-PATH attribute would fail validation unless the BGPSEC speaker generated multiple signatures (one for each copy of the AS number placed in the AS-PATH).

The Subject Key Identifier field (see Section 3) is populated with the identifier contained in the Subject Key Identifier extension of the RPKI end-entity certificate used by the BGPSEC speaker. This Subject Key Identifier will be used by recipients of the route advertisement to identify the proper certificate to use in verifying the signature.

The Subject Key Identifier Length field is populated with the length (in octets) of the Subject Key Identifier.

The Signature field contains a digital signature that binds the NLRI, AS_Path attribute and BGPSEC_Path_Signatures attribute to the RPKI end-entity certificate used by the BGPSEC speaker. The digital signature is computed as follows:

- Construct a sequence of octets by concatenating the Expire Time, Target AS Number, Origin AS Number, Algorithm Suite Identifier, pCount and NLRI. The Target AS Number is the AS to whom the BGPSEC speaker intends to send the update message. (Note that the Target AS number is the AS number announced by the peer in the OPEN message of the BGP session within which the update is sent.) The Origin AS number prepend to this sequence the Target AS (the AS to whom the BGPSEC speaker intends to send the update message) and the Origin AS Number refers to the AS of the BGPSEC speaker who is originating the route advertisement.



- o Apply to this octet sequence the digest algorithm (for the algorithm suite of this Signature-List) to obtain a digest value.

- o Apply to this digest value the signature algorithm, (for the algorithm suite of this Signature-List) to obtain the digital signature. Then populate the Signature Field with this digital signature.

4.2. Propagating a Route Advertisement

When a BGPSEC speaker receives a BGPSEC update message containing a BGPSEC_Path_Signatures algorithm (with one or more signatures) from a (internal or external) peer, it may choose to propagate the route advertisement by sending to its (internal or external) peers by creating a new BGPSEC advertisement for the same prefix.

A BGPSEC speaker MUST NOT generate an update message containing the BGPSEC_Path_Signatures attribute unless it has selected, as the best route to the given prefix, a route that it received in an update message containing the BGPSEC_Path_Signatures attribute. In particular, this means that whenever a BGPSEC speaker generates an update message with a BGPSEC_Path_Signatures attribute that it will possess a received update message for the same prefix that also contains a BGPSEC_Path_Signatures attribute.

Additionally, whenever a BGPSEC speaker selects as the best route to a given prefix a route that it received in an update message containing the BGPSEC_Path_Signatures attribute, it is RECOMMENDED that if the BGPSEC speaker chooses to propagate the route that it generate an update message containing the BGPSEC_Path_Signatures attribute. However, a BGPSEC speaker MAY propagate a route

advertisement by generating a (non-BGPSEC) update message that does not contain the BGPSEC_Path_Signatures attribute. Note that if a BGPSEC speaker receives a route advertisement containing the BGPSEC_Path_Signatures attribute and chooses for any reason (e.g., its peer is a non-BGPSEC speaker) to propagate the route advertisement as a non-BGPSEC update message without the BGPSEC_Path_Signatures attribute, then it MUST follow the instructions in Section 4.2.1.

Note that removing BGPSEC signatures (i.e., propagating a route advertisement without the BGPSEC_Path_Signatures attribute) has significant security ramifications. (See Section 7 for discussion of the security ramifications of removing BGPSEC signatures.) Therefore, when a route advertisement is received via a BGPSEC update message, propagating the route advertisement without the BGPSEC_Path_Signatures attribute is NOT RECOMMENDED. Furthermore, note that when a BGPSEC speaker propagates a route advertisement with the BGPSEC_Path_Signatures attribute it is attesting to the fact that: (1) it received a BGPSEC update message that advertised this route; and (2) it chose this route as its best path to the given prefix. That is, the BGPSEC speaker is not attesting to the validation state of the update message it received. (See Section 7 for more discussion of the security semantics of BGPSEC signatures.)

If the BGPSEC speaker is producing an update message which contains an AS-SET (e.g., the BGPSEC speaker is performing proxy aggregation), then the BGPSEC speaker MUST NOT include the BGPSEC_Path_Signatures attribute. In such a case, the BGPSEC speaker must remove any existing BGPSEC_Path_Signatures in the received advertisement(s) for this prefix and produce a standard (non-BGPSEC) update message.

To generate the BGPSEC_Path_Signatures attribute on the outgoing update message, the BGPSEC first copies the Expire Time directly from the received update message to the new update message (that it is constructing). Note that the BGPSEC speaker MUST NOT change the Expire Time as any change to Expire Time will cause the new BGPSEC update message to fail validation (see Section 5).

If the received BGPSEC update message contains two Signature-List Blocks and the BGPSEC speaker supports both of the corresponding algorithms suites, then the BGPSEC speaker SHOULD generate a new update message that includes both of the Signature-List Blocks. If the received BGPSEC update message contains two Signature-List Blocks and the BGPSEC speaker only supports one of the two corresponding algorithm suites, then the BGPSEC speaker MUST remove the Signature-List Block corresponding to the algorithm suite that it does not understand. If the BGPSEC speaker does not support the algorithm suites in any of the Signature-List Blocks contained in the received

update message, then the BGPSEC speaker MUST NOT propagate the route advertisement with the BGPSEC_Path_Signatures attribute. (See Section 4.2.1 for information on removing the BGPSEC_Path_Signatures attribute when propagating route advertisements.)

Note that in the case where there are two Signature-List Blocks (corresponding to different algorithm suites) that the validation algorithm (see Section 5.1) deems a BGPSEC update message to be 'Good' if there is at least one supported algorithm suite (and corresponding Signature-List Block) that is deemed 'Good'. This means that a 'Good' BGPSEC update message may contain a Signature-List Block which is deemed 'Not Good' (e.g., contains signatures that the BGPSEC is unable to verify). Nonetheless, such Signature-List Blocks MUST NOT be removed. (See Section 7 for a discussion of the security ramifications of this design choice.)

For each Signature-List Block corresponding to an algorithm suite that the BGPSEC speaker does support, the BGPSEC speaker then adds a new Signature-Segment to the Signature-List Block. This Signature-Segment is prepended to the list of Signature-Segments (placed in the first position) so that the list of Signature-Segments appears in the same order as the corresponding AS numbers in the AS-Path attribute. The BGPSEC speaker populates the fields of this new signature-segment as follows.

The pCount is typically set to the value 1. A BGPSEC speaker may set the pCount field to a value greater than 1. (See Section 4.1 for a discussion of setting pCount to a value greater than 1.) A route server that participates in the BGP control path, but does not act as a transit AS in the data plane, may choose to set pCount to 0. This option enables the route server to participate in BGPSEC and obtain the associated security guarantees without increasing the effective length of the AS-PATH. (Note that BGPSEC speakers compute the effective length of the AS-PATH by summing the pCount values in the BGPSEC_Path_Signatures attribute, see Section 5.) However, when a route server sets the pCount value to 0, it still inserts its AS number into the AS-PATH, as this information is needed to validate the signature added by the route server. Note that the option of setting pCount to 0 is intended only for use by route servers that desire not to increase the effective AS-PATH length of routes they advertise. The pCount field SHOULD NOT be set to 0 in other circumstances. BGPSEC speakers SHOULD drop incoming update messages with pCount set to zero in cases where the BGPSEC speaker does not expect its peer to set pCount to zero (i.e., cases where the peer is not acting as a route server).

The Subject Key Identifier field in the new segment is populated with the identifier contained in the Subject Key Identifier extension of

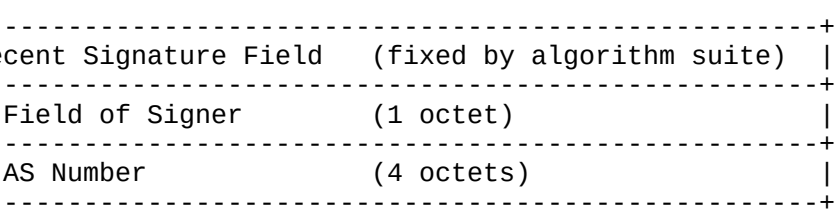
the RPKI end-entity certificate used by the BGPSEC speaker. This Subject Key Identifier will be used by recipients of the route advertisement to identify the proper certificate to use in verifying the signature.

The Subject Key Identifier Length field is populated with the length (in octets) of the Subject Key Identifier.

The Signature field in the new segment contains a digital signature that binds the NLRI, AS_Path attribute and BGPSEC_Path_Signatures attribute to the RPKI end-entity certificate used by the BGPSEC speaker. The digital signature is computed as follows:

- Construct a sequence of octets by concatenating the signature field of the most recent Signature-Segment (the one corresponding to AS from whom the BGPSEC speaker's AS received the announcement) with the pCount field inserted by the signer, and the Target AS (the AS to whom the BGPSEC speaker intends to send the update message). Note that the Target AS number is the AS number announced by the peer in the OPEN message of the BGP session within which the BGPSEC update message is sent.

Sequence of Octets to be Signed



- Apply to this octet sequence the digest algorithm (for the algorithm suite of this Signature-List) to obtain a digest value.

- Apply to this digest value the signature algorithm, (for the algorithm suite of this Signature-List) to obtain the digital signature. Then populate the Signature Field with this digital signature.

4.2.1. Propogating an Update without the Path_Signatures attribute

As discussed earlier in Section 4.2, a BGPSEC speaker may receive a BGPSEC update message that contains the BGPSEC_Path_Signatures Attribute and propagate the associated route in a non-BGPSEC update message that does not contain the BGPSEC_Path_Signatures attribute.

A BGPSEC speaker MUST remove the BGPSEC_Path_Signatures attribute

when propagating a route advertisement to a peer that has not advertised support BGPSEC (see Section 2), when propagating a route advertisement that contains an AS-SET in the AS-PATH, or when the BGPSEC speaker does not support any algorithm suite used to generate signatures in the received update message. In all other cases, the BGPSEC speaker SHOULD NOT remove the BGPSEC_Path_Signatures attribute.

When the BGPSEC speaker receives a BGPSEC update message that contains the BGPSEC_Path_Signatures Attribute and propagates the associated route in a non-BGPSEC update message, the BGPSEC MUST perform a transformation on the AS-PATH in the non-BGPSEC update message that it generates. The reason for this is that the AS-PATH attribute has slightly different semantics in a BGPSEC update message than it has in a non-BGPSEC update message.

To generate the AS-PATH in the outgoing non-BGPSEC update message, the BGPSEC speaker performs the following steps for each AS number in the AS-PATH of the received BGPSEC update message. (Note that there is a one-to-one correspondence between the AS numbers in the AS-PATH of a BGPSEC update message and the Signature Segments in the Signature-List Block of the BGPSEC_Path_Signatures attribute. The follows step will make use of this correspondence.)

- For each AS number in the AS-PATH of the received BGPSEC update message, locate the pCount value in the corresponding Signature Segment.

- If the pCount value is equal to 0, then do not include the corresponding AS in the AS-PATH of the outgoing non-BGPSEC update message.

- If the pCount value is greater than or equal to 1, insert into the AS-PATH of the outgoing update message a number of copies of the corresponding AS number equal to the pCount value.

Other than the above transformation that is applied to the AS-PATH, no additional special behavior is required when removing BGPSEC signatures from BGPSEC update messages. That is, all other attributes in the outgoing non-BGPSEC update message are generated as they would normally be generated by the BGP speaker in a non-BGPSEC update message.

5. Processing a Received BGPSEC Update

Validation of a BGPSEC update messages makes use of data from RPKI certificates and signed Route Origination Authorizations (ROA). In

particular, to validate update messages containing the BGPSEC_Path_Signatures attribute, it is necessary that the recipient have access to the following data obtained from valid RPKI certificates and ROAs:

- For each valid RPKI end-entity certificate containing an AS Number extension, the AS Number, Public Key and Subject Key Identifier are required

- For each valid ROA, the AS Number and the list of IP address prefixes

Note that the BGPSEC speaker could perform the validation of RPKI certificates and ROAs on its own and extract the required data, or it could receive the same data from a trusted cache that performs RPKI validation on behalf of (some set of) BGPSEC speakers. (The latter case in analogous to the use of the RPKI-RTR protocol [10] for origin validation.)

To validate a BGPSEC update message containing the BGPSEC_Path_Signatures attribute, the recipient performs the validation steps specified in Section 5.1. The validation procedure results in one of two states: 'Good' and 'Not Good'.

It is expected that the output of the validation procedure will be used as an input to BGP route selection. However, BGP route selection and thus the handling of the two validation states is a matter of local policy, and shall be handled using existing local policy mechanisms. It is expected that BGP peers will generally prefer routes received via 'Good' BGPSEC update messages over routes received via 'Not Good' BGPSEC update messages as well as routes received via update messages that do not contain the BGPSEC_Path_Signatures attribute. However, BGPSEC specifies no changes to the BGP decision process and leaves to the operator the selection of an appropriate policy mechanism to achieve the operator's desired results within the BGP decision process.

BGPSEC validation need only be performed at eBGP edge. The validation status of a BGP signed/unsigned update MAY be conveyed via iBGP from an ingress edge router to an egress edge router. Local policy in the AS determines the specific means for conveying the validation status through various pre-existing mechanisms (e.g., modifying an attribute). As discussed in Section 4, when a BGPSEC speaker chooses to forward a (syntactically correct) BGPSEC update message, it SHOULD be forwarded with its BGPSEC_Path_Signatures attribute intact (regardless of the validation state of the update message). Based entirely on local policy settings, an egress router MAY trust the validation status conveyed by an ingress router or it

MAY perform its own validation.

