Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2012

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

July 8, 2011

Algorithm Agility Procedure for RPKI. draft-ietf-sidr-algorithm-agility-01

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 January 9, 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.

A failure to comply with this process during an algorithm transition MUST be considered as non-compliance with the RPKI Certificate Policy (CP) [I-D.ietf-sidr-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 entities not under its administrative control.

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

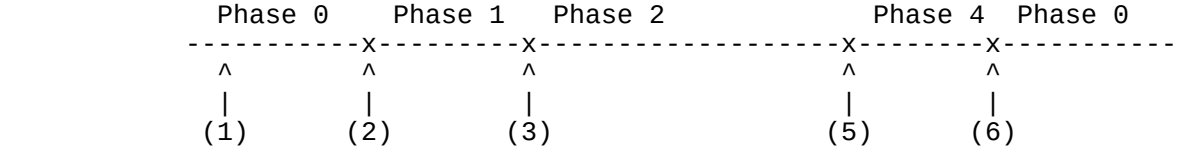
The migration process described in this document involves a series of steps that MUST be executed in chronological order by CAs and RPs. The only milestone that affects both CAs and RPs, at the same moment is the EOL date. Due to the decentralized nature of the RPKI infrastructure, it is expected that 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 certificate signed

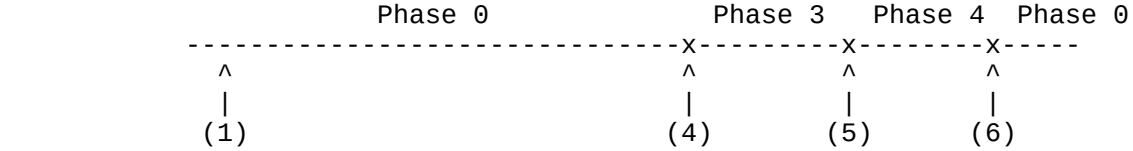
using Suite A, containing a key from Suite B. In RPKI an algorithm suite MUST NOT sign a certificate carrying a subject key that corresponds to another algorithm suite.

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, during which 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. During 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.

The following figure shows the status of the repository entries for the three example CAs during 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 during this phase, in preparation for Phase 4.

There are no changes to the CA behavior during 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 and Arturo Servin.

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-16 (work in progress), December 2010.

[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 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 12, 2011

M. Lepinski, Ed.
BBN

June 10, 2011

BGPSEC Protocol Specification
draft-ietf-sidr-bgpsec-protocol-00

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 December 12, 2011.

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

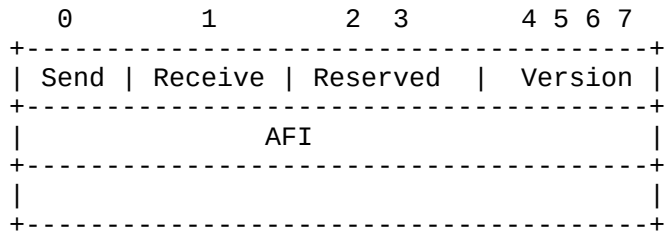
This document defines a new BGP capability [3]that allows a BGP speaker to advertise to its neighbors the ability to send and/or receive BGPSEC update messages (i.e., update messages containing the BGPSEC_Path_Signatures attribute).

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 peer, 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.

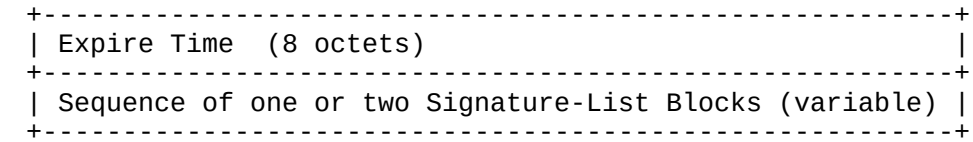
3. The BGPSEC_Path_Signatures Attribute

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

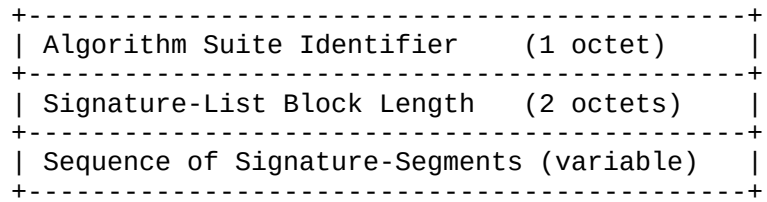


Expire Time contains a binary representation of a time as an unsigned integer number of (non-leap) seconds that have elapsed since midnight

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

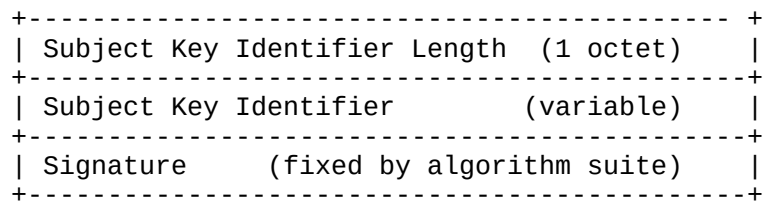


An algorithm suite consists of a digest algorithm and a signature algorithm. This version of BGPSEC only supports signature algorithms that produce a signatures of fixed length. Future registrations of algorithm suites for BGPSEC must specify the length of signatures produced by the algorithm suite. 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 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, possibly multiple occurrences of, 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 the BGPSEC speaker wishes to withdraw the route, but intermediate (malicious) BGP speakers fail to propagate the withdrawal to their peers.

It is therefore necessary for the originating BGPSEC speaker to issue a new BGPSEC update prior to reaching the Expire Time. It is RECOMMENDED that a BGPSEC speaker originate a new route advertisement for a given NLRI at intervals equal to roughly one-third the validity period of the route advertisement. (Note that it is necessary to add some small amount of random jitter to the interval to avoid synchronization effects.) For instance, if a BGPSEC speaker is originating route advertisements that are valid for one day (i.e., the Expire Time is 24 hours after the generation of the update message), then it is recommended that the BGPSEC speaker re-issue new a new BGPSEC update message for advertising the given prefix roughly once every 8 hours (plus or minus a small random value).

(Editor's Note: The parameter recommendations in the previous paragraph are preliminary and will need to be updated based on further implementation and deployment experience.)

There is a natural trade-off in setting the Expire Time. Setting a later Expire Time increases the amount of time by which a malicious intermediate can delay a future route withdrawal. Similarly, setting a later Expire Time also increases the window of opportunity for malicious replay attacks in which a previous BGPSEC announcement is replayed while suppressing a more recent withdrawal for the same prefix. However, setting a sooner Expire Timed increases the frequency with which the BGPSEC speaker needs to send new announcements for the given prefix.