Upon receiving a BGPSEC update message, a BGPSEC speaker SHOULD sum the pCount values within BGPSEC_Path_Signatures attribute to determine the effective length of the AS Path. The BGPSEC speaker SHOULD use this sum of pCount values in precisely the same way as it uses the length of the AS Path in non-BGPSEC update messages.

5.1. Validation Algorithm

This section specifies an algorithm for validation of BGPSEC update messages. A conformant implementation MUST include an BGPSEC update validation algorithm that is functionally equivalent to the external behavior of this algorithm.

First, the recipient of a BGPSEC update message performs a check to ensure that the message is properly formed. Specifically, the recipient performs the following checks:

- Check to ensure that the entire BGPSEC_Path_Signatures attribute is syntactically correct (conforms to the specification in this document).
- Check to ensure that the AS-Path attribute contains no AS-Set segments.
- Check that each Signature-List Block contains one Signature-Segment for each AS in the AS-Path attribute. (Note that the entirety of each Signature-List Block must be checked to ensure that it is well formed, even though the validation process may terminate before all signatures are cryptographically verified.)

If there are two Signature-List Blocks within the BGPSEC_Path_Signatures attribute and one of them is poorly formed (or contains the wrong number of Signature-Segments) , then the recipient should log that an error occurred, strip off that particular Signature-List Block and process the update message as though it arrived with a single Signature-List Block. If the BGPSEC_Path_Signatures attribute contains a syntax error that is not local to one of two Signature-List Blocks, then the recipient should log that an error occurred and drop the update message containing the error. Similarly, if an update message contains both the BGPSEC_Path_Signatures attribute and an AS-Path attribute that contains an AS-Set segment, then the recipient should log that an error occurred and drop the update message containing the error.

Second, the BGPSEC speaker verifies that the update message has not yet expired. To do this, locate the Expire Time field in the

BGPSEC_Path_Signatures attribute, and compare it with the current time. If the current time is later than the Expire Time, the BGPSEC update is 'Not Good' and the validation algorithm terminates.

Third, the BGPSEC speaker verifies that the origin AS is authorized to advertise the prefix in question. To do this, consult the valid ROA data to obtain a list of AS numbers that are associated with the given IP address prefix in the update message. Then locate the last (least recently added) AS number in the AS-Path. If the origin AS in the AS-Path is not in the set of AS numbers associated with the given prefix, then BGPSEC update message is 'Not Good' and the validation algorithm terminates.

Finally, the BGPSEC speaker examines the Signature-List Blocks in the BGPSEC_Path_Signatures attribute. Any Signature-List Block corresponding to an algorithm suite that the BGPSEC speaker does not support is not considered in validation. If there does not exist a Signature-List Block corresponding to an algorithm suite that the BGPSEC speaker supports, then the BGPSEC speaker MUST treat the update message in the same manner that the BGPSEC speaker would treat an update message that arrived without a BGPSEC_Path_Signatures attribute.

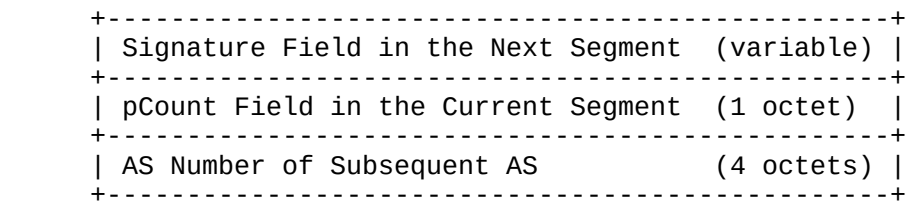
For each remaining Signature-List Block (corresponding to an algorithm suite supported by the BGPSEC speaker), the BGPSEC speaker iterates through the Signature-Segments in the Signature-List block, starting with the most recently added segment (and concluding with the least recently added segment). Note that there is a one-to-one correspondence between Signature-Segments and AS numbers in the AS-Path attribute, and the following steps make use of this correspondence.

- (Step I): Locate the public key needed to verify the signature (in the current Signature-Segment). To do this, consult the valid RPKI end-entity certificate data and look for an SKI that matches the value in the SKI field of the Signature-Segment. If no such SKI value is found in the valid RPKI data then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature-List Block. Similarly, if the SKI exists but the AS Number associated with the SKI does NOT match the AS Number (in the AS-Path attribute) which corresponds to the current Signature-Segment, then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature-List Block.

- (Step II): Compute the digest function (for the given algorithm suite) on the appropriate data. If the segment is not the (least recently added) segment corresponding to the origin AS, then the digest function should be computed on the following sequence of

octets:

Sequence of Octets to be Hashed



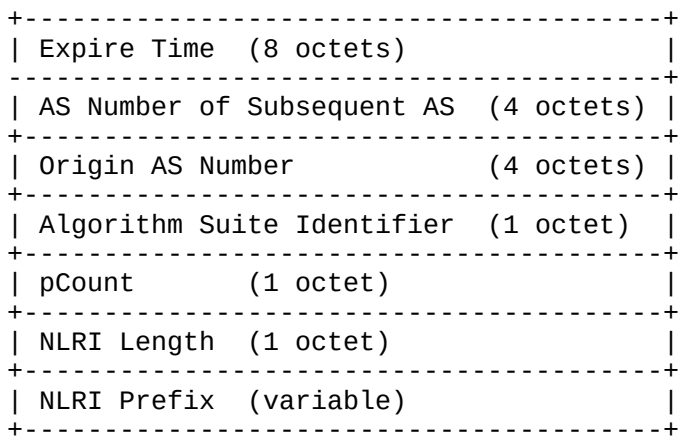
The 'Signature Field in the Next Segment' is the Signature field found in the Signature-Segment that is next to be processed (that is, the next most recently added Signature- Segment). The 'pCount Field in the Current Segment' is the pCount field found in the Signature-Segment that is currently being processed.

For the first segment to be processed (the most recently added segment), the 'AS Number of Subsequent AS' is the AS number of the BGPSEC speaker validating the update message. Note that if a BGPSEC speaker uses multiple AS Numbers (e.g., the BGPSEC speaker is a member of a confederation), the AS number used here MUST be the AS number announced in the OPEN message for the BGP session over which the BGPSEC update was received.

For each other Signature-Segment, the 'AS Number of Subsequent AS' is the AS that corresponds to the Signature-Segment added immediately after the one being processed. (That is, find the AS number corresponding to the Signature-Segment currently being processed and the 'AS Number of Subsequent AS' is the next AS number that was added to the AS-Path attribute.)

Alternatively, if the segment being processed corresponds to the origin AS, then the digest function should be computed on the following sequence of octets:

Sequence of Octets to be Hashed



The NLRI Length, NLRI Prefix, Expire Time, and Algorithm Suite Identifier are all obtained in a straight forward manner from the NLRI of the update message or the BGPSEC_Path_Signatures attribute being validated. The pCount field is taken from the Signature-Segment currently being processed.

The Origin AS Number is the same Origin AS Number that was located in Step I above. (That is, the AS number corresponding to the least recently added Signature-Segment.)

The 'AS Number of Subsequent AS' is the AS Number added to the AS-Path immediately after the Origin AS Number. (That is, the second AS Number that was added to the AS Path.)

- (Step III): Use the signature validation algorithm (for the given algorithm suite) to verify the signature in the current segment. That is, invoke the signature validation algorithm on the following three inputs: the value of the Signature field in the current segment; the digest value computed in Step II above; and the public key obtained from the valid RPKI data in Step I above. If the signature validation algorithm determines that the signature is invalid, then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature-List Block. If the signature validation algorithm determines that the signature is valid, then continue processing Signature-Segments (within the current Signature-List Block).

If all Signature-Segments within a Signature-List Block pass validation (i.e., all segments are processed and the Signature-List Block has not yet been marked 'Not Good'), then the Signature-List

Block is marked as 'Good'.

If at least one Signature-List Block is marked as 'Good', then the validation algorithm terminates and the BGPSEC update message is deemed to be 'Good'. (That is, if a BGPSEC update message contains two Signature-List Blocks then the update message is deemed 'Good' if the first Signature-List block is marked 'Good' OR the second Signature-List block is marked 'Good'.)

6. Algorithms and Extensibility

6.1. Algorithm Suite Considerations

Note that there is currently no support for bilateral negotiation between BGPSEC peers to use of a particular (digest and signature) algorithm suite using BGP capabilities. This is because the algorithm suite used by the sender of a BGPSEC update message must be understood not only by the peer to whom he is directly sending the message, but also by all BGPSEC speakers to whom the route advertisement is eventually propagated. Therefore, selection of an algorithm suite cannot be a local matter negotiated by BGP peers, but instead must be coordinated throughout the Internet.

To this end, a mandatory algorithm suites document will be created which specifies a mandatory-to-use 'current' algorithm suite for use by all BGPSEC speakers. Additionally, the document specifies an additional 'new' algorithm suite that is recommended to implement.

It is anticipated that in the future the mandatory algorithm suites document will be updated to specify a transition from the 'current' algorithm suite to the 'new' algorithm suite. During the period of transition (likely a small number of years), all BGPSEC update messages SHOULD simultaneously use both the 'current' algorithm suite and the 'new' algorithm suite. (Note that Sections 3 and 4 specify how the BGPSEC_Path_Signatures attribute can contain signatures, in parallel, for two algorithm suites.) Once the transition is complete, use of the old 'current' algorithm will be deprecated, use of the 'new' algorithm will be mandatory, and a subsequent 'even newer' algorithm suite may be specified as recommend to implement. Once the transition has successfully been completed in this manner, BGPSEC speakers SHOULD include only a single Signature-List Block (corresponding to the 'new' algorithm).

6.2. Extensibility Considerations

This section discusses potential changes to BGPSEC that would require substantial changes to the processing of the BGPSEC_Path_Signatures

and thus necessitate a new version of BGPSEC. Examples of such changes include:

- o A new type of signature algorithm that produces signatures of variable length
- o A new type of signature algorithm for which the number of signatures in the Signature-List Block is not equal to the number of ASes in the AS-PATH (e.g., aggregate signatures)
- o Changes to the data that is protected by the BGPSEC signatures (e.g., protection of attributes other than AS-PATH)

In the case that such a change to BGPSEC were deemed desirable, it is expected that a subsequent version of BGPSEC would be created and that this version of BGPSEC would specify a new BGP Path Attribute, let's call it BGPSEC_PATH_SIG_TWO, which is designed to accommodate the desired changes to BGPSEC. In such a case, the mandatory algorithm suites document would be updated to specify algorithm suites appropriate for the new version of BGPSEC.

At this point a transition would begin which is analogous to the algorithm transition discussed in Section 6.2. During the transition period all BGPSEC speakers SHOULD simultaneously include both the BGPSEC_PATH_SIGNATURES attribute and the new BGPSEC_PATH_SIG_TWO attribute. Once the transition is complete, the use of BGPSEC_PATH_SIGNATURES could then be deprecated, at which point BGPSEC speakers SHOULD include only the new BGPSEC_PATH_SIG_TWO attribute. Such a process could facilitate a transition to a new BGPSEC semantics in a backwards compatible fashion.

7. Security Considerations

For discussion of the BGPSEC threat model and related security considerations, please see [8].

A BGPSEC speaker who receives a valid BGPSEC update message, containing a route advertisement for a given prefix, is provided with the following security guarantees:

- o The origin AS number corresponds to an autonomous system that has been authorized by the IP address space holder to originate route advertisements for the given prefix.
- o For each subsequent AS number in the AS-Path, a BGPSEC speaker authorized by the holder of the AS number selected the given route as the best route to the given prefix.

- For each AS number in the AS Path, a BGPSEC speaker authorized by the holder of the AS number intentionally propagated the route advertisement to the next AS in the AS-Path.

That is, the recipient of a valid BGPSEC Update message is assured that the AS-Path corresponds to a sequence of autonomous systems who have all agreed in principle to forward packets to the given prefix along the indicated path. (It should be noted BGPSEC does not offer a precise guarantee that the data packets would propagate along the indicated path; it only guarantees that the BGP update conveying the path indeed propagated along the indicated path.) Furthermore, the recipient is assured that this path terminates in an autonomous system that has been authorized by the IP address space holder as a legitimate destination for traffic to the given prefix.

Note that although BGPSEC provides a mechanism for an AS to validate that a received update message has certain security properties, the use of such a mechanism to influence route selection is completely a matter of local policy. Therefore, a BGPSEC speaker can make no assumptions about the validity of a route received from an external BGPSEC peer. That is, a compliant BGPSEC peer may (depending on the local policy of the peer) send update messages that fail the validity test in Section 5. Thus, a BGPSEC speaker MUST completely validate all BGPSEC update messages received from external peers. (Validation of update messages received from internal peers is a matter of local policy, see Section 5).

Note that there may be cases where a BGPSEC speaker deems 'Good' (as per the validation algorithm in Section 5.1) a BGPSEC update message that contains both a 'Good' and a 'Not Good' Signature-List Block. That is, the update message contains two sets of signatures corresponding to two algorithm suites, and one set of signatures verifies correctly and the other set of signatures fails to verify. In this case, the protocol specifies that if the BGPSEC speaker propagates the route advertisement received in such an update message then the BGPSEC speaker SHOULD add its signature to each of the Signature-List Blocks using both the corresponding algorithm suite. Thus the BGPSEC speaker creates a signature using both algorithm suites and creates a new update message that contains both the 'Good' and the 'Not Good' set of signatures (from its own vantage point).