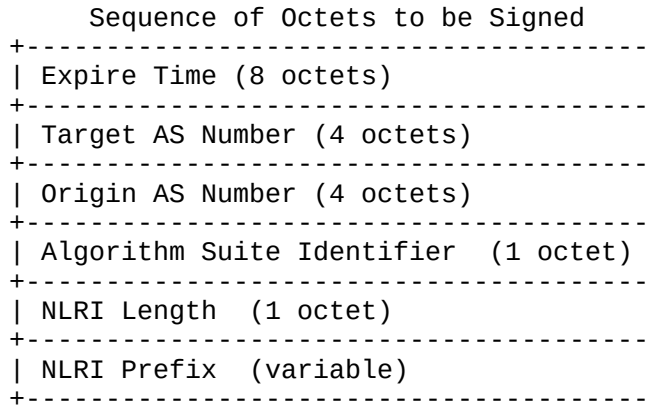
When originating a new route advertisement, each Signature-List Block MUST consist of a single Signature-Segment. The following describes how the BGPSEC speaker populates the fields of the Signature-List Block (see Section 3 for more information on the syntax of Signature-List Blocks).

The 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, 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. (See Section 7 for

discussion of the security ramifications of removing 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).

The BGPSEC speaker next removes from the BGPSEC_Path_Signatures attribute any Signature-List Blocks corresponding to algorithm suites that it does not support. The BGPSEC_Path_Signatures attribute for the new update message SHOULD contain a Signature-List Block for every algorithm suite that is both present in the received update message and which is supported by the BGPSEC speaker.

Note 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 Signature-List Blocks which are deemed 'Not Good' (e.g., contain 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 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

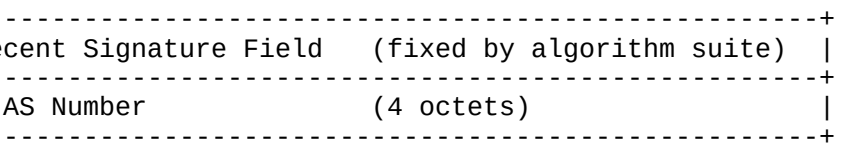
the signature.

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 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.

5. Validating a 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.

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.

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 which is not local to a single Signature-List Block, or if the AS-Path attribute contains an AS-Set segment, then the recipient should log that an error occurred, strip off the BGPSEC_Path_Signatures attribute and process the update message as though it arrived without a BGPSEC_Path_Signatures attribute.

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 MUST be discarded. If all Signature-List Blocks are discarded in this manner then the BGPSEC speaker MUST treat the

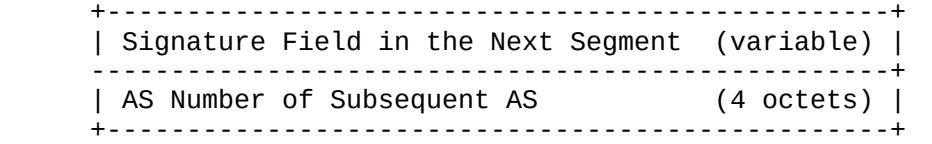
update message as though it arrived without a BGPSEC_Path_Signatures attribute.

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.

- o (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.

- o (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).

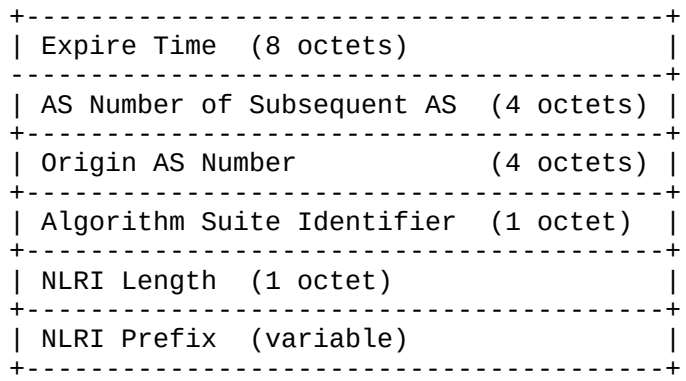
For the first segment to be processed (the most recently added segment), the 'AS Number of Subsequent AS' is the AS number of the BGPSEC speaker validating the update message. Note that if a BGPSEC speaker uses multiple AS Numbers (e.g., the BGPSEC speaker is a member of a confederation), the AS number 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 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.)

o (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:

- A new type of signature algorithm that produces signatures of variable length
- 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)
- 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:

- 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.

- 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.

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
Internet Systems Consortium
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
weiler@watson.org
Cobham

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", February 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: January 12, 2012

S. Weiler
A. Sonalker
SPARTA, Inc.
R. Austein
ISC

July 11, 2011

A Publication Protocol for the Resource Public Key Infrastructure (RPKI)
draft-ietf-sidr-publication-01

Abstract

This document defines a protocol for publishing Resource Public Key Infrastructure (RPKI) objects. Even though the RPKI will have many participants issuing certificates and creating other objects, it is operationally useful to consolidate the publication of those objects. This document provides the protocol for doing so.

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 January 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document assumes a working knowledge of the Resource Public Key Infrastructure (RPKI), which is intended to support improved routing security on the Internet. [I-D.ietf-sidr-arch]

In order to make participation in the RPKI easier, it is helpful to have a few consolidated repositories for RPKI objects, thus saving every participant from the cost of maintaining a new service. Similarly, relying parties using the RPKI objects will find it faster and more reliable to retrieve the necessary set from a smaller number of repositories.

These consolidated RPKI object repositories will in many cases be outside the administrative scope of the organization issuing a given RPKI object. Hence the need for a protocol to publish RPKI objects.

This document defines the RPKI publication protocol, including a sub-protocol for configuring the publication engine.

1.1. Terminology

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].

"Publication engine" and "publication server" are used interchangeably to refer to the server providing the service described in this document.

"Business Public Key Infrastructure" ("Business PKI" or "BPKI") refers to a PKI, separate from the RPKI, used to authenticate clients to the publication engine.

2. Context

This protocol was designed specifically for the case where an internet registry, already issuing RPKI certificates to its children, also wishes to run a publication service for its children.

We use the term "Business PKI" here because an internet registry might already have a PKI, separate from the RPKI, for authenticating its clients and might wish to reuse that PKI for this protocol. Such reuse is not a requirement.

3. Protocol Specification

In summary, the publication protocol uses XML messages wrapped in CMS, carried over HTTP transport.

The publication procotol consists of two separate subprotocols. The first is a control protocol used to configure a publication engine. The second subprotocol, which we refer to by the overloaded term "publication protocol", is used to request publication of specific objects. The publication engine operates a single HTTP server on a single port. It distinguishes between the two protocols by using different URLs for them.

3.1. Common Details

This section discusses details that the two subprotocols have in common, including the transport and CMS wrappers.

Both protocols use a simple request/response interaction. The client passes a request to the server, and the server generates a corresponding response.

A message exchange commences with the client initiating an HTTP POST with content type of "application/rpki-publication", with the message object as the body. The server’s response will similarly be the body of the response with a content type of "application/ rpki-publication".

The content of the POST and the server’s response will be a well-formed Cryptographic Message Syntax (CMS) [RFC5652] object with OID = 1.2.840.113549.1.7.2 as described in Section 3.1 of [I-D.ietf-sidr-rescerts-provisioning].

3.1.1. Common XML Message Format

The XML schema for this protocol (including both subprotocols) is below in Section 3.5. Both subprotocols use the same basic XML message format, which looks like:

```
<?xml version='1.0' encoding='us-ascii'?>
<msg xmlns="http://www.hactrn.net/uris/rpki/publication-spec/"
   version="2"
   type="message type">
   [one or more PDUs]
</msg>
```

version:
The value of this attribute is the version of this protocol. This document describes version 2.

type:
The possible values of this attribute are "reply" and "query".

A query PDU may be one of four types: config_query, client_query, publish_query, or withdraw_query. The first two are used by the control sub-protocol, the latter two by the publication sub-protocol.

A reply PDU may be one of five types: config_reply, client_reply, publish_reply, withdraw_reply, or report_error_reply.

Each of these PDUs may include an optional tag to facilitate bulk operation. If a tag is set in a query PDU, the corresponding reply(s) MUST have the tag attribute set to the same value.

3.2. Control Sub-Protocol

The control sub-protocol is used to configure a publication server. It can set global variables (at the moment, limited to a BPKI CRL) and manage clients who are allowed to publish data on the server.

3.2.1. Config Object

The <config/> object allows configuration of data that apply to the entire publication server rather than a particular client. There is exactly one <config/> object in the publication server, and it only supports the "set" and "get" actions -- it cannot be created or destroyed. Its use is typically restricted to the repository operator.

The <config/> object only has one data element that can be set: the bpki_crl. This is used by the publication server when authenticating clients.

3.2.2. Client Object

Unlike the <config/> object, the <client/> object represents one client authorized to use the publication server. There may well be

more than one <client/> object on each publication server. Again, its use is typically restricted to the respository operator.

The <client/> object supports five actions: "create", "set", "get", "list", and "destroy". Each client has a "client_handle" attribute, which is used in responses and must be specified in "create", "set", "get", or "destroy" actions.

Payload data which can be configured in a <client/> object include:

- base_uri (attribute): This attribute represents the base URI below which the client will be allowed to publish data. Additional constraints may be imposed by the publication server in certain cases, for e.g., a child publishing directly under its parent.

- bpki_cert (element): This represents the X.509 BPKI CA certificate for this client. This should be used as part of the certificate chain when validating incoming CMS messages. Two valid approaches exist. If the optional bpki_glue certificate is being used, then the bpki_cert certificate should be issued by the bpki_glue certificate; otherwise, the bpki_cert certificate should be issued by the publication engine's bpki_ta certificate.

- bpki_glue (element): This is an additional (optional) type of X.509 certificate for this client. It may be used in certain pathological cross-certification cases which require a two-certificate chain due to issuer name conflicts. When being used, issuing order is that the bpki_glue certificate should be the issuer of the bpki_cert certificate. Otherwise, it should be issued by the publication engine's bpki_ta certificate. Since this is an optional use certificate, it may be left unset if not needed.

3.3. Publication Sub-Protocol

The sub-publication protocol requests publication or withdrawal from publication of RPKI objects.

The publication protocol uses a common message format to request publication of any RPKI object. This format was chosen specifically to allow this protocol to accommodate new types of RPKI objects without needing changes to this protocol.

Both the <publish/> and <withdraw/> objects have a payload of an optional tag and a URI. The <publish/> query also contains the DER object to be published, encoded in Base64.

Note that every publish and withdraw action requires a new manifest,

thus every publish or withdraw action will involve at least two objects.

3.4. Error handling

Errors are handled similarly in both subprotocols, and they're handled at two levels.

Since all messages in this protocol are conveyed over HTTP connections, basic errors are indicated via the HTTP response code. 4xx and 5xx responses indicate that something bad happened. Errors that make it impossible to decode a query or encode a response are handled in this way.

Where possible, errors will result in an XML <report_error/> message which takes the place of the expected protocol response message. <report_error/> messages are CMS-signed XML messages like the rest of this protocol, and thus can be archived to provide an audit trail.

<report_error/> messages only appear in replies, never in queries. The <report_error/> message can appear in both the control and publication subprotocols.

Like all other messages in this protocol, the <report_error/> message includes a "tag" attribute to assist in matching the error with a particular query when using batching. It is optional to set the tag on queries but, if set on the query, it MUST be set on the reply or error.

The error itself is conveyed in the error_code (attribute). The value of this attribute is a token indicating the specific error that occurred.

The body of the <report_error/> element itself is an optional text string; if present, this is debugging information.

3.5. XML Schema

The following is a RelaxNG compact form schema describing the Publication Protocol.

```
 default namespace = "http://www.hactrn.net/uris/rpki/publication-spec/"

 # Top level PDU
 start = element msg {
    attribute version { "2" } ,
    ( ( attribute type { "query" }, query_elt*) |
```

```
      (attribute type { "reply" }, reply_elt*))
   }

   # PDUs allowed in a query
   query_elt = ( config_query | client_query | publish_query |
      withdraw_query )

   # PDUs allowed in a reply
   reply_elt = ( config_reply | client_reply | publish_reply |
      withdraw_reply | report_error_reply )

   # Tag attributes for bulk operations
   tag = attribute tag { xsd:token {maxLength="1024" } }

   # Base64 encoded DER stuff
   base64 = xsd:base64Binary

   # Publication URLs
   uri_t = xsd:anyURI { maxLength="4096" }
   uri = attribute uri { uri_t }

   # Handles on remote objects (replaces passing raw SQL IDs).  NB:
   # Unlike the up-down protocol, handles in this protocol allow
   # "/" as a hierarchy delimiter.
   object_handle = xsd:string {
      maxLength="255" pattern="[\-_A-Za-z0-9/]*" }

   # <config/> element (use restricted to repository operator)
   # config_handle attribute: create, list, and destroy commands
   # omitted deliberately.
   config_payload = (element bpki_crl { base64 }?)
   config_query |= element config { attribute action { "set" }, tag?,
      config_payload }
   config_reply |= element config { attribute action { "set" }, tag? }
   config_query |= element config { attribute action { "get" }, tag? }
   config_reply |= element config { attribute action { "get" }, tag?,
      config_payload }

   # <client/> element (use restricted to repository operator)
   client_handle = attribute client_handle { object_handle }
   client_payload = (attribute base_uri { uri_t }?, element bpki_cert {
      base64 }?, element bpki_glue { base64 }?)
   client_query |= element client { attribute action { "create" },
      tag?, client_handle, client_payload }
   client_reply |= element client { attribute action { "create" },
      tag?, client_handle }
   client_query |= element client { attribute action { "set" }, tag?,
      client_handle, client_payload }
```

```
client_reply |= element client { attribute action { "set" }, tag?,
   client_handle }
client_query |= element client { attribute action { "get" }, tag?,
   client_handle }
client_reply |= element client { attribute action { "get" }, tag?,
   client_handle, client_payload }
client_query |= element client { attribute action { "list" }, tag? }
client_reply |= element client { attribute action { "list" }, tag?,
   client_handle, client_payload }
client_query |= element client { attribute action { "destroy" },
   tag?, client_handle }
client_reply |= element client { attribute action { "destroy" },
   tag?, client_handle }

# <publish/> element
publish_query |= element publish { tag?, uri, base64 }
publish_reply |= element publish { tag?, uri }

# <withdraw/> element
withdraw_query |= element withdraw { tag?, uri }
withdraw_reply |= element withdraw { tag?, uri }

# <report_error/> element
error = xsd:token { maxLength="1024" }
report_error_reply = element report_error {
   tag?,
   attribute error_code { error },
   xsd:string { maxLength="512000" }?
   }
```

4. Operational Considerations

Placeholder section to talk about nesting children under parents in the same repository, to allow for a single rsync to fetch both (observing that the rsync setup times tends to dominate over the sync time). And, more distressingly, talk about the access control impacts of that nesting.