To understand the reason for such a design decision consider the case where the BGPSEC speaker receives an update message with both a set of algorithm A signatures which are 'Good' and a set of algorithm B signatures which are 'Not Good'. In such a case it is possible (perhaps even quite likely) that some of the BGPSEC speaker's peers (or other entities further 'downstream' in the BGP topology) do not support algorithm A. Therefore, if the BGPSEC speaker were to remove

the 'Not Good' set of signatures corresponding to algorithm B, such entities would treat the message as though it were unsigned. By including the 'Not Good' set of signatures when propagating a route advertisement, the BGPSEC speaker ensures that 'downstream' entities have as much information as possible to make an informed opinion about the validation status of a BGPSEC update.

Note also that during a period of partial BGPSEC deployment, a 'downstream' entity might reasonably treat unsigned messages different from BGPSEC updates that contain a single set of 'Not Good' signatures. That is, by removing the set of 'Not Good' signatures the BGPSEC speaker might actually cause a downstream entity to 'upgrade' the status of a route advertisement from 'Not Good' to unsigned. Finally, note that in the above scenario, the BGPSEC speaker might have deemed algorithm A signatures 'Good' only because of some issue with RPKI state local to his AS (for example, his AS might not yet have obtained a CRL indicating that a key used to verify an algorithm A signature belongs to a newly revoked certificate). In such a case, it is highly desirable for a downstream entity to treat the update as 'Not Good' (due to the revocation) and not as 'unsigned' (which would happen if the 'Not Good' Signature-List Blocks were removed).

A similar argument applies to the case where a BGPSEC speaker (for some reason such as lack of viable alternatives) selects as his best route to a given prefix a route obtained via a 'Not Good' BGPSEC update message. (That is, a BGPSEC update containing only 'Not Good' Signature-List Blocks.) In such a case, the BGPSEC speaker should propagate a signed BGPSEC update message, adding his signature to the 'Not Good' signatures that already exist. Again, this is to ensure that 'downstream' entities are able to make an informed decision and not erroneously treat the route as unsigned. It may also be noted here that due to possible differences in RPKI data at different vantage points in the network, a BGPSEC update that was deemed 'Not Good' at an upstream BGPSEC speaker may indeed be deemed 'Good' at another BGP speaker downstream.

Therefore, it is important to note that when a BGPSEC speaker signs an outgoing update message, it is not attesting to a belief that all signatures prior to its are valid. Instead it is merely asserting that:

1. The BGPSEC speaker received the given route advertisement with the indicated NLRI and AS Path;

2. The BGPSEC speaker selected this route as the best route to the given prefix; and

3. The BGPSEC speaker chose to propagate an advertisement for this route to the peer (implicitly) indicated by the 'Target AS'

The BGPSEC update validation procedure is a potential target for denial of service attacks against a BGPSEC speaker. To mitigate the effectiveness of such denial of service attacks, BGPSEC speakers should implement an update validation algorithm that performs expensive checks (e.g., signature verification) after less expensive checks (e.g., syntax checks). The validation algorithm specified in Section 5.1 was chosen so as to perform checks which are likely to be expensive after checks that are likely to be inexpensive. However, the relative cost of performing required validation steps may vary between implementations, and thus the algorithm specified in Section 5.1 may not provide the best denial of service protection for all implementations.

Finally, the mechanism of setting the pCount field to zero is included in this specification to enable route servers in the control path to participate in BGPSEC without increasing the effective length of the AS-PATH. However, entities other than route servers could conceivably use this mechanism (set the pCount to zero) to attract traffic (by reducing the effective length of the AS-PATH) illegitimately. This risk is largely mitigated if every BGPSEC speaker drops incoming update messages that set pCount to zero but come from a peer that is not a route server. However, note that a recipient of a BGPSEC update message in which an upstream entity that is two or more hops away set pCount to zero is unable to verify for themselves whether pCount was set to zero legitimately.

8. IANA Considerations

IANA is requested to create a registry of BGPSEC algorithm suite identifiers. This registry shall contain four fields, a one octet Algorithm Suite Identifier, the name of the suite's digest algorithm, the name of the suite's signature algorithm, and a specification pointer containing a reference to the formal specification of the algorithm suite. That is, entries in the registry have the following form:

Algorithm Suite Identifier	Digest Algorithm	Signature Algorithm	Specification Pointer

The entries in this registry shall be managed by IETF consensus.

9. Contributors

9.1. Authors

Rob Austein
Dragon Research Labs
sra@hactrn.net

Steven Bellovin
Columbia University
smb@cs.columbia.edu

Randy Bush
Internet Initiative Japan
randy@psg.com

Russ Housley
Vigil Security
housley@vigilsec.com

Matt Lepinski
BBN Technologies
lepinski@bbn.com

Stephen Kent
BBN Technologies
kent@bbn.com

Warren Kumari
Google
warren@kumari.net

Doug Montgomery
USA National Institute of Standards and Technology
dougm@nist.gov

Kotikalapudi Sriram
USA National Institute of Standards and Technology
kotikalapudi.sriram@nist.gov

Samuel Weiler
Cobham
weiler+ietf@watson.org

9.2. Acknowledgements

The authors would like to thank Luke Berndt, Sharon Goldberg, Ed Kern, Chris Morrow, Doug Maughan, Pradosh Mohapatra, Russ Mundy, Sandy Murphy, Keyur Patel, Mark Reynolds, Heather Schiller, Jason Schiller, John Scudder, Ruediger Volk and David Ward for their valuable input and review.

10. Normative References

[1] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4", RFC 4271, January 2006.

[2] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

[3] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, February 2009.

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

[5] Patel, K., Ward, D., and R. Bush, "Extended Message support for BGP", March 2011.

[6] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations", February 2011.

[7] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", February 2011.

[8] Kent, S., "Threat Model for BGP Path Security", June 2011.

[9] Bush, R., "BGPsec Operational Considerations", October 2011.

[10] Bush, R. and R. Austein, "The RPKI/Router Protocol", October 2011.

Author's Address

Matthew Lepinski (editor)
BBN
10 Moulton St
Cambridge, MA 55409
US

Phone: +1 617 873 5939
Email: mlepinski@bbn.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2012

S. Bellovin
Columbia University
R. Bush
Internet Initiative Japan
D. Ward
Juniper Networks
October 19, 2011

Security Requirements for BGP Path Validation
draft-ietf-sidr-bgpsec-reqs-01

Abstract

This document describes requirements for a future BGP security protocol design to provide cryptographic assurance that the origin AS had the right to announce the prefix and to provide assurance of the AS Path of the announcement.

Requirements Language

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 [RFC2119].

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 April 21, 2012.

Copyright Notice

Table of Contents

1. Introduction

RPKI-based Origin Validation ([I-D.ietf-sidr-pfx-validate]) provides a measure of resilience to accidental mis-origination of prefixes. But it provides neither cryptographic assurance (announcements are not signed), nor assurance of the AS Path of the announcement.

This document describes requirements to be placed on a BGP security protocol, herein termed BGPsec, intended to rectify these gaps.

The threat model assumed here is documented in [RFC4593] and [I-D.ietf-sidr-bgpsec-threats].

2. Recommended Reading

This document assumes knowledge of the RPKI see [I-D.ietf-sidr-arch] and the RPKI Repository Structure, see [I-D.ietf-sidr-repos-struct].

This document assumes ongoing incremental deployment of ROAs, see [I-D.ietf-sidr-roa-format], the RPKI to Router Protocol, see [I-D.ietf-sidr-rpki-rtr], and RPKI-based Prefix Validation, see [I-D.ietf-sidr-pfx-validate].

And, of course, a knowledge of BGP [RFC4271] is required.

3. General Requirements

The following are general requirements for a BGPsec protocol:

3.1 A BGPsec design must allow the receiver of a BGP announcement to determine, to a strong level of certainty, that the received PATH attribute accurately represents the sequence of eBGP exchanges that propagated the prefix from the origin AS to the receiver.

3.2 A BGPsec design must allow the receiver of an announcement to detect if an AS has added or deleted any AS number other than its own in the path attribute. This includes modification to the number of AS prepends.

3.3 A BGPsec design MUST be amenable to incremental deployment. Any incompatible protocol capabilities MUST be negotiated.

3.4 A BGPsec design MUST provide analysis of the operational considerations for deployment and particularly of incremental deployment, e.g, contiguous islands, non-contiguous islands, universal deployment, etc..

3.5 As cryptographic payloads and memory requirements on routers are likely to increase, a BGPsec design MAY require use of new hardware. I.e. compatibility with current hardware abilities is not a requirement that this document imposes on a solution. As BGPsec will likely not be rolled out for some years, this should not be a major problem.

3.6 A BGPsec design need not prevent attacks on data plane traffic. It need not provide assurance that the data plane even follows the control plane.

3.7 A BGPsec design MUST resist attacks by an enemy who has access to the link layer, per Section 3.1.1.2 of [RFC4593]. In particular, such a design must provide mechanisms for authentication of all data, including protecting against message insertion, deletion, modification, or replay. Mechanisms that suffice include TCP sessions authenticated with IPsec [RFC4301] or TLS [RFC5246].

3.8 A BGPsec design MAY make use of a security infrastructure (e.g., a PKI) to distribute authenticated data used as input to routing decisions. Such data include information about holdings of address space and ASNs, and assertions about binding of address space to ASNs.

3.9 If message signing increases message size, the 4096 byte limit on BGP PDU size MAY be removed.

3.10 It is entirely OPTIONAL to secure AS SETs and prefix aggregation. The long range solution to this is the deprecation of AS-SETs, see [I-D.ietf-idr-deprecate-as-sets].

3.11 If a BGPsec design uses signed prefixes, given the difficulty of splitting a signed message while preserving the signature, it need NOT handle multiple prefixes in a single UPDATE PDU.

3.12 A BGPsec design MUST enable each BGPsec speaker to configure use of the security mechanism on a per-peer basis.

3.13 A BGPsec design MUST provide backward compatibility in the message formatting, transmission, and processing of routing information carried through a mixed security environment. Message formatting in a fully secured environment MAY be

handled in a non-backward compatible manner.

3.14 While the trust level of an NLRI should be determined by the BGPsec protocol, local routing preference and policy MUST then be applied to best path and other decisions. Such mechanisms MUST conform with [I-D.ietf-sidr-ltamgmt].

3.15 A BGPsec design MUST support 'transparent' route servers, meaning that the AS of the route server is not counted in downstream BGP AS-path-length tie-breaking decisions.

3.16 If a BGPsec design makes use of a security infrastructure, that infrastructure SHOULD enable each network operator to select the entities it will trust when authenticating data in the security infrastructure. See, for example, [I-D.ietf-sidr-ltamgmt].

3.17 A BGPsec design MUST NOT require operators to reveal more than is currently revealed in the operational inter-domain routing environment, other than the inclusion of necessary security credentials to allow others to ascertain for themselves the necessary degree of assurance regarding the validity of NLRI received via BGPsec. This includes peering, customer, and provider relationships, an ISP's internal infrastructure, etc. It is understood that some data are revealed to the savvy seeker by BGP, traceroute, etc. today.

3.18 A BGPsec design SHOULD flag security exceptions which are significant enough to be logged. The specific data to be logged are an implementation matter.

3.19 Any routing information database MAY be re-authenticated periodically or in an event-driven manner, especially in response to events such as, for example, PKI updates.

3.20 Should a BGPsec design use hashes or signatures, it should provide mechanisms for algorithm agility.

3.21 A BGPsec design SHOULD NOT presume to know the intent of the originator of a NLRI, nor that of any AS on the AS Path.

3.22 A BGP listener SHOULD NOT trust non-BGPsec markings, such as communities, across trust boundaries.

4. BGP UPDATE Security Requirements

The following requirements MUST be met in the processing of BGP UPDATE messages:

4.1 A BGPsec design MUST enable each recipient of an UPDATE to formally validate that the origin AS in the message is authorized to originate a route to the prefix(es) in the message.

4.2 A BGPsec design MUST enable the recipient of an UPDATE to formally determine that the NLRI has traversed the AS path indicated in the UPDATE. Note that this is more stringent than showing that the path is merely not impossible.

4.3 Replay of BGP UPDATE messages need not be completely prevented, but a BGPsec design MUST provide a mechanism to control the window of exposure to replay attacks.

4.4 A BGPsec design SHOULD provide some level of assurance that the origin of a prefix is still 'alive', i.e. that a monkey in the middle has not withheld a WITHDRAW message or the effects thereof.

4.5 NLRI of the UPDATE message SHOULD be able to be authenticated in real-time as the message is processed.

4.6 Normal sanity checks of received announcements MUST be done, e.g. verification that the first element of the AS_PATH list corresponds to the locally configured AS of the peer from which the UPDATE was received.

4.7 The output of a router applying BGPsec to a received signed UPDATE MUST be either Valid or Unverified. There should be no shades of grey.

5. IANA Considerations

This document asks nothing of the IANA.

6. Security Considerations

The data plane may not follow the control plane.

Security for subscriber traffic is outside the scope of this document, and of BGP security in general. IETF standards for payload

data security should be employed. While adoption of BGP security measures may ameliorate some classes of attacks on traffic, these measures are not a substitute for use of subscriber-based security.