5. IANA Considerations

IANA is asked to register the application/rpki-publication MIME media type as follows:

```
MIME media type name:  application
MIME subtype name:  rpki-publication
Required parameters:  None
Optional parameters:  None
Encoding considerations:  binary
Security considerations:  Carries an RPKI Publication Protocol
   Message, as defined in this document.
Interoperability considerations:  None
Published specification:  This document
Applications which use this media type: HTTP
Additional information:
   Magic number(s):  None
   File extension(s):
   Macintosh File Type Code(s):
Person & email address to contact for further information:
   Rob Austein <sra@isc.org>
Intended usage:  COMMON
Author/Change controller: Rob Austein <sra@isc.org>
```

6. Security Considerations

7. References

7.1. Normative References

[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.

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

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

7.2. 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.

Authors' Addresses

Samuel Weiler
SPARTA, Inc.
7110 Samuel Morse Drive
Columbia, Maryland 21046
US

Email: weiler@tislabs.com

Anuja Sonalker
SPARTA, Inc.
7110 Samuel Morse Drive
Columbia, Maryland 21046
US

Email: Anuja.Sonalker@sparta.com

Rob Austein
ISC
950 Charter Street
Redwood City, CA 94063
USA

Email: sra@isc.org

Secure Inter-Domain Routing
Internet-Draft
Intended status: Informational
Expires: December 24, 2011

T. Manderson
ICANN
K. Sriram
US NIST
R. White
Cisco

June 22, 2011

Use Cases and Interpretation of RPKI Objects for Issuers and Relying Parties
draft-ietf-sidr-usecases-02

Abstract

This document provides use cases, directions, and interpretations for organizations and relying parties when creating or encountering RPKI object scenarios in the public RPKI in relation to the Internet routing system.

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 December 24, 2011.

Copyright Notice

Table of Contents

1. Introduction

This document provides suggested use cases, directions, and interpretations for organizations and relying parties when creating or encountering RPKI object scenarios in the public RPKI in relation to the Internet routing system.

1.1. Terminology

It is assumed that the reader is familiar with the terms and concepts described in "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [RFC5280], "A Profile for X.509 PKIX Resource Certificates" [I-D.ietf-sidr-res-certs] "X.509 Extensions for IP Addresses and AS Identifiers" [RFC3779], "A Profile for Route Origin Authorizations (ROAs)" [I-D.ietf-sidr-roa-format], "Validation of Route Origination in BGP using the Resource Certificate PKI and ROAs" [I-D.ietf-sidr-roa-validation], and BGP Prefix Origin Validation" [I-D.ietf-sidr-pfx-validate].

1.2. Definitions

The following definitions are in use in this document.

Autonomous System - A network under a single technical administration that presents a consistent picture of what destinations are reachable through it.

Autonomous System Number (ASN) - An officially registered number representing an autonomous system.

Prefix - A network address and an integer that specifies the length of a mask to be applied to the address to represent a set of numerically adjacent addresses.

Route - A prefix and a sequence of one or more autonomous system numbers.

Origin AS - The Autonomous System, designated by an ASN, which originates a route. Seen as the "First" ASN in a route.

Specific route - A route that has a longer prefix than an aggregate.

Aggregate route - A more general route in the presence of a specific route.

Covering Aggregate - A route that covers one or more specific routes.

Multi-homed Autonomous System - An Autonomous System that is connected, and announces routes, to two or more Autonomous Systems.

Multi-homed prefix or subnet - A prefix (i.e., subnet) that is originated via two or more Autonomous Systems to which the subnet is connected.

Resource - Internet (IP) addresses or Autonomous System Number.

Allocation - The set of resources provided to an entity or organization for its use.

Sub-allocation - The set of a resources subordinate to an allocation assigned to another entity or organization.

Transit Provider - An Autonomous System that carries traffic that neither originates nor is the destination of that traffic.

Upstream - See "Transit Provider".

Child - A Sub-allocation that has resulted from an Allocation.

Parent - An allocation from which the subject prefix is a Child.

Grandchild - A Sub-allocation from one or more previous Sub-allocations.

Grandparent - The allocation from which the prefix is a Grandchild.

Update prefix - The prefix seen in a routing update.

ROA prefix - The prefix described in a ROA.

Covering Prefix - The ROA Prefix is an exact match or a less specific when compared to the update prefix.

No relevant ROA - No ROA exists that has a covering prefix for the update prefix.

No other relevant ROA - No other ROA (besides any that is(are) already cited) that has a covering prefix for the update prefix.

1.3. 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.

2. Overview

2.1. General interpretation of RPKI object semantics

It is important that in the interpretation of relying parties (RP), or relying party routing software, that a 'make before break' stance is applied. This means that a RP should implement a routing decision process where a routing update ("route") is assumed to be intended unless proven otherwise by the existence of a valid RPKI object. For all of the cases in this document it is assumed that RPKI objects validate (or otherwise) in accordance with [I-D.ietf-sidr-res-certs], [I-D.ietf-sidr-arch], [I-D.ietf-sidr-roa-validation] unless otherwise stated.

While many of the examples provided here illustrate organizations using their own autonomous system numbers to originate routes, it should be recognised that a prefix holder need not necessarily be the holder of the autonomous system number used for the route origination.

3. Origination Use Cases

This section deals with the various use cases where an organization has Internet resources and will announce routes to the Internet. It is based on operational observations of the existing routing system.

3.1. Single Announcement

An organization (Org A with ASN 64496) has been allocated the prefix 192.168.2.0/24. It wishes to announce the /24 prefix from ASN 64496 such that relying parties interpret the route as intended.

The desired announcement (and organization) would be:

Prefix	Origin AS	Organization
192.168.2.0/24	AS64496	Org A

The issuing party should create a ROA containing the following:

asID	address	maxLength
64496	192.168.2.0/24	24

3.2. Aggregate with a More Specific

An organization (Org A with ASN 64496) has been allocated the prefix 10.1.0.0/16. It wishes to announce the more specific prefix 10.1.0.0/20 from ASN 64496 as well as the aggregate route such that relying parties interpret the routes as intended.

The desired announcements (and organization) would be:

Prefix	Origin AS	Organization
10.1.0.0/16	AS64496	Org A
10.1.0.0/20	AS64496	Org A

The issuing party should create a ROA containing the following:

asID	address	maxLength
64496	10.1.0.0/16	16
	10.1.0.0/20	20

3.3. Aggregate with a More Specific from a Different ASN

An organization (Org A with ASN 64496 and ASN 64499) has been allocated the prefix 10.1.0.0/16. It wishes to announce the more specific prefix 10.1.0.0/20 from ASN 64499 as well as the aggregate route from ASN 64496 such that relying parties interpret the routes as intended.

The desired announcements (and organization) would be:

Prefix	Origin AS	Organization
10.1.0.0/16	AS64496	Org A
10.1.0.0/20	AS64499	Org A

The issuing party should create ROAs containing the following:

asID	address	maxLength
64496	10.1.0.0/16	16

asID	address	maxLength
64499	10.1.0.0/20	20

3.4. Sub-allocation to a Multi-homed Customer

An organization (Org A with ASN 64496) has been allocated the prefix 10.1.0.0/16, it wishes to announce the more specific prefix 10.1.0.0/20 from ASN 64496. It has further delegated 10.1.16.0/20 to a customer (Org B with ASN 64511) who is multi-homed and will originate the prefix route from ASN 64511. ASN 64496 will also announce the aggregate route such that relying parties interpret the routes as intended.