7. Acknowledgments

The author wishes to thank the authors of [I-D.ietf-rpsec-bgpsecrec] from whom we liberally stole, Russ Housley, Geoff Huston, Steve Kent, Sandy Murphy, John Scudder, Sam Weiler, and a number of others.

8. References

8.1. Normative References

[I-D.ietf-sidr-bgpsec-threats]
Kent, S., "Threat Model for BGP Path Security", draft-ietf-sidr-bgpsec-threats-00 (work in progress), June 2011.

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

[RFC4593] Barbir, A., Murphy, S., and Y. Yang, "Generic Threats to Routing Protocols", RFC 4593, October 2006.

8.2. Informative References

[I-D.ietf-idr-deprecate-as-sets]
Kumari, W. and K. Sriram, "Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP", draft-ietf-idr-deprecate-as-sets-06 (work in progress), October 2011.

[I-D.ietf-rpsec-bgpsecrec]
Christian, B. and T. Tauber, "BGP Security Requirements", draft-ietf-rpsec-bgpsecrec-10 (work in progress), November 2008.

[I-D.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch-13 (work in progress), May 2011.

[I-D.ietf-sidr-ltamgmt]
Reynolds, M. and S. Kent, "Local Trust Anchor Management for the Resource Public Key Infrastructure",

draft-ietf-sidr-ltamgmt-02 (work in progress), June 2011.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-02 (work in progress), July 2011.

[I-D.ietf-sidr-repos-struct]
Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", draft-ietf-sidr-repos-struct-09 (work in progress), July 2011.

[I-D.ietf-sidr-roa-format]
Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", draft-ietf-sidr-roa-format-12 (work in progress), May 2011.

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-18 (work in progress), October 2011.

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Authors' Addresses

Steven M. Bellovin
Columbia University
1214 Amsterdam Avenue, MC 0401
New York, New York 10027
US

Phone: +1 212 939 7149
Email: bellovin@acm.org

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Dave Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206
US

Phone: +1-408-745-2000
Email: dward@juniper.net

Network Working Group
Internet-Draft
Intended status: BCP
Expires: May 3, 2012

R. Bush
Internet Initiative Japan
October 31, 2011

RPKI-Based Origin Validation Operation
draft-ietf-sidr-origin-ops-12

Abstract

Deployment of RPKI-based BGP origin validation has many operational considerations. This document attempts to collect and present them. It is expected to evolve as RPKI-based origin validation is deployed and the dynamics are better understood.

Requirements Language

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 [RFC2119].

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 May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

RPKI-based origin validation relies on widespread deployment of the Resource Public Key Infrastructure (RPKI) [I-D.ietf-sidr-arch]. How the RPKI is distributed and maintained globally is a serious concern from many aspects.

The global RPKI is in very initial stages of deployment, there is no single root trust anchor, initial testing is being done by the IANA and the RIRs, and there is a technical testbed. It is thought that origin validation based on the RPKI will be deployed incrementally over the next year to five years.

Origin validation needs to be done only by an AS's border routers and is designed so that it can be used to protect announcements which are originated by any network participating in Internet BGP routing: large providers, upstreams and down-streams, and by small stub/ enterprise/edge routers.

Origin validation has been designed to be deployed on current routers without significant hardware upgrade. It should be used in border routers by operators from large backbones to small stub/entetprise/ edge networks.

RPKI-based origin validation has been designed so that, with prudent local routing policies, there is little risk that what is seen as today's normal Internet routing is threatened by imprudent deployment of the global RPKI, see Section 5.

2. Suggested Reading

It is assumed that the reader understands BGP, [RFC4271], the RPKI, see [I-D.ietf-sidr-arch], the RPKI Repository Structure, see [I-D.ietf-sidr-repos-struct], ROAs, see [I-D.ietf-sidr-roa-format], the RPKI to Router Protocol, see [I-D.ietf-sidr-rpki-rtr], RPKI-based Prefix Validation, see [I-D.ietf-sidr-pfx-validate], and Ghostbusters Records, see [I-D.ietf-sidr-ghostbusters].

3. RPKI Distribution and Maintenance

The RPKI is a distributed database containing certificates, CRLs, manifests, ROAs, and Ghostbusters Records as described in [I-D.ietf-sidr-repos-struct]. Policies and considerations for RPKI object generation and maintenance are discussed elsewhere.

A local valid cache containing all RPKI data may be gathered from the

global distributed database using the rsync protocol, [RFC5781], and a validation tool such as rcynic [rcynic].

Validated caches may also be created and maintained from other validated caches. Network operators SHOULD take maximum advantage of this feature to minimize load on the global distributed RPKI database. Of course, the recipient SHOULD re-validate the data.

Timing of inter-cache synchronization is outside the scope of this document, but depends on things such as how often routers feed from the caches, how often the operator feels the global RPKI changes significantly, etc.

As RPKI-based origin validation relies on the availability of RPKI data, operators SHOULD locate caches close to routers that require these data and services. 'Close' is, of course, complex. One should consider trust boundaries, routing bootstrap reachability, latency, etc.

For redundancy, a router SHOULD peer with more than one cache at the same time. Peering with two or more, at least one local and others remote, is recommended.

If an operator trusts upstreams to carry their traffic, they MAY also trust the RPKI data those upstreams cache, and SHOULD peer with caches made available to them by those upstreams. Note that this places an obligation on those upstreams to maintain fresh and reliable caches, and to make them available to their customers. And, as usual, the recipient SHOULD re-validate the data.

A transit provider or a network with peers SHOULD validate origins in announcements made by upstreams, down-streams, and peers. They still SHOULD trust the caches provided by their upstreams.

Before issuing a ROA for a super-block, an operator MUST ensure that any sub-allocations from that block which are announced by other ASs, e.g. customers, have correct ROAs in the RPKI. Otherwise, issuing a ROA for the super-block will cause the announcements of sub-allocations with no ROAs to be viewed as Invalid, see [I-D.ietf-sidr-pfx-validate].

Use of RPKI-based origin validation removes any need to originate more specifics into BGP to protect against mis-origination of a less specific prefix. Having a ROA for the covering prefix should protect it.

To aid translation of ROAs into efficient search algorithms in routers, ROAs SHOULD be as precise as possible, i.e. match prefixes

as announced in BGP. E.g. software and operators SHOULD avoid use of excessive max length values in ROAs unless operationally necessary.

One advantage of minimal ROA length is that the forged origin attack does not work for sub-prefixes that are not covered by overly long max length. E.g. if, instead of 10.0.0.0/16-24, one issues 10.0.0.0/16 and 10.0.42.0/24, a forged origin attack can not succeed against 10.0.66.0/24. They must attack the whole /16, which is more likely to be noticed.

Therefore, ROA generation software MUST use the prefix length as the max length if the user does not specify a max length.

Operators SHOULD be conservative in use of max length in ROAs. E.g., if a prefix will have only a few sub-prefixes announced, multiple ROAs for the specific announcements SHOULD be used as opposed to one ROA with a long max length.

If a prefix is legitimately announced by more than one AS, ROAs for all of the ASs SHOULD be issued so that all are considered Valid.

An environment where private address space is announced in eBGP the operator MAY have private RPKI objects which cover these private spaces. This will require a trust anchor created and owned by that environment, see [I-D.ietf-sidr-ltamgmt].

Operators owning prefix P should issue ROAs for all ASs which may announce P.

Operators issuing ROAs may have customers which announce their own prefixes and ASs into global eBGP but who do not wish to go though the work to manage the relevant certificates and ROAs. Operators SHOULD offer to provision the RPKI data for these customers just as they provision many other things for them.

While an operator using RPKI data MAY choose any polling frequency they wish for ensuring they have a fresh RPKI cache. However, if they use RPKI data as an input to operational routing decisions, they SHOULD ensure local cache freshness at least every four to six hours.

4. Within a Network

Origin validation need only be done by edge routers in a network, those which border other networks/ASs.

A validating router will use the result of origin validation to influence local policy within its network, see Section 5. In

deployment this policy should fit into the AS's existing policy, preferences, etc. This allows a network to incrementally deploy validation-capable border routers.

eBGP speakers which face more critical peers or up/down-streams are candidates for the earliest deployment. Validating more critical received announcements should be considered in partial deployment.

5. Routing Policy

Origin validation based on the RPKI marks a received announcement as having an origin which is Valid, NotFound, or Invalid. See [I-D.ietf-sidr-pfx-validate]. How this is used in routing SHOULD be specified by the operator's local policy.

Local policy using relative preference is suggested to manage the uncertainty associated with a system in early deployment, applying local policy to eliminate the threat of unroutability of prefixes due to ill-advised certification policies and/or incorrect certification data. E.g. until the community feels comfortable relying on RPKI data, routing on Invalid origin validity, though at a low preference, MAY occur.

As origin validation will be rolled out incrementally, coverage will be incomplete for a long time. Therefore, routing on NotFound validity state SHOULD be done for a long time. As the transition moves forward, the number of BGP announcements with validation state NotFound should decrease. Hence an operator's policy SHOULD NOT be overly strict, preferring Valid announcements, attaching a lower preference to, but still using, NotFound announcements, and dropping or giving very low preference to Invalid announcements.

Some providers may choose to set Local-Preference based on the RPKI validation result. Other providers may not want the RPKI validation result to be more important than AS-path length -- these providers would need to map RPKI validation result to some BGP attribute that is evaluated in BGP's path selection process after AS-path is evaluated. Routers implementing RPKI-based origin validation MUST provide such options to operators.

When using a metric which is also influenced by other local policy, an operator should be careful not to create privilege upgrade vulnerabilities. E.g. if Local Pref is set depending on validity state, be careful that peer community signaling MAY NOT upgrade an Invalid announcement to Valid or better.

Announcements with Valid origins SHOULD be preferred over those with

NotFound or Invalid origins, if the latter are accepted at all.

Announcements with NotFound origins SHOULD be preferred over those with Invalid origins.

Announcements with Invalid origins SHOULD NOT be used, but MAY be used to meet special operational needs. In such circumstances, the announcement SHOULD have a lower preference than that given to Valid or NotFound.

Validity state signialing SHOULD NOT be accepted from a neighbor AS. The validity state of a received announcement has only local scope due to issues such as scope of trust, RPKI synchrony, and [I-D.ietf-sidr-ltamgmt].

6. Notes

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no 'fix' for this, it is the nature of distributed data with distributed caches.

There is some uncertainty about the origin AS of aggregates and what, if any, ROA can be used. The long range solution to this is the deprecation of AS-SETs, see [I-D.wkumari-deprecate-as-sets].

Operators who manage certificates SHOULD associate RPKI Ghostbusters Records (see [I-D.ietf-sidr-ghostbusters]) with each publication point they control. These are publication points holding the CRL, ROAs, and other signed objects issued by the operator, and made available to other ASs in support of routing on the public Internet.

7. Security Considerations

As the BGP origin AS of an update is not signed, origin validation is open to malicious spoofing. Therefore, RPKI-based origin validation is designed to deal only with inadvertent mis-advertisement.

Origin validation does not address the problem of AS-Path validation. Therefore paths are open to manipulation, either malicious or accidental.

As BGP does not ensure that traffic will flow via the paths it advertises, the data plane may not follow the control plane.

Be aware of the class of privilege escalation issues discussed in Section 5 above.

8. IANA Considerations

This document has no IANA Considerations.

9. Acknowledgments

The author wishes to thank Rob Austein, Steve Bellovin, Jay Borkenhagen, Steve Kent, Pradosh Mohapatra, Chris Morrow, Sandy Murphy, Keyur Patel, Heather and Jason Schiller, John Scudder, Kotikalapudi Sriram, Maureen Stillman, and Dave Ward.

10. References

10.1. Normative References

[I-D.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch-13 (work in progress), May 2011.

[I-D.ietf-sidr-ghostbusters]
Bush, R., "The RPKI Ghostbusters Record", draft-ietf-sidr-ghostbusters-15 (work in progress), October 2011.

[I-D.ietf-sidr-ltamgmt]
Reynolds, M. and S. Kent, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-02 (work in progress), June 2011.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-02 (work in progress), July 2011.

[I-D.ietf-sidr-repos-struct]
Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", draft-ietf-sidr-repos-struct-09 (work in progress), July 2011.

[I-D.ietf-sidr-roa-format]
Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", draft-ietf-sidr-roa-format-12 (work in progress), May 2011.

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-18 (work in progress), October 2011.

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

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, February 2010.

10.2. Informative References

[I-D.wkumari-deprecate-as-sets]
Kumari, W., "Deprecation of BGP AS_SET, AS_CONFED_SET.", draft-wkumari-deprecate-as-sets-01 (work in progress), September 2010.

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

[rcynic] "rcynic read-me", <http://subvert-rpki.hactrn.net/rcynic/README>.

Author's Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Network Working Group
Internet-Draft
Intended status: BCP
Expires: May 17, 2012

R. Bush
Internet Initiative Japan
November 14, 2011

RPKI-Based Origin Validation Operation
draft-ietf-sidr-origin-ops-13

Abstract

Deployment of RPKI-based BGP origin validation has many operational considerations. This document attempts to collect and present them. It is expected to evolve as RPKI-based origin validation is deployed and the dynamics are better understood.

Requirements Language

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 [RFC2119].

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 May 17, 2012.