The desirable announcements (and organization) would be:

Prefix	Origin AS	Organization
10.1.0.0/16	AS64496	Org A
10.1.0.0/20	AS64496	Org A
10.1.16.0/20	AS64511	Org B

The issuing party should create ROAs containing the following:

Org A.

asID	address	maxLength
64496	10.1.0.0/16	16
	10.1.0.0/20	20

Org B.

asID	address	maxLength
64511	10.1.16.0/20	20

3.5. Restriction of a New Allocation

An organization has recently been allocated the prefix 10.1.0.0/16. Its network deployment is not yet ready to announce the prefix and wishes to restrict all possible announcements of 10.1.0.0/16 and more specifics in routing using RPKI.

The following announcements would be considered undesirable:

Prefix	Origin AS	Organization
10.1.0.0/16	ANY AS	ANY
10.1.0.0/20	ANY AS	ANY
10.1.17.0/24	ANY AS	ANY

The issuing party should create a ROA containing the following:

asID	address	maxLength
0	10.1.0.0/16	32

This is known as an AS0-ROA [I-D.ietf-sidr-roa-validation]

3.6. Restriction of New ASN

An organization has recently been allocated an additional ASN 65535. Its network deployment is not yet ready to use this ASN and wishes to restrict all possible uses of ASN 65535 using RPKI.

The following announcements would be considered undesirable:

Prefix	Origin AS	Organization
ANY	AS65535	ANY

It is currently not possible to restrict use of Autonomous System Numbers

3.7. Restriction of a Part of an Allocation

An organization (Org A with ASN 64496) has been allocated the prefix 10.1.0.0/16. Its network topology permits the announcement of 10.1.0.0/17 and the /16 aggregate. However it wishes to restrict any possible announcement of 10.1.128.0/17 or more specifics of that /17 using RPKI.

The desired announcements would be:

Prefix	Origin AS	Organization
10.1.0.0/16	AS64496	Org A
10.1.0.0/17	AS64496	Org A

The following announcements would be considered undesirable:

Prefix	Origin AS	Organization
10.1.128.0/17	ANY AS	ANY
10.1.128.0/24	ANY AS	ANY

The issuing party should create ROAs containing the following:

asID	address	maxLength
64496	10.1.0.0/16	16
	10.1.0.0/17	17

asID	address	maxLength
0	10.1.128.0/17	32

3.8. Restriction of Prefix Length

An organization (Org A with ASN 64496) has been allocated the prefix 10.1.0.0/16, it wishes to announce the aggregate and any or all more specific prefixes up to and including a maximum length of /20, but never any more specific than a /20.

Examples of the desired announcements (and organization) would be:

Prefix	Origin AS	Organization
10.1.0.0/16	AS64496	Org A
10.1.0.0/17	AS64496	Org A
...	AS64496	Org A
10.1.128.0/20	AS64496	Org A

The following announcements would be considered undesirable:

Prefix	Origin AS	Organization
10.1.0.0/21	ANY AS	ANY
10.1.0.0/22	ANY AS	ANY
...	ANY AS	ANY
10.1.128.0/24	ANY AS	ANY

The issuing party should create a ROA containing the following:

asID	address	maxLength
64496	10.1.0.0/16	20

3.9. Restriction of Sub-allocation Prefix Length

An organization (Org A with ASN 64496) has been allocated the prefix 10.1.0.0/16, it sub-allocates several /20 prefixes to its multi-homed customers Org B with ASN 65535, and Org C with ASN 64499. It wishes to restrict those customers from advertising any corresponding routes more specific than a /22.

The desired announcements would be:

Prefix	Origin AS	Organization
10.1.0.0/16	AS64496	Org A
10.1.0.0/20	AS65535	Org B
10.1.128.0/20	AS64499	Org C
10.1.4.0/22	AS65535	Org B

The following example announcements (and organization) would be considered undesirable:

Prefix	Origin AS	Organization
10.1.0.0/24	AS65535	Org B
10.1.128.0/24	AS64499	Org C
.....	...	...
10.1.0.0/23	ANY AS	ANY

The issuing party (Org A) should create ROAs containing the following:

For Org A.

asID	address	maxLength
64496	10.1.0.0/16	16

For Org B.

asID	address	maxLength
65535	10.1.0.0/20	22

For Org C.

asID	address	maxLength
64499	10.1.128.0/20	22

3.10. Aggregation and Origination by an Upstream

Consider four organizations with the following resources, which were acquired independently from any transit provider. .

Organization	ASN	Prefix
Org A	AS64496	10.1.0.0/24
Org B	AS65535	10.1.3.0/24
Org C	AS64499	10.1.1.0/24
Org D	AS64512	10.1.2.0/24

These organizations share a common upstream provider Transit A (ASN 64497) that originates an aggregate of these prefixes with the permission of all four organizations.

The desired announcements (and organization) would be:

Prefix	Origin AS	Organization
10.1.0.0/24	AS64496	Org A
10.1.3.0/24	AS65535	Org B
10.1.1.0/24	AS64499	Org C
10.1.2.0/24	AS64512	Org D
10.1.0.0/22	AS64497	Transit A

It is currently not possible for an upstream to make a valid aggregate annoucement of indepentant prefixes. However the issuing parties should create ROAs containing the following:

Org A.

asID	address	maxLength
64496	10.1.0.0/24	24

Org B.

asID	address	maxLength
65535	10.1.3.0/24	24

Org C.

asID	address	maxLength
64499	10.1.1.0/24	24

Org D.

asID	address	maxLength
64512	10.1.2.0/24	24

3.11. Rogue Aggregation and Origination by an Upstream

Consider four organizations with the following resources which were acquired independently from any transit provider.

Organization	ASN	Prefix
Org A	AS64496	10.1.0.0/24
Org B	AS65535	10.1.3.0/24
Org C	AS64499	10.1.1.0/24
Org D	AS64512	10.1.2.0/24

These organizations share a common upstream provider Transit A (ASN 64497) that originates an aggregate of these prefixes where possible. In this situation organization B (ASN 65535, 10.1.3.0/24) does not wish for its prefix to be aggregated by the upstream provider.

The desired announcements (and organization) would be:

Prefix	Origin AS	Organization
10.1.0.0/24	AS64496	Org A
10.1.3.0/24	AS65535	Org B
10.1.1.0/24	AS64499	Org C
10.1.2.0/24	AS64512	Org D
10.1.0.0/23	AS64497	Transit A

The following announcement would be undesirable:

Prefix	Origin AS	Organization
10.1.0.0/22	AS64497	Transit A

It is currently not possible for an upstream to make a valid aggregate annoucement of indepentant prefixes. However the issuing parties should create ROAs containing the following:

Org A.

asID	address	maxLength
64496	10.1.0.0/24	24

Org B.

asID	address	maxLength
65535	10.1.3.0/24	24

Org C.

asID	address	maxLength
64499	10.1.1.0/24	24

Org D.

asID	address	maxLength
64512	10.1.2.0/24	24

4. Adjacency or Path Validation Use Cases

The SIDR WG was recently re-chartered (April 2011) to address AS path validation. Use cases pertaining to adjacency or path validation are beyond the scope of this document and would be addressed in a separate document.