Copyright Notice

Table of Contents

1. Introduction

RPKI-based origin validation relies on widespread deployment of the Resource Public Key Infrastructure (RPKI) [I-D.ietf-sidr-arch]. How the RPKI is distributed and maintained globally is a serious concern from many aspects.

The global RPKI is in very initial stages of deployment, there is no single root trust anchor, initial testing is being done by the IANA and the RIRs, and there is a technical testbed. It is thought that origin validation based on the RPKI will be deployed incrementally over the next year to five years.

Origin validation needs to be done only by an AS's border routers and is designed so that it can be used to protect announcements which are originated by any network participating in Internet BGP routing: large providers, upstreams and down-streams, and by small stub/ enterprise/edge routers.

Origin validation has been designed to be deployed on current routers without significant hardware upgrade. It should be used in border routers by operators from large backbones to small stub/entetprise/ edge networks.

RPKI-based origin validation has been designed so that, with prudent local routing policies, there is little risk that what is seen as today's normal Internet routing is threatened by imprudent deployment of the global RPKI, see Section 5.

2. Suggested Reading

It is assumed that the reader understands BGP, [RFC4271], the RPKI, see [I-D.ietf-sidr-arch], the RPKI Repository Structure, see [I-D.ietf-sidr-repos-struct], ROAs, see [I-D.ietf-sidr-roa-format], the RPKI to Router Protocol, see [I-D.ietf-sidr-rpki-rtr], RPKI-based Prefix Validation, see [I-D.ietf-sidr-pfx-validate], and Ghostbusters Records, see [I-D.ietf-sidr-ghostbusters].

3. RPKI Distribution and Maintenance

The RPKI is a distributed database containing certificates, CRLs, manifests, ROAs, and Ghostbusters Records as described in [I-D.ietf-sidr-repos-struct]. Policies and considerations for RPKI object generation and maintenance are discussed elsewhere.

A local valid cache containing all RPKI data may be gathered from the

global distributed database using the rsync protocol, [RFC5781], and a validation tool such as rcynic [rcynic].

Validated caches may also be created and maintained from other validated caches. Network operators SHOULD take maximum advantage of this feature to minimize load on the global distributed RPKI database. Of course, the recipient SHOULD re-validate the data.

Timing of inter-cache synchronization is outside the scope of this document, but depends on things such as how often routers feed from the caches, how often the operator feels the global RPKI changes significantly, etc.

As inter-cache synchronization within an operator does not impact global RPKI resources, an operator MAY choose to synchronize quite frequently.

As RPKI-based origin validation relies on the availability of RPKI data, operators SHOULD locate caches close to routers that require these data and services. 'Close' is, of course, complex. One should consider trust boundaries, routing bootstrap reachability, latency, etc.

For redundancy, a router SHOULD peer with more than one cache at the same time. Peering with two or more, at least one local and others remote, is recommended.

If an operator trusts upstreams to carry their traffic, they MAY also trust the RPKI data those upstreams cache, and SHOULD peer with caches made available to them by those upstreams. Note that this places an obligation on those upstreams to maintain fresh and reliable caches, and to make them available to their customers. And, as usual, the recipient SHOULD re-validate the data.

A transit provider or a network with peers SHOULD validate origins in announcements made by upstreams, down-streams, and peers. They still SHOULD trust the caches provided by their upstreams.

Before issuing a ROA for a super-block, an operator MUST ensure that any sub-allocations from that block which are announced by other ASs, e.g. customers, have correct ROAs in the RPKI. Otherwise, issuing a ROA for the super-block will cause the announcements of sub-allocations with no ROAs to be viewed as Invalid, see [I-D.ietf-sidr-pfx-validate].

Use of RPKI-based origin validation removes any need to originate more specifics into BGP to protect against mis-origination of a less specific prefix. Having a ROA for the covering prefix should protect

it.

To aid translation of ROAs into efficient search algorithms in routers, ROAs SHOULD be as precise as possible, i.e. match prefixes as announced in BGP. E.g. software and operators SHOULD avoid use of excessive max length values in ROAs unless operationally necessary.

One advantage of minimal ROA length is that the forged origin attack does not work for sub-prefixes that are not covered by overly long max length. E.g. if, instead of 10.0.0.0/16-24, one issues 10.0.0.0/16 and 10.0.42.0/24, a forged origin attack can not succeed against 10.0.66.0/24. They must attack the whole /16, which is more likely to be noticed because of its size.

Therefore, ROA generation software MUST use the prefix length as the max length if the user does not specify a max length.

Operators SHOULD be conservative in use of max length in ROAs. E.g., if a prefix will have only a few sub-prefixes announced, multiple ROAs for the specific announcements SHOULD be used as opposed to one ROA with a long max length.

If a prefix is legitimately announced by more than one AS, ROAs for all of the ASs SHOULD be issued so that all are considered Valid.

An environment where private address space is announced in eBGP the operator MAY have private RPKI objects which cover these private spaces. This will require a trust anchor created and owned by that environment, see [I-D.ietf-sidr-ltamgmt].

Operators owning prefix P should issue ROAs for all ASs which may announce P.

Operators issuing ROAs may have customers which announce their own prefixes and ASs into global eBGP but who do not wish to go though the work to manage the relevant certificates and ROAs. Operators SHOULD offer to provision the RPKI data for these customers just as they provision many other things for them.

While an operator using RPKI data MAY choose any polling frequency they wish for ensuring they have a fresh RPKI cache. However, if they use RPKI data as an input to operational routing decisions, they SHOULD ensure local cache freshness at least every four to six hours.

4. Within a Network

Origin validation need only be done by edge routers in a network,

those which border other networks/ASs.

A validating router will use the result of origin validation to influence local policy within its network, see Section 5. In deployment this policy should fit into the AS's existing policy, preferences, etc. This allows a network to incrementally deploy validation-capable border routers.

eBGP speakers which face more critical peers or up/down-streams are candidates for the earliest deployment. Validating more critical received announcements should be considered in partial deployment.

5. Routing Policy

Origin validation based on the RPKI marks a received announcement as having an origin which is Valid, NotFound, or Invalid. See [I-D.ietf-sidr-pfx-validate]. How this is used in routing SHOULD be specified by the operator's local policy.

Local policy using relative preference is suggested to manage the uncertainty associated with a system in early deployment, applying local policy to eliminate the threat of unroutability of prefixes due to ill-advised certification policies and/or incorrect certification data. E.g. until the community feels comfortable relying on RPKI data, routing on Invalid origin validity, though at a low preference, MAY occur.

As origin validation will be rolled out incrementally, coverage will be incomplete for a long time. Therefore, routing on NotFound validity state SHOULD be done for a long time. As the transition moves forward, the number of BGP announcements with validation state NotFound should decrease. Hence an operator's policy SHOULD NOT be overly strict, preferring Valid announcements, attaching a lower preference to, but still using, NotFound announcements, and dropping or giving very low preference to Invalid announcements.

Some providers may choose to set Local-Preference based on the RPKI validation result. Other providers may not want the RPKI validation result to be more important than AS-path length -- these providers would need to map RPKI validation result to some BGP attribute that is evaluated in BGP's path selection process after AS-path is evaluated. Routers implementing RPKI-based origin validation MUST provide such options to operators.

Local-Preference may be used to carry both the validity state of a prefix along with it's traffic engineering characteristic(s). It is likely that an operator already using Local-Preference will have to

change policy so they can encode these two separate characteristics in the same BGP attribute without negatively impact or opening privilege escalation attacks.

When using a metric which is also influenced by other local policy, an operator should be careful not to create privilege upgrade vulnerabilities. E.g. if Local Pref is set depending on validity state, be careful that peer community signaling MAY NOT upgrade an Invalid announcement to Valid or better.

Announcements with Valid origins SHOULD be preferred over those with NotFound or Invalid origins, if the latter are accepted at all.

Announcements with NotFound origins SHOULD be preferred over those with Invalid origins.

Announcements with Invalid origins SHOULD NOT be used, but MAY be used to meet special operational needs. In such circumstances, the announcement SHOULD have a lower preference than that given to Valid or NotFound.

Validity state signialing SHOULD NOT be accepted from a neighbor AS. The validity state of a received announcement has only local scope due to issues such as scope of trust, RPKI synchrony, and [I-D.ietf-sidr-ltamgmt].

6. Notes

Like the DNS, the global RPKI presents only a loosely consistent view, depending on timing, updating, fetching, etc. Thus, one cache or router may have different data about a particular prefix than another cache or router. There is no 'fix' for this, it is the nature of distributed data with distributed caches.

It is hoped that testing and deployment will produce advice on relying party cache loading and timing.

There is some uncertainty about the origin AS of aggregates and what, if any, ROA can be used. The long range solution to this is the deprecation of AS-SETs, see [I-D.wkumari-deprecate-as-sets].

Operators who manage certificates SHOULD associate RPKI Ghostbusters Records (see [I-D.ietf-sidr-ghostbusters]) with each publication point they control. These are publication points holding the CRL, ROAs, and other signed objects issued by the operator, and made available to other ASs in support of routing on the public Internet.

7. Security Considerations

As the BGP origin AS of an update is not signed, origin validation is open to malicious spoofing. Therefore, RPKI-based origin validation is designed to deal only with inadvertent mis-advertisement.

Origin validation does not address the problem of AS-Path validation. Therefore paths are open to manipulation, either malicious or accidental.

As BGP does not ensure that traffic will flow via the paths it advertises, the data plane may not follow the control plane.

Be aware of the class of privilege escalation issues discussed in Section 5 above.

8. IANA Considerations

This document has no IANA Considerations.

9. Acknowledgments

The author wishes to thank Shane Amante, Rob Austein, Steve Bellovin, Jay Borkenhagen, Steve Kent, Pradosh Mohapatra, Chris Morrow, Sandy Murphy, Keyur Patel, Heather and Jason Schiller, John Scudder, Kotikalapudi Sriram, Maureen Stillman, and Dave Ward.

10. References

10.1. Normative References

[I-D.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch-13 (work in progress), May 2011.

[I-D.ietf-sidr-ghostbusters]
Bush, R., "The RPKI Ghostbusters Record", draft-ietf-sidr-ghostbusters-15 (work in progress), October 2011.

[I-D.ietf-sidr-ltamgmt]
Reynolds, M. and S. Kent, "Local Trust Anchor Management for the Resource Public Key Infrastructure", draft-ietf-sidr-ltamgmt-02 (work in progress), June 2011.

[I-D.ietf-sidr-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-03 (work in progress), October 2011.

[I-D.ietf-sidr-repos-struct]
Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", draft-ietf-sidr-repos-struct-09 (work in progress), July 2011.

[I-D.ietf-sidr-roa-format]
Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", draft-ietf-sidr-roa-format-12 (work in progress), May 2011.

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-19 (work in progress), October 2011.

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

[RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, February 2010.

10.2. Informative References

[I-D.wkumari-deprecate-as-sets]
Kumari, W., "Deprecation of BGP AS_SET, AS_CONFED_SET.", draft-wkumari-deprecate-as-sets-01 (work in progress), September 2010.

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

[rcynic] "rcynic read-me", <http://subvert-rpki.hactrn.net/rcynic/README>.

Author's Address

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

P. Mohapatra, Ed.
Cisco Systems
J. Scudder, Ed.
D. Ward, Ed.
Juniper Networks
R. Bush, Ed.
Internet Initiative Japan, Inc.
R. Austein, Ed.
Internet Systems Consortium

October 31, 2011

BGP Prefix Origin Validation
draft-ietf-sidr-pfx-validate-03

Abstract

To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

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 May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

A BGP route associates an address prefix with a set of autonomous systems (AS) that identify the interdomain path the prefix has traversed in the form of BGP announcements. This set is represented as the AS_PATH attribute in BGP [RFC4271] and starts with the AS that originated the prefix. To help reduce well-known threats against BGP including prefix mis-announcing and monkey-in-the-middle attacks, one of the security requirements is the ability to validate the origination AS of BGP routes. More specifically, one needs to validate that the AS number claiming to originate an address prefix (as derived from the AS_PATH attribute of the BGP route) is in fact authorized by the prefix holder to do so. This document describes a simple validation mechanism to partially satisfy this requirement.

The Resource Public Key Infrastructure (RPKI) describes an approach to build a formally verifyable database of IP addresses and AS numbers as resources. The overall architecture of RPKI as defined in [I-D.ietf-sidr-arch] consists of three main components:

- A public key infrastructure (PKI) with the necessary certificate objects,
- Digitally signed routing objects,
- A distributed repository system to hold the objects that would also support periodic retrieval.

The RPKI system is based on resource certificates that define extensions to X.509 to represent IP addresses and AS identifiers [RFC3779], thus the name RPKI. Route Origin Authorizations (ROA) [I-D.ietf-sidr-roa-format] are separate digitally signed objects that define associations between ASes and IP address blocks. Finally the repository system is operated in a distributed fashion through the IANA, RIR hierarchy, and ISPs.

In order to benefit from the RPKI system, it is envisioned that relying parties either at AS or organization level obtain a local copy of the signed object collection, verify the signatures, and process them. The cache must also be refreshed periodically. The exact access mechanism used to retrieve the local cache is beyond the scope of this document.

Individual BGP speakers can utilize the processed data contained in the local cache to validate BGP announcements. The protocol details to retrieve the processed data from the local cache to the BGP speakers is beyond the scope of this document (refer to [I-D.ietf-sidr-rpki-rtr] for such a mechanism). This document

proposes a means by which a BGP speaker can make use of the processed data in order to assign a "validity state" to each prefix in a received BGP UPDATE message.

Note that the complete path attestation against the AS_PATH attribute of a route is outside the scope of this document.

Although RPKI provides the context for this draft, it is equally possible to use any other database which is able to map prefixes to their authorized origin ASes. Each distinct database will have its own particular operational and security characteristics; such characteristics are beyond the scope of this document.

1.1. Requirements Language

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 [RFC2119].

2. Prefix-to-AS Mapping Database

The BGP speaker loads validated objects from the cache into local storage. The objects loaded have the content (IP address, prefix length, maximum length, origin AS number). We refer to such a locally stored object colloquially as a "ROA" in the discussion below although we note that this is not a strictly accurate use of the term.

We define several terms in addition to "ROA". Where these terms are used, they are capitalized:

- Prefix: (IP address, prefix length), interpreted as is customary (see [RFC4632]).
- Route: Data derived from a received BGP UPDATE, as defined in [RFC4271], Section 1.1. The Route includes one Prefix and an AS_PATH, among other things.
- ROA Prefix: The Prefix from a ROA.
- ROA ASN: The origin ASN from a ROA.
- Route Prefix: A Prefix derived from a route.
- Route Origin ASN: The origin AS number derived from a Route. The origin AS number is the rightmost AS in the final segment of the AS_PATH attribute in the Route if that segment is of type

AS_SEQUENCE, or NONE if the final segment of the AS_PATH attribute is of any type other than AS_SEQUENCE. No ROA can match an origin AS number of "NONE". No Route can match a ROA whose origin AS number is zero.

- Covered: A Route Prefix is said to be Covered by a ROA when the ROA prefix length is less than or equal to the Route prefix length and the ROA prefix address matches the Route prefix address for all bits specified by the ROA prefix length. (This is simply a statement of the well-known concept of determining a prefix match.)

- Matched: A Route Prefix is said to be Matched by a ROA when the Route Prefix is Covered by that ROA and in addition, the Route prefix length is less than or equal to the ROA maximum length and the Route Origin ASN is equal to the ROA ASN, keeping in mind that a ROA ASN of zero can never be matched, nor can a route origin AS number of "NONE".

Given these definitions, any given BGP Route learned from an EBGP peer will be found to have one of the following "validation states":

- Not found: No ROA Covers the Route Prefix.

- Valid: At least one ROA Matches the Route Prefix.

- Invalid: At least one ROA Covers the Route Prefix, but no ROA Matches it.

When a BGP speaker receives an UPDATE from one of its EBGP peers, it SHOULD perform a lookup as described above for each of the Routes in the UPDATE message. The "validation state" of the Route SHOULD be set to reflect the result of the lookup. Note that the validation state of the Route does not determine whether the Route is stored in the local BGP speaker’s Adj-RIB-In. This procedure SHOULD NOT be performed for Routes learned from peers of types other than EBGP. (Any of these MAY be overridden by configuration.)

Use of the validation state is discussed in Section 3 and Section 5.

We observe that a Route can be Matched or Covered by more than one ROA. This procedure does not mandate an order in which ROAs must be visited; however, the "validation state" output is fully determined.

2.1. Pseudo-Code

The following pseudo-code illustrates the procedure above. In case of ambiguity, the procedure above, rather than the pseudo-code,

should be taken as authoritative.

```
   //Input are the variables derived from a BGP UPDATE message
   //that need to be validated.
   //
   //The input prefix is comprised of prefix.address and
   //prefix.length.
   //
   //origin_as is the rightmost AS in the final segment of the
   //AS_PATH attribute in the UPDATE message if that segment is
   //AS_SEQUENCE.  If the final segment of AS_PATH is not an
   //AS_SEQUENCE, origin_as is NONE.
   //
   //Collectively, the prefix and origin_as correspond to the
   //Route defined in the preceding section.
   input = {prefix, origin_as};

   //Initialize result to "not found" state
   result = BGP_PFXV_STATE_NOT_FOUND;

   //pfx_validate_table organizes all the ROA entries retrieved
   //from the RPKI cache based on the IP address and the prefix
   //length field. There can be multiple such entries that match
   //the input. Iterate through all of them.
   entry = next_lookup_result(pfx_validate_table, input.prefix);

   while (entry != NULL) {
       prefix_exists = TRUE;

       if (input.prefix.length <= entry->max_length) {
           if (input.origin_as != NONE
               && entry->origin_as != 0
               && input.origin_as == entry->origin_as) {
               result = BGP_PFXV_STATE_VALID;
               return (result);
           }
       }
       entry = next_lookup_result(pfx_validate_table, input.prefix);
   }

   //If pfx_validate_table contains one or more prefixes that
   //match the input, but none of them resulted in a "valid"
   //outcome since the origin_as did not match, return the
   //result state as "invalid". Else the initialized state of
   //"not found" applies to this validation operation.
   if (prefix_exists == TRUE) {
       result = BGP_PFXV_STATE_INVALID;
   }

   return (result);
```

3. Policy Control

An implementation MUST provide the ability to match and set the validation state of routes as part of its route policy filtering function. Use of validation state in route policy is elaborated in Section 5. For more details on operational policy considerations, see [I-D.ietf-sidr-origin-ops].

4. Interaction with Local Cache

Each BGP speaker supporting prefix validation as described in this document is expected to communicate with one or multiple local caches that store a database of RPKI signed objects. The protocol mechanisms used to fetch the data and store them locally at the BGP speaker is beyond the scope of this document (please refer [I-D.ietf-sidr-rpki-rtr]). Irrespective of the protocol, the prefix validation algorithm as outlined in this document is expected to function correctly in the event of failures and other timing conditions that may result in an empty and/or partial prefix-to-AS mapping database. Indeed, if the (in-PoP) cache is not available and the mapping database is empty on the BGP speaker, all the lookups will result in "not found" state and the prefixes will be advertised to rest of the network (unless restricted by policy configuration). Similarly, if BGP UPDATEs arrive at the speaker while the fetch operation from the cache is in progress, some prefix lookups will also result in "not found" state. The implementation is expected to handle these timing conditions and MUST re-validate affected prefixes once the fetch operation is complete. The same applies during any subsequent incremental updates of the validation database.

In the event that connectivity to the cache is lost, the router should make a reasonable effort to fetch a new validation database (either from the same, or a different cache), and SHOULD wait until the new validation database has been fetched before purging the previous one. A configurable timer MUST be provided to bound the length of time the router will wait before purging the previous validation database.

5. Deployment Considerations

Once a route is received from an EBGP peer it is categorized according the procedure given in Section 2. Subsequently, routing policy as discussed in Section 3 can be used to take action based on the validation state.

Policies which could be implemented include filtering routes based on

validation state (for example, rejecting all "invalid" routes) or adjusting a route's degree of preference in the selection algorithm based on its validation state. The latter could be accomplished by adjusting the value of such attributes as LOCAL_PREF. Considering invalid routes for BGP decision process is a pure local policy matter and should be done with utmost care.

In some cases (particularly when the selection algorithm is influenced by the adjustment of a route property that is not propagated into IBGP) it could be necessary for routing correctness to propagate the validation state to the IBGP peer. This can be accomplished on the sending side by setting a community or extended community based on the validation state, and on the receiving side by matching the (extended) community and setting the validation state.

6. Contributors

Rex Fernando rex@cisco.com
Keyur Patel keyupate@cisco.com
Cisco Systems

Miya Kohno mkohno@juniper.net
Juniper Networks

Shin Miyakawa miyakawa@nttv6.jp
Taka Mizuguchi
Tomoya Yoshida
NTT Communications

Russ Housley housley@vigilsec.com
Vigil Security

Junaid Israr jisra052@uottawa.ca
Mouhcine Guennoun mguennou@uottawa.ca
Hussein Mouftah mouftah@site.uottawa.ca
University of Ottawa School of Information Technology and Engineering(SITE) 800 King Edward Avenue, Ottawa, Ontario, Canada, K1N 6N5

7. Acknowledgements

Junaid Israr's contribution to this specification is part of his PhD research work and thesis at University of Ottawa, Canada. Hannes Gredler provided valuable feedback.

8. IANA Considerations

9. Security Considerations

Although this specification discusses one portion of a system to validate BGP routes, it should be noted that it relies on a database (RPKI or other) to provide validation information. As such, the security properties of that database must be considered in order to determine the security provided by the overall solution. If "invalid" routes are blocked as this specification suggests, the overall system provides a possible denial-of-service vector, for example if an attacker is able to inject one or more spoofed records into the validation database which lead a good route to be declared invalid. In addition, this system is only able to provide limited protection against a determined attacker -- the attacker need only prepend the "valid" source AS to a forged BGP route announcement in order to defeat the protection provided by this system. This mechanism does not protect against "AS in the middle attacks" or provide any path validation. It only attempts to verify the origin. In general, this system should be thought of more as a protection against misconfiguration than as true "security" in the strong sense.

10. References

10.1. Normative References

[I-D.ietf-sidr-roa-format]
Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", draft-ietf-sidr-roa-format-12 (work in progress), May 2011.

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

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

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

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

10.2. Informational References

[I-D.ietf-sidr-arch]
Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", draft-ietf-sidr-arch-13 (work in progress), May 2011.

[I-D.ietf-sidr-origin-ops]
Bush, R., "RPKI-Based Origin Validation Operation", draft-ietf-sidr-origin-ops-12 (work in progress), October 2011.

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-19 (work in progress), October 2011.

Authors' Addresses

Pradosh Mohapatra (editor)
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: pmohapat@cisco.com

John Scudder (editor)
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
USA

Email: jgs@juniper.net

David Ward (editor)
Juniper Networks
1194 N. Mathilda Ave
Sunnyvale, CA 94089
USA

Email: dward@juniper.net

Randy Bush (editor)
Internet Initiative Japan, Inc.
5147 Crystral Springs
Bainbridge Island, Washington 98110
USA

Email: randy@psg.com

Rob Austein (editor)
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
USA

Email: sra@isc.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

R. Bush
Internet Initiative Japan
B. Wijnen
RIPE NCC
K. Patel
Cisco Systems
M. Baer
SPARTA
October 31, 2011

Definitions of Managed Objects for the RPKI-Router Protocol
draft-ymbk-rpki-rtr-protocol-mib-02

Abstract

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes objects used for monitoring the RPKI Router protocol.

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 May 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects used for monitoring the RPKI Router protocol [I-D.ietf-sidr-rpki-rtr].

1.1. Requirements Language

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 [RFC2119].

2. Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of [RFC3410]. Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This document specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, [RFC2578], STD 58, [RFC2579] and STD 58, [RFC2580].

3. Overview

The objects defined in this document are used to monitor the RPKI Router protocol [I-D.ietf-sidr-rpki-rtr]. The MIB module defined in this draft is broken into these tables: the RPKI Router Cache Server (connection) Table, the RPKI Router Cache Server Errors Table, and the RPKI Router Prefix Origin Table.

The RPKI Router Cache Server Table contains information about state and current activity of connections with the RPKI Router Cache Servers. It also contains counters for the number of messages receibd and sent plus the number of accouncements, withdrawals and active records. The RPKI Router Cache Server Errors Table contains counters of occurences of errors on the connections (if any). The RPKI RoOuter Prefix Origin Table contains IP prefixes with their minumum and maximum prefix lengths and the Origin AS. This data is the collective set of information received from all RPKI Cache Servers that the router is connected with. The Cache Servers are running the RPKI Router protocol.

Two Notification have been defined to inform a Network Management Station (NMS) or operators about changes in the connection state of the connections listed in the RPKI Cache Server (Connection) Table.

4. Definitions

The Following MIB module imports definitions from [RFC2578], STD 58, [RFC2579] STD 58, [RFC2580], [RFC4001], [RFC2287]. That means we have a normative reference to those documents.

The MIB module also has a normative reference to the RPKI Router protocol [I-D.ietf-sidr-rpki-rtr]. Furthermore, for background and informative information, the MIB module refers to [RFC1982], [RFC2385], [RFC4252], [RFC5246], [RFC5925].

```
RPKI-RTR-MIB DEFINITIONS ::= BEGIN

IMPORTS

    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Integer32, Unsigned32, mib-2, Gauge32, Counter32
            FROM SNMPv2-SMI                          -- RFC2578

    InetAddressType, InetAddress, InetPortNumber,
    InetAddressPrefixLength, InetAutonomousSystemNumber
            FROM INET-ADDRESS-MIB                    -- RFC4001

    TEXTUAL-CONVENTION, TimeStamp
            FROM SNMPv2-TC                           -- RFC2579

    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
            FROM SNMPv2-CONF                         -- RFC2580

    LongUtf8String FROM SYSAPPL-MIB                  -- RFC2287

    ;

rpkiRtrMIB  MODULE-IDENTITY
    LAST-UPDATED "201110140000Z"
    ORGANIZATION "IETF Secure Inter-Domain Routing (SIDR)
                  Working Group
                 "
    CONTACT-INFO "Working Group Email: sidr@ietf.org

                  Randy Bush
```

Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington, 98110
USA
Email: randy@psg.com

Bert Wijnen
RIPE NCC
Schagen 33
3461 GL Linschoten
Netherlands
Email: bertietf@bwijnen.net

Keyur Patel
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA
Email: keyupate@cisco.com

Michael Baer
SPARTA
P.O. Box 72682
Davis, CA 95617
USA
Email: michael.baer@sparta.com
"

DESCRIPTION "This MIB module contains management objects to
support monitoring of the Resource Public Key
Infrastructure (RPKI) protocol on routers.