5. Partial Deployment Use Cases

5.1. Parent does not do RPKI

An organization (Org A with ASN 64511) is multi-homed has been assigned the prefix 10.1.0.0/20 from its upstream (Transit X with ASN 64496). Org A wishes to announce the prefix 10.1.0.0/20 from ASN 64511 to its other upstream(s). Org A also wishes to create RPKI statements about the resource, however Transit X (ASN 64496) which

announces the aggregate 10.1.0.0/16 has not yet adopted RPKI.

The desired announcements (and organization with RPKI adoption) would be:

Prefix	Origin AS	Organization	RPKI
10.1.0.0/20	AS64511	Org A	Yes
10.1.0.0/16	AS64496	Transit X	No

RPKI is strictly hierarchical, therefore if Transit X does not do RPKI, Org A is unable to validly issue RPKI objects.

5.2. Only Some Children Participate in RPKI

An organization (Org A with ASN 64496) has been allocated the prefix 10.1.0.0/16 and participates in RPKI, it wishes to announce the more specific prefix 10.1.0.0/20 from ASN 64496. It has further delegated 10.1.16.0/20 and 10.1.32.0/20 to customers Org B with ASN 64511 and and Org C with ASN 65535 (respectively) who are multi-homed. Org B (ASN 64511) does not participate in RPKI. Org C (ASN 65535) participates in RPKI.

The desired announcements (and organization with RPKI adoption) would be:

Prefix	Origin AS	Organization	RPKI
10.1.0.0/16	AS64496	Org A	Yes
10.1.0.0/20	AS64496	Org A	Yes
10.1.16.0/20	AS64511	Org B	No
10.1.32.0/20	AS65535	Org C	YES

The issuing parties should create ROAs containing the following:

Org A.

asID	address	maxLength
64496	10.1.0.0/16	20

Org A issues for Org B.

asID	address	maxLength
64511	10.1.16.0/20	20

Org C.

asID	address	maxLength
65535	10.1.32.0/20	20

5.3. Grandchild Does Not Participate in RPKI

Consider the previous example with an extension by where Org B, who does not participate in RPKI, further allocates 10.1.17.0/24 to Org X with ASN 64512. Org X does not participate in RPKI.

The desired announcements (and organization with RPKI adoption) would be:

Prefix	Origin AS	Organization	RPKI
10.1.0.0/16	AS64496	Org A	Yes
10.1.0.0/20	AS64496	Org A	Yes
10.1.16.0/20	AS64511	Org B	No
10.1.32.0/20	AS65535	Org C	YES
10.1.17.0/24	AS64512	Org X	No

The issuing parties should create ROAs containing the following:

Org A.

asID	address	maxLength
64496	10.1.0.0/16	20

Org A issues for Org B.

asID	address	maxLength
64511	10.1.16.0/20	20

Org A issues for Org B's customer Org X..

asID	address	maxLength
64512	10.1.17.0/24	24

Org C.

asID	address	maxLength
65535	10.1.32.0/20	20

6. Transfer Use Cases

For transfer use cases, based on the preceding sections it would be easy to deduce what existing ROAs would need to be maintained or revoked and what new ROAs would need to be created. The resource transfer and timing of revocation/creation of the ROAs need to be performed based on the make-before-break principle and using suitable RIR procedures.

6.1. Transfer of in-use prefix and autonomous system number

Organization A holds the resource 10.1.0.0/20 and it is currently in use and originated from AS64496 with valid RPKI objects in place. Organization B has acquired both the prefix and ASN and desires an RPKI transfer on a particular date and time without adversely affecting the operational use of the resource.

The following RPKI objects would be created/revoked:

For Org. A, revoke the following ROA:

asID	address	maxLength
64496	10.1.0.0/20	20

For Org. B, add the following ROA:

asID	address	maxLength
64496	10.1.0.0/20	20

6.2. Transfer of in-use prefix

Organization A holds the resource 10.1.0.0/8 and it is currently in use and originated from AS64496 with valid RPKI objects in place. Organization B has acquired the address and desires an RPKI transfer on a particular date and time. This prefix will be originated by AS65535 as a result of this transfer.

The following RPKI objects would be created/revoked:

For Org. A, revoke the following ROA:

asID	address	maxLength
64496	10.1.0.0/8	8

For Org. B, add the following ROA when the resource certificate for 10.1.0.0/8 is issued to them (Org. B):

asID	address	maxLength
65535	10.1.0.0/8	8

6.3. Transfer of un-used prefix

Organization A holds the resource 10.1.0.0/8 and AS65535 (with RPKI objects). Organization B has acquired an unused portion (10.1.4.0/24) of the prefix and desires an RPKI transfer on a particular date and time. Organization B will originate a route 10.1.4.0/24 from AS64496

The following RPKI objects would be created/revoked:

For Org. A, leave the following ROA unchanged:

asID	address	maxLength
65535	10.1.0.0/8	8

For Org. B, add the following ROA when the resource certificate for 10.1.4.0/24 is issued to them (Org. B):

asID	address	maxLength
64496	10.1.4.0/24	24

Organization A may optionally provide ROA coverage for Organisation B by creating the following ROA preceding the RPKI transfer. The ROA itself is then naturally revoked when 10.1.4.0/24 is transferred to Organization B's resource certificate.

Org. A, adds the following ROA:

asID	address	maxLength
64496	10.1.4.0/24	24

7. Relying Party Use Cases

7.1. Prefix-Origin Validation use cases

These use cases try to systematically enumerate the situations a relying party may encounter while receiving a BGP update and making use of ROA information to interpret the validity of the prefix-origin information in the update. We enumerate the situations or scenarios

and include a recommendation for the expected outcome of prefix-origin validation. For description of prefix-origin validation algorithms, see [I-D.ietf-sidr-roa-validation] and [I-D.ietf-sidr-pfx-validate]. We use the terms Valid, Invalid, and Unknown as defined in [I-D.ietf-sidr-roa-validation]. Also see [I-D.ietf-idr-deprecate-as-sets] for work-in-progress in the IDR WG to deprecate AS_SETs in BGP updates. The use cases described here can be potentailly used as test cases for testing and evaluation of prefix-origin validation in router implementations; see for example [BRITE].

7.1.1. Covering ROA Prefix, maxLength Satisfied, and AS Match

ROA: {10.1.0.0/16, maxLength = 20, AS64496}

Update has {10.1.0.0/17, Origin = AS64496}

Recommended RPKI prefix-origin validation interpretation: Update is Valid.

Comment: This is a straight forward prefix-origin validation use case; it follows from the primary intention of creation of ROA by a resource owner.

7.1.2. Covering ROA Prefix, maxLength Exceeded, and AS Match

ROA: {10.1.0.0/16, maxLength = 20, AS64496}

Update has {10.1.0.0/22, Origin = AS64496}

No other relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update is Invalid.

Comment: In this case the maxLength specified in the ROA is exceeded by the update prefix.

7.1.3. Covering ROA Prefix, maxLength Satisfied, and AS Mismatch:

ROA: {10.1.0.0/16, maxLength = 24, AS64496}

Update has {10.1.88.0/24, Origin = AS65535}

No other relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update is Invalid.