This version of this MIB module is part of
RFCxxxx; see the RFC itself for full legal
notices.
"

```
        REVISION     "201110140000Z"
        DESCRIPTION  "Initial version, published as RFCxxxx."
        -- Note to RFC Editor: pls fill in above (2 times) RFC
        --  number for xxxx and delete these 2 lines.
        ::= { mib-2 XXX }  -- XXX to be assigned by IANA

    rpkiRtrNotifications  OBJECT IDENTIFIER ::= { rpkiRtrMIB 0 }
    rpkiRtrObjects        OBJECT IDENTIFIER ::= { rpkiRtrMIB 1 }
    rpkiRtrConformance    OBJECT IDENTIFIER ::= { rpkiRtrMIB 2 }

    -- ================================================================
    -- Textual Conventions used in this MIB module
    -- ================================================================

    RpkiRtrConnectionType ::= TEXTUAL-CONVENTION
        STATUS      current
        DESCRIPTION "The connection type or transport security suite
                     (transport plus security mecahnism) used between
                     a router (as a client) and a cache server.

                     The following types have been defined in RFCnnnn:
    --  RFC Editor: pls fill out RFCnnnn number that will be or has
    --              been assigned to draft-ietf-sidr-rpki-rtr-nn.txt
                       ssh(1)    - sect 7.1, see also RFC4252.
                       tls(2)    - sect 7.2, see also RFC5246.
                       tcpMD5(3) - sect 7.3, see also RFC2385.
                       tcpAO(4)  - sect 7.4, see also RFC5925.
                       tcp(5)    - sect 7.
                       ipsec(6)  - sect 7, see also RFC4301.
                       other(7)  - non of the above
                    "
        REFERENCE   "The RPKI/Rtr Protocol, RFCnnnn - section 7"
    --  RFC Editor: pls fill out RFCnnnn number that will be or has been
    --              assigned to draft-ietf-sidr-rpki-rtr-nn.txt
        SYNTAX       INTEGER {
                               ssh(1),
                               tls(2),
                               tcpMD5(3),
                               tcpAO(4),
                               tcp(5),
                               ipsec(6),
                               other(7)
                     }

    -- ================================================================
    -- Scalar objects
    -- ================================================================
    rpkiRtrDiscontinuityTimer OBJECT-TYPE
```

```
        SYNTAX       TimeStamp
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "This timer represents the timestamp (value
                     of sysUpTime) at which time any of the
                     Counter32 ojects in this MIB module
                     encountered a discontinuity.

                     In principle that should only happen if the
                     SNMP agent or the instrumentation for this
                     MIB module (re-)starts."
        ::= { rpkiRtrObjects 1 }

    -- ============================================================
    -- RPKI Router Cache Server Connection Table
    -- ============================================================

    rpkiRtrCacheServerTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF RpkiRtrCacheServerTableEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "This table lists the RPKI cache servers
                     known to this router/system."
        ::= { rpkiRtrObjects 2 }

    rpkiRtrCacheServerTableEntry OBJECT-TYPE
        SYNTAX       RpkiRtrCacheServerTableEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "An entry in the rpkiRtrCacheServerTable.
                     It holds management attributes associated
                     with one connection to a RPKI cache server."
        INDEX        { rpkiRtrCacheServerAddressType,
                       rpkiRtrCacheServerRemoteAddress,
                       rpkiRtrCacheServerRemotePort
                     }
        ::= { rpkiRtrCacheServerTable 1 }

    RpkiRtrCacheServerTableEntry ::= SEQUENCE {
        rpkiRtrCacheServerAddressType          InetAddressType,
        rpkiRtrCacheServerRemoteAddress        InetAddress,
        rpkiRtrCacheServerRemotePort           InetPortNumber,
        rpkiRtrCacheServerLocalAddress         InetAddress,
        rpkiRtrCacheServerLocalPort            InetPortNumber,
        rpkiRtrCacheServerPreference           Unsigned32,
        rpkiRtrCacheServerConnectionType       RpkiRtrConnectionType,
        rpkiRtrCacheServerConnectionStatus     INTEGER,
        rpkiRtrCacheServerDescription          LongUtf8String,
```

```
        rpkiRtrCacheServerMsgsReceived         Counter32,
        rpkiRtrCacheServerMsgsSent             Counter32,
        rpkiRtrCacheServerV4ActiveRecords      Gauge32,
        rpkiRtrCacheServerV4Announcements      Counter32,
        rpkiRtrCacheServerV4Withdrawals        Counter32,
        rpkiRtrCacheServerV6ActiveRecords      Gauge32,
        rpkiRtrCacheServerV6Announcements      Counter32,
        rpkiRtrCacheServerV6Withdrawals        Counter32,
        rpkiRtrCacheServerLatestSerial         Unsigned32,
        rpkiRtrCacheServerNonce                Unsigned32,
        rpkiRtrCacheServerRefreshTimer         Unsigned32,
        rpkiRtrCacheServerTimeToRefresh        Integer32,
        rpkiRtrCacheServerId                   Unsigned32
    }

    rpkiRtrCacheServerAddressType OBJECT-TYPE
        SYNTAX       InetAddressType { ipv4(1), ipv6 (2) }
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "The network address type of the connection
                     to this RPKI cache server.

                     Only IPv4 and IPv6 are supported."
        ::= { rpkiRtrCacheServerTableEntry 1 }

    rpkiRtrCacheServerRemoteAddress OBJECT-TYPE
        SYNTAX       InetAddress (SIZE(4|16))
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "The remote network address for this connection
                     to this RPKI cache server.

                     The format of the address is defined by the
                     value of the corresponding instance of
                     rpkiRtrCacheServerAddressType."
        ::= { rpkiRtrCacheServerTableEntry 2 }

    rpkiRtrCacheServerRemotePort OBJECT-TYPE
        SYNTAX       InetPortNumber (1..65535)
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "The remote port number for this connection
                     to this RPKI cache server."
        ::= { rpkiRtrCacheServerTableEntry 3 }

    rpkiRtrCacheServerLocalAddress OBJECT-TYPE
        SYNTAX       InetAddress (SIZE(4|16))
        MAX-ACCESS   read-only
```

```
        STATUS       current
        DESCRIPTION "The local network address for this connection
                     to this RPKI cache server.

                     The format of the address is defined by the
                     value of the corresponding instance of
                     rpkiRtrCacheServerAddressType."
        ::= { rpkiRtrCacheServerTableEntry 4 }

    rpkiRtrCacheServerLocalPort OBJECT-TYPE
        SYNTAX       InetPortNumber (1..65535)
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The local port number for this connection
                     to this RPKI cache server."
        ::= { rpkiRtrCacheServerTableEntry 5 }

    rpkiRtrCacheServerPreference OBJECT-TYPE
        SYNTAX       Unsigned32 (0..255)
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The routers' preference for this
                     RPKI cache server.

                     A lower value means more preferred. If two
                     entries have the same preference, then the
                     order is arbitrary.

                     If no order is specified in the configuration
                     then this value is set to 255."
        REFERENCE    "The RPKI/Rtr Protocol, RFCnnnn - section 8."
    --  RFC-Editor: pls update RFCnnnn with the actual RFC number
    --              assigned to draft-ietf-sidr-rpki-rtr-nn.txt
        ::= { rpkiRtrCacheServerTableEntry 6 }

    rpkiRtrCacheServerConnectionType OBJECT-TYPE
        SYNTAX       RpkiRtrConnectionType
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The connection type or transport security suite
                     in use for this RPKI cache server."
        ::= { rpkiRtrCacheServerTableEntry 7 }

    rpkiRtrCacheServerConnectionStatus OBJECT-TYPE
        SYNTAX       INTEGER { up(1), down(2) }
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The connection status for this entry
```

```
                    (connection to this RPKI cache server)."
        ::= { rpkiRtrCacheServerTableEntry 8 }

    rpkiRtrCacheServerDescription OBJECT-TYPE
        SYNTAX       LongUtf8String
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "Free form description/information for this
                    connection to this RPKI cache server."
        ::= { rpkiRtrCacheServerTableEntry 9 }

    rpkiRtrCacheServerMsgsReceived OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "Number of messages received from this
                    RPKI cache server via this connection.

                    Discontinuities are indicated by the value
                    of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerTableEntry 10 }

    rpkiRtrCacheServerMsgsSent OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "Number of messages sent to this
                    RPKI cache server via this connection.

                    Discontinuities are indicated by the value
                    of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerTableEntry 11 }

    rpkiRtrCacheServerV4ActiveRecords OBJECT-TYPE
        SYNTAX       Gauge32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "Number of active IPv4 records received from
                    this RPKI cache server via this connection."
        ::= { rpkiRtrCacheServerTableEntry 12 }

    rpkiRtrCacheServerV4Announcements OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of IPv4 records announced by the
                    RPKI cache Server via this connection.
```

```
                    Discontinuities are indicated by the value
                    of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerTableEntry 13 }

    rpkiRtrCacheServerV4Withdrawals OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "The number of IPv4 records withdrawn by the
                     RPKI cache Server via this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerTableEntry 14 }

    rpkiRtrCacheServerV6ActiveRecords OBJECT-TYPE
        SYNTAX       Gauge32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "Number of active IPv6 records received from
                     this RPKI cache server via this connection."
        ::= { rpkiRtrCacheServerTableEntry 15 }

    rpkiRtrCacheServerV6Announcements OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of IPv6 records announced by the
                     RPKI cache Server via this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerTableEntry 16 }

    rpkiRtrCacheServerV6Withdrawals OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of IPv6 records withdrawn by the
                     RPKI cache Server via this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerTableEntry 17 }

    rpkiRtrCacheServerLatestSerial OBJECT-TYPE
        SYNTAX       Unsigned32
        MAX-ACCESS   read-only
```

```
        STATUS       current
        DESCRIPTION "The latest serial number of data received from
                     this RPKI server on this connection.

                     Note: this value wraps back to zero when it
                     reaches its maximum value."
        REFERENCE   "RFCnnnn section 2 and RFC1982"
    --  RFC-Editor: please fill out nnnn with the RFC number assigned
    --              to draft-ietf-sidr-rpki-rtr-nn.txt
        ::= { rpkiRtrCacheServerTableEntry 18 }

    rpkiRtrCacheServerNonce OBJECT-TYPE
        SYNTAX      Unsigned32 (0..65535)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "The nonce associated with the RPKI cache server
                     at the other end of this connection."
        REFERENCE   "RFCnnnn section 2"
        ::= { rpkiRtrCacheServerTableEntry 19 }

    rpkiRtrCacheServerRefreshTimer OBJECT-TYPE
        SYNTAX       Unsigned32 (60..7200)
        UNITS       "seconds"
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of seconds configured for the refresh
                     timer for this connection to this RPKI cache
                     server."
        ::= { rpkiRtrCacheServerTableEntry 20 }

    rpkiRtrCacheServerTimeToRefresh OBJECT-TYPE
        SYNTAX       Integer32
        UNITS       "seconds"
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of seconds remaining before a new
                     refresh is performed via a Serial Query to
                     this cache server over this connection.

                     A negative value means that the refresh time
                     has passed this many seconds and the refresh
                     has not yet been completed.

                     Upon a completed refresh (i.e. a successful
                     rnd complete esponse to a Serial Query) the
                     value of this attribute will be re-initialized
                     with the value of the corresponding
                     rpkiRtrCacheServerRefreshTimer attribute."
```

```
        ::= { rpkiRtrCacheServerTableEntry 21 }

    rpkiRtrCacheServerId OBJECT-TYPE
        SYNTAX       Unsigned32 (1..4294967295)
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The unique ID for this connection.

                     An implementation must make sure this ID is unique
                     within this table. It is this ID that can be used
                     to find entries in the rpkiRtrPrefixOriginTable
                     that were created by announcements received on this
                     connection from this cache server."
        ::= { rpkiRtrCacheServerTableEntry 22 }

    -- ================================================================
    -- Errors Table
    -- ================================================================

    rpkiRtrCacheServerErrorsTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF RpkiRtrCacheServerErrorsTableEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "This table provides statistics on errors per
                     RPKI peer connection. These can be used for
                     debuging."
        ::= { rpkiRtrObjects 3 }

    rpkiRtrCacheServerErrorsTableEntry OBJECT-TYPE
        SYNTAX       RpkiRtrCacheServerErrorsTableEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "An entry in the rpkiCacheServerErrorTable.  It holds
                     management objects associated with errors that
                     were detected for the specified connection to
                     a specific cache server."
        AUGMENTS    { rpkiRtrCacheServerTableEntry }
        ::= { rpkiRtrCacheServerErrorsTable 1 }

    RpkiRtrCacheServerErrorsTableEntry ::= SEQUENCE {
        rpkiRtrCacheServerErrorsCorruptData        Counter32,
        rpkiRtrCacheServerErrorsInternalError      Counter32,
        rpkiRtrCacheServerErrorsNoData             Counter32,
        rpkiRtrCacheServerErrorsInvalidRequest     Counter32,
        rpkiRtrCacheServerErrorsUnsupportedVersion Counter32,
        rpkiRtrCacheServerErrorsUnsupportedPdu     Counter32,
        rpkiRtrCacheServerErrorsWithdrawalUnknown  Counter32,
        rpkiRtrCacheServerErrorsDuplicateAnnounce  Counter32
```

```
    }

  rpkiRtrCacheServerErrorsCorruptData OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION "The number of 'Corrupt Data' errors received
                   from the RPKI cache server at the other end
                   of this connection.

                   Discontinuities are indicated by the value
                   of rpkiRtrDiscontinuityTimer."
      ::= { rpkiRtrCacheServerErrorsTableEntry 1 }

  rpkiRtrCacheServerErrorsInternalError OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION "The number of 'Internal Error' errors received
                   from the RPKI cache server at the other end
                   of this connection.

                   Discontinuities are indicated by the value
                   of rpkiRtrDiscontinuityTimer."
      ::= { rpkiRtrCacheServerErrorsTableEntry 2 }

  rpkiRtrCacheServerErrorsNoData OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION "The number of 'No Data Available' errors received
                   from the RPKI cache server at the other end
                   of this connection.

                   Discontinuities are indicated by the value
                   of rpkiRtrDiscontinuityTimer."
      ::= { rpkiRtrCacheServerErrorsTableEntry 3 }

  rpkiRtrCacheServerErrorsInvalidRequest OBJECT-TYPE
      SYNTAX       Counter32
      MAX-ACCESS   read-only
      STATUS       current
      DESCRIPTION "The number of 'Invalid Request' errors received
                   from the RPKI cache server at the other end
                   of this connection.

                   Discontinuities are indicated by the value
                   of rpkiRtrDiscontinuityTimer."
```

```
        ::= { rpkiRtrCacheServerErrorsTableEntry 4 }

    rpkiRtrCacheServerErrorsUnsupportedVersion OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of 'Unsupported Protocol Version'
                     errors received from the RPKI cache server at
                     the other end of this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerErrorsTableEntry 5 }

    rpkiRtrCacheServerErrorsUnsupportedPdu OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of 'Unsupported PDU Type' errors
                     received from the RPKI cache server at the
                     other end of this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerErrorsTableEntry 6 }

    rpkiRtrCacheServerErrorsWithdrawalUnknown OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of 'Withdrawal of Unknown Record'
                     errors received from the RPKI cache server at
                     the other end of this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
        ::= { rpkiRtrCacheServerErrorsTableEntry 7 }

    rpkiRtrCacheServerErrorsDuplicateAnnounce OBJECT-TYPE
        SYNTAX       Counter32
        MAX-ACCESS   read-only
        STATUS       current
        DESCRIPTION "The number of 'Duplicate Announcement Received'
                     errors received from the RPKI cache server at
                     the other end of this connection.

                     Discontinuities are indicated by the value
                     of rpkiRtrDiscontinuityTimer."
```

```
        ::= { rpkiRtrCacheServerErrorsTableEntry 8 }

    -- ============================================================
    -- The rpkiRtrPrefixOriginTable (was refered to as ROATable in an
    -- earlier version of this table)
    -- ============================================================

    rpkiRtrPrefixOriginTable OBJECT-TYPE
        SYNTAX       SEQUENCE OF RpkiRtrPrefixOriginTableEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "This table lists the prefixes that were
                     announced by RPKI cache servers to this system.
                     That is the prefixes and their Origin ASN
                     as recieved by announcements via the
                     rpki-rtr protocol."
        ::= { rpkiRtrObjects 4 }

    rpkiRtrPrefixOriginTableEntry OBJECT-TYPE
        SYNTAX       RpkiRtrPrefixOriginTableEntry
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "An entry in the rpkiRtrPrefixOriginTable.
                     This represents one announced prefix."
        INDEX       { rpkiRtrPrefixOriginAddressType,
                      rpkiRtrPrefixOriginAddress,
                      rpkiRtrPrefixOriginMinLength
                    }
        ::= { rpkiRtrPrefixOriginTable 1 }

    RpkiRtrPrefixOriginTableEntry ::= SEQUENCE {
        rpkiRtrPrefixOriginAddressType    InetAddressType,
        rpkiRtrPrefixOriginAddress        InetAddress,
        rpkiRtrPrefixOriginMinLength      InetAddressPrefixLength,
        rpkiRtrPrefixOriginMaxLength      InetAddressPrefixLength,
        rpkiRtrPrefixOriginASN            InetAutonomousSystemNumber,
        rpkiRtrPrefixOriginCacheServerId  Unsigned32
    }

    rpkiRtrPrefixOriginAddressType OBJECT-TYPE
        SYNTAX       InetAddressType { ipv4(1), ipv6(2) }
        MAX-ACCESS   not-accessible
        STATUS       current
        DESCRIPTION "The network Address Type for this prefix.

                     Only IPv4 and IPv6 are supported."
        ::= { rpkiRtrPrefixOriginTableEntry 1 }
```

```
   rpkiRtrPrefixOriginAddress OBJECT-TYPE
       SYNTAX       InetAddress (SIZE(4|16))
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION "The network Address for this prefix.

                    The format of the address is defined by the
                    value of the corresponding instance of
                    rpkiRtrCacheServerAddressType."
       ::= { rpkiRtrPrefixOriginTableEntry 2 }

   rpkiRtrPrefixOriginMinLength OBJECT-TYPE
       SYNTAX       InetAddressPrefixLength
       MAX-ACCESS   not-accessible
       STATUS       current
       DESCRIPTION "The minimum prefix length allowed for this prefix."
       ::= { rpkiRtrPrefixOriginTableEntry 3 }

   rpkiRtrPrefixOriginMaxLength OBJECT-TYPE
       SYNTAX       InetAddressPrefixLength
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION "The maximum prefix length allowed for this prefix.

                    Note, this value must be greater or equal to the
                    value of rpkiRtrPrefixOriginMinLength."
       ::= { rpkiRtrPrefixOriginTableEntry 4 }

   rpkiRtrPrefixOriginASN OBJECT-TYPE
       SYNTAX       InetAutonomousSystemNumber
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION "The ASN that is authorized to announce the
                    prefix or sub-prefixes covered by this entry."
       ::= { rpkiRtrPrefixOriginTableEntry 5 }

   rpkiRtrPrefixOriginCacheServerId OBJECT-TYPE
       SYNTAX       Unsigned32 (1..4294967295)
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION "The unique ID of the connection to the cache
                    server from which this announcement was received.
                    That connection is identified/found by a matching
                    value in attribute rpkiRtrCacheServerId."
       ::= { rpkiRtrPrefixOriginTableEntry 6 }

   -- ============================================================
   -- Notifications
```

```
   -- ============================================================

   rpkiRtrCacheServerConnectionStateChange NOTIFICATION-TYPE
       OBJECTS     { rpkiRtrCacheServerConnectionStatus,
                     rpkiRtrCacheServerLatestSerial,
                     rpkiRtrCacheServerNonce
                   }
       STATUS      current
       DESCRIPTION "This notification signals a change in the status
                    of an rpkiRtrCacheServerConnection.

                    The SNMP agent MUST throttle the generation of
                    consecutive rpkiRtrCacheServerConnectionStateChange
                    notifications such that there is at least a
                    5 second gap between them.
                   "
       ::= { rpkiRtrNotifications 1 }

   rpkiRtrCacheServerConnectionToGoStale NOTIFICATION-TYPE
       OBJECTS     { rpkiRtrCacheServerV4ActiveRecords,
                     rpkiRtrCacheServerV6ActiveRecords,
                     rpkiRtrCacheServerLatestSerial,
                     rpkiRtrCacheServerNonce,
                     rpkiRtrCacheServerRefreshTimer,
                     rpkiRtrCacheServerTimeToRefresh
                   }
       STATUS      current
       DESCRIPTION "This notification signals that an RPKI cache
                    server connection is about to go stale.
                    It is suggested that this notifiation is
                    generated when the value of the
                    rpkiRtrCacheServerTimeToRefresh attribute
                    goes below 60 seconds.

                    The SNMP agent MUST throttle the generation of
                    consecutive rpkiRtrCacheServerConnectionToGoStale
                    notifications such that there is at least a
                    5 second gap between them.
                   "
       ::= { rpkiRtrNotifications 2 }

   -- ============================================================
   -- Module Compliance information
   -- ============================================================

   rpkiRtrCompliances OBJECT IDENTIFIER ::=
                                          {rpkiRtrConformance 1}
   rpkiRtrGroups      OBJECT IDENTIFIER ::=
```

```
                                          {rpkiRtrConformance 2}

   rpkiRtrReadOnlyCompliance MODULE-COMPLIANCE
       STATUS       current
       DESCRIPTION "The comliance statement for the rpkiRtrMIB
                    module. There are only read-only objects in this
                    MIB module, so the 'ReadOnly' in the name of this
                    compliance statement is there only for clarity
                    and truth in advertising.
                   "
       MODULE      -- This module
       MANDATORY-GROUPS { rpkiRtrCacheServerGroup,
                          rpkiRtrPrefixOriginGroup,
                          rpkiRtrNotificationsGroup
                        }
       GROUP       rpkiRtrCacheServerErrorsGroup
       DESCRIPTION "Implemntation of this group is optional and
                    would be useful for debugging."
       ::= { rpkiRtrCompliances 1 }

   rpkiRtrCacheServerGroup OBJECT-GROUP
       OBJECTS     { rpkiRtrDiscontinuityTimer,
                     rpkiRtrCacheServerLocalAddress,
                     rpkiRtrCacheServerLocalPort,
                     rpkiRtrCacheServerPreference,
                     rpkiRtrCacheServerConnectionType,
                     rpkiRtrCacheServerConnectionStatus,
                     rpkiRtrCacheServerDescription,
                     rpkiRtrCacheServerMsgsReceived,
                     rpkiRtrCacheServerMsgsSent,
                     rpkiRtrCacheServerV4ActiveRecords,
                     rpkiRtrCacheServerV4Announcements,
                     rpkiRtrCacheServerV4Withdrawals,
                     rpkiRtrCacheServerV6ActiveRecords,
                     rpkiRtrCacheServerV6Announcements,
                     rpkiRtrCacheServerV6Withdrawals,
                     rpkiRtrCacheServerLatestSerial,
                     rpkiRtrCacheServerNonce,
                     rpkiRtrCacheServerRefreshTimer,
                     rpkiRtrCacheServerTimeToRefresh,
                     rpkiRtrCacheServerId
                   }
       STATUS      current
       DESCRIPTION "The collection of objects to monitor the RPKI peer
                    connections."
       ::= { rpkiRtrGroups 1 }

   rpkiRtrCacheServerErrorsGroup OBJECT-GROUP
```

```
        OBJECTS     { rpkiRtrCacheServerErrorsCorruptData,
                      rpkiRtrCacheServerErrorsInternalError,
                      rpkiRtrCacheServerErrorsNoData,
                      rpkiRtrCacheServerErrorsInvalidRequest,
                      rpkiRtrCacheServerErrorsUnsupportedVersion,
                      rpkiRtrCacheServerErrorsUnsupportedPdu,
                      rpkiRtrCacheServerErrorsWithdrawalUnknown,
                      rpkiRtrCacheServerErrorsDuplicateAnnounce
                    }
        STATUS      current
        DESCRIPTION "The collection of objects that may help in
                     debugging the communication between rpki
                     clients and cache servers."
        ::= { rpkiRtrGroups 2 }

    rpkiRtrPrefixOriginGroup OBJECT-GROUP
        OBJECTS     { rpkiRtrPrefixOriginMaxLength,
                      rpkiRtrPrefixOriginASN,
                      rpkiRtrPrefixOriginCacheServerId
                    }
        STATUS      current
        DESCRIPTION "The collection of objects that represent
                     the prefix(es) and their validated origin
                     ASes."
        ::= { rpkiRtrGroups 3 }

    rpkiRtrNotificationsGroup NOTIFICATION-GROUP
        NOTIFICATIONS { rpkiRtrCacheServerConnectionStateChange,
                        rpkiRtrCacheServerConnectionToGoStale
                      }
        STATUS      current
        DESCRIPTION "The set of notifications to alert an NMS of change
                     in connections to RPKI cache servers."
        ::= { rpkiRtrGroups 4 }

    END
```