Comment: In this case an AS other than the one specified in the ROA is originating an update. This may be a prefix or subprefix hijack situation.

7.1.4. Covering ROA Prefix, maxLength Exceeded, and AS Mismatch

ROA: {10.1.0.0/16, maxLength = 22, AS64496}

Update has {10.1.88.0/24, Origin = AS65535}

No other relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update is Invalid.

Comment: In this case the maxLength specified in the ROA is exceeded by the update prefix, and also an AS other than the one specified in the ROA is originating the update. This may be a subprefix hijack situation.

7.1.5. Covering ROA Prefix Not Found

Update has {240.1.1.0/24, Origin = AS65535}

No relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update’s validation status is Unknown.

Comment: In this case there is no relevant ROA that has a covering prefix for the update prefix. It could be a case of prefix or subprefix hijack situation, but this announcement does not contradict any existing ROA. During partial deployment, there would be some legitimate prefix-origin announcements for which ROAs may not have been issued yet.

7.1.6. Covering ROA Prefix Not Found but ROAs Exist for a Covering Set of More Specifics

ROA: {10.1.0.0/18, maxLength = 20, AS64496}

ROA: {10.1.64.0/18, maxLength = 20, AS64496}

ROA: {10.1.128.0/18, maxLength = 20, AS64496}

ROA: {10.1.192.0/18, maxLength = 20, AS64496}

Update has {10.1.0.0/16, Origin = AS64496}

No (directly) relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update’s validation status is Unknown.

Comment: In this case the update prefix is an aggregate, and it turns out that there exit ROAs for more specifics which, if combined, can help support validation of the announced prefix-origin pair. But it is very hard in general to breakup an announced prefix into constituent more specifics and check for ROA coverage for those more specifics, and hence this type of accommodation is not recommended.

7.1.7. AS_SET in Update and Covering ROA Prefix Not Found

Update has {10.1.0.0/16, Origin = [AS64496, AS64497, AS64498, AS64497]}

No relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update’s validation status is Unknown.

Comment: An extremely small percentage (˜0.1%) of eBGP updates are seen to have an AS_SET in them as origin; this is known as proxy aggregation. In this case, update with the AS_SET does not conflict with any ROA.

7.1.8. Singleton AS in AS_SET (in the Update), Covering ROA Prefix, and AS Match

Update has {10.1.0.0/24, Origin = [AS64496]} (Note: AS_SET with singleton AS appears in origin AS position.)

ROA: {10.1.0.0/22, maxLength = 24, AS64496}

Recommended RPKI prefix-origin validation interpretation: Update is Invalid.

Comment: In the spirit of [I-D.ietf-idr-deprecate-as-sets], any update with an AS_SET in it should not be considered valid (by ROA-based validation). If the update contains an AS_SET and a covering ROA exists, then no attempt should be made to match the ASN in the update with that in the covering ROA and the update should get an Invalid status.

7.1.9. Singleton AS in AS_SET (in the Update), Covering ROA Prefix, and AS Mismatch

Update has {10.1.0.0/24, Origin = [AS64496]}

(Note: AS_SET with singleton AS appears in origin AS position.)

ROA: {10.1.0.0/22, maxLength = 24, AS65535}

Recommended RPKI prefix-origin validation interpretation: Update is Invalid.

Comment: When there is at least one covering ROA, then the update with an AS_SET should get an Invalid status regardless of whether there is AS match or mismatch.

7.1.10. Multiple ASs in AS_SET (in the Update) and Covering ROA Prefix

Update has {10.1.0.0/22, Origin = [AS64496, AS64497, AS64498, AS64497]}

ROA: {10.1.0.0/22, maxLength = 24, AS65535}

No other relevant ROA.

Recommended RPKI prefix-origin validation interpretation: Update is Invalid.

Comment: When there is at least one covering ROA, then the update with an AS_SET should get an Invalid status.

7.1.11. Update has an AS_SET as Origin and ROAs Exist for a Covering Set of More Specifics

ROA: {10.1.0.0/18, maxLength = 20, AS64496}

ROA: {10.1.64.0/18, maxLength = 20, AS64497}

ROA: {10.1.128.0/18, maxLength = 20, AS64498}

ROA: {10.1.192.0/18, maxLength = 20, AS64499}

Update has {10.1.0.0/16, Origin = [AS64496, AS64497, AS64498, AS64497]}

No (directly) relevant ROA

Recommended RPKI prefix-origin validation interpretation: Update’s

validation status is Unknown.

Comment: In this case the aggregate of the prefixes in the ROAs is a covering prefix for the update prefix. The ASs in each of the contributing ROAs together form a set that matches the AS_SET in the update. But it is very hard in general to breakup an announced prefix into constituent more specifics and check for ROA coverage for those more specifics. In any case, it may be noted once again that in the spirit of [I-D.ietf-idr-deprecate-as-sets], any update with an AS_SET in it should not be considered valid (by ROA-based validation). In fact, the update in consideration would have received an Invalid staus if there were at least one covering ROA.

7.2. ROA Expiry or receipt of a CRL covering a ROA

Here we enumerate use cases corresponding to router actions when RPKI objects expire or are revoked. In the cases which follow, the terms "expired ROA" or "revoked ROA" are shorthand, and describe the appropriate expiry or revocation of the EE or Resource Certificate(s) that causes a relying party to consider the corresponding ROA to have expired or revoked.

7.2.1. ROA of Parent Prefix is Revoked

A certificate revocation list (CRL) is received which reveals that the ROA containing the prefix 10.1.0.0/22; maxLength 24 with ASN64496 is revoked. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. In absence of the revoked ROA, no covering ROA exists for 10.1.4.0/24.

The Relying Party interpretation would be: Route validation status is Unknown

7.2.2. ROA of Prefix Revoked while Parent Has Covering ROA with Different ASN

A CRL is received which reveals that the ROA containing the prefix 10.1.4.0/24; maxLength 24 with ASN64496 is revoked. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. A covering ROA exists for a parent prefix 10.1.4.0/22; maxlength 24 with ASN65535. No other covering ROA exists for the 10.1.4.0/24 prefix.

The Relying Party interpretation would be: Route is Invalid.

7.2.3. ROA of Prefix Revoked while that of Parent Prefix Prevails

A CRL is received which reveals that the ROA containing the prefix 10.1.4.0/24; maxLength 24 with ASN64496 is revoked. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. Additionally, the current ROA list has a valid ROA containing the parent prefix 10.1.0.0/22; maxLength 24 with ASN64496.

The Relying Party interpretation would be: Route is Valid.

(Clarification: Perhaps the revocation of ROA for prefix 10.1.4.0/24 was initiated just to eliminate redundancy.)

7.2.4. ROA of Grandparent Prefix Revoked while that of Parent Prefix Prevails

A CRL is received which reveals that the ROA containing the prefix 10.1.0.0/20; maxLength 24 with ASN64496 is revoked. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. Additionally, the current ROA list has a valid ROA containing the prefix 10.1.0.0/22; maxLength 24 with ASN64496.

The Relying Party interpretation would be: Route is Valid.

(Clarification: ROA for less specific grandparent prefix 10.1.0.0/20 was revoked or withdrawn.)