5. IANA Considerations

The MIB module in this document will required an IANA assigned OBJECT IDENTIFIER within the SMI Numbers registry. For example, replacing XXX below:

Descriptor	OBJECT IDENTIFIER value

```
rpkiRouter            { mib-2 XXX }
```

6. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Most of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. They are vulnerable in the sense that when an intruder sees the information in this MIB module, then it might help him/her to setup a an attack on the router or cache server. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

7. References

7.1. Normative References

[I-D.ietf-sidr-rpki-rtr]
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-18 (work in progress), October 2011.

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

[RFC2287] Krupczak, C. and J. Saperia, "Definitions of System-Level Managed Objects for Applications", RFC 2287, February 1998.

[RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

7.2. Informative References

[RFC1982] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[RFC4252] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Authentication Protocol", RFC 4252, January 2006.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

Authors' Addresses

Randy Bush
Internet Initiative Japan
5147 Crystal Springs
Bainbridge Island, Washington 98110
US

Phone: +1 206 780 0431 x1
Email: randy@psg.com

Bert Wijnen
RIPE NCC
Schagen 33
3461 GL Linschoten
Netherlands

Email: bertietf@bwijnen.net

Keyur Patel
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: keyupate@cisco.com

Michael Baer
SPARTA
P.O. Box 72682
Davis, CA 95617
USA

Email: michael.baer@sparta.com