7.2.5. Expiry of ROA of Parent Prefix

A scan of the ROA list reveals that the ROA containing the prefix 10.1.0.0/22; maxLength 24 with ASN64496 has expired. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. In absence of the expired ROA, no covering ROA exists for 10.1.4.0/24.

The Relying Party interpretation would be: Route validation status is Unknown

7.2.6. Expiry of ROA of Prefix while Parent Has Covering ROA with Different ASN

A scan of the ROA list reveals that the ROA containing the prefix 10.1.4.0/24; maxLength 24 with ASN64496 has expired. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. A valid covering ROA exists for a parent prefix 10.1.4.0/22; maxlength 24 with ASN65535. No other covering

ROA exists for the prefix.

The Relying Party interpretation would be: Route is Invalid.

7.2.7. Expiry of ROA of Prefix while that of Parent Prefix Prevails

A scan of the ROA list reveals that the ROA containing the prefix 10.1.4.0/24; maxLength 24 with ASN64496 has expired. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. Additionally, the current ROA list has a valid ROA containing the parent prefix 10.1.0.0/22; maxLength 24 with ASN64496.

The Relying Party interpretation would be: Route is Valid.

7.2.8. Expiry of ROA of Grandparent Prefix while that of Parent Prefix Prevails

A scan of the ROA list reveals that the ROA containing the prefix 10.1.0.0/20; maxLength 24 with ASN64496 is revoked. Further, a prefix route exists in the Internet routing system for 10.1.4.0/24 originated from ASN64496. Additionally, the current ROA list has a valid ROA containing the prefix 10.1.0.0/22; maxLength 24 with ASN64496.

The Relying Party interpretation would be: Route is Valid.

8. Acknowledgements

The authors are indebted to both Sandy Murphy and Sam Weiler for their guidance. Further, the authors would like to thank Curtis Villamizar, Steve Kent, and Danny McPherson for their technical insight and review.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This memo requires no security considerations

11. References

11.1. Normative References

[I-D.ietf-idr-deprecate-as-sets]
Kumari, W. and K. Sriram, "Deprecation of the use of BGP AS_SET, AS_CONFED_SET.", draft-ietf-idr-deprecate-as-sets-04 (work in progress), May 2011.

[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-pfx-validate]
Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-01 (work in progress), February 2011.

[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-22 (work in progress), May 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-roa-validation]
Huston, G. and G. Michaelson, "Validation of Route Origination using the Resource Certificate PKI and ROAs", draft-ietf-sidr-roa-validation-10 (work in progress), November 2010.

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

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, June 2005.

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

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

[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.

11.2. Informative References

[BRITE] "BRITE: BGPSEC/RPKI Interoperability Test and Evaluation", Developed by the National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, <http://brite.antd.nist.gov/statics/about>.

Authors' Addresses

Terry Manderson
ICANN

Email: terry.manderson@icann.org

Kotikalapudi Sriram
US NIST

Email: ksriram@nist.gov

Russ White
Cisco

Email: russ@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 6, 2012

K. Sriram, Ed.
USA National Institute of
Standards and Technology (NIST)

July 5, 2011

BGPSEC Design Choices and Summary of Supporting Discussions
draft-sriram-bgpsec-design-choices-00

Abstract

A wide range of design choices are being discussed for BGPSEC. This document has been written to capture the design rationale for the evolving BGPSEC specification. It lists the decisions that have been made in favor of or against each choice, and presents brief summaries of the arguments that aided the decision process. The document will be updated periodically as the BGPSEC design discussions make further progress and additional design considerations are discussed and possibly finalized.

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 January 6, 2012.

Table of Contents

1. Introduction

The goal of BGPSEC effort is to enhance the security of BGP by enabling full AS path validation based on cryptographic principles. Work on prefix-origin validation based on a Resource certificate PKI (RPKI) is already nearing completion in the IETF SIDR WG. The BGPSEC effort is aimed at taking advantage of the same RPKI infrastructure developed in the SIDR WG to add cryptographic signatures to BGP updates, so that routers can perform full AS path validation [I-D.ietf-sidr-bgpsec-threats] [I-D.ietf-sidr-bgpsec-reqs] [I-D.ietf-sidr-bgpsec-overview] [I-D.ietf-sidr-bgpsec-protocol]. The key high-level design goals of BGPSEC protocol are as follow [I-D.ietf-sidr-bgpsec-reqs]:

- Rigorous path validation for all announced prefixes; not merely showing that a path is not impossible.
- Incremental deployment capability; no flag-day requirement for global deployment.
- Protection of AS paths only in inter-domain routing (eBGP); not applicable to iBGP (or to IGPs).
- Aim for no increase in provider’s data exposure (e.g., require no disclosure of peering relations, etc).

Specification of the BGPSEC protocol is provided in [I-D.ietf-sidr-bgpsec-protocol]. This document is a companion to [I-D.ietf-sidr-bgpsec-protocol] intended to provide design justifications for BGPSEC. This document lists the decisions that have been made in favor of or against various design choices, and presents brief summaries of the discussions that weighed in the pros and cons and aided the decision process. The document will be updated periodically as the BGPSEC design discussions make further progress and additional design considerations are discussed and possibly finalized.

The design choices and discussions are presented under the following eight broad categories (with many subtopics within each category): (1) Creating Signatures and the Structure of BGPSEC Update Messages, (2) Withdrawal Protection, (3) Signature Algorithms and Router Keys, (4) Optimizations and Resource Sizing, (5) Incremental Deployment and Negotiation of BGPSEC, (6) Interaction of BGPSEC with Common BGP Features, (7) BGPSEC Validation, and (8) Operational Considerations.

2. Creating Signatures and the Structure of BGPSEC Update Messages

2.1. Origin Validation Using ROA

2.1.1. Decision

Prefix-Origin validation using Route Origin Authorization (ROA) is necessary and complements AS path attestation based on signed updates. Thus the BGPSEC design makes use of the origin AS validation capability provided by the RPKI.

2.1.2. Discussion

Prefix-Origin validation using RPKI constructs as developed in the IETF SIDR WG is a necessary component of BGPSEC, i.e., it provides cryptographic validation that the first hop AS is authorized to originate a route for the prefix in question.

2.2. Attributes Signed by an Originating AS

2.2.1. Decision

An originating AS will sign over the prefix, its own ASN, the next ASN, and a signature Expire Time (see Section 3.2) for the update. The update signatures will be carried in a new optional, non-transitive BGP attribute.

2.2.2. Discussion

The next hop ASN is included in the signature to prevent an update from being re-directed to another ASN (e.g., by a MITM).

It was decided that only the originating AS needs to insert a signature Expire Time in the update, as it is the originator of the route. The origin AS also will re-originate, i.e., beacon, the update prior to the Expire Time of said advertisement (see Section 3.2). (For an explanation of why upstream ASes do not insert their respective signature Expire Times, please see Section 3.2.2.)

It was decided that each signed update would include only one prefix. If more than one prefix were included, and an upstream AS elected to propagate the advertisement for a subset of the prefixes, then the signature on the update would break (see Section 5.1 and Section 5.2). If a mechanism were employed to preserve prefixes that were dropped, this would reveal info to later ASes that is not revealed in normal BGP operation. Thus a tradeoff was made, to preserve the level of route info exposure that is intrinsic to BGP, vs. the performance hit implied by limiting each update to carry only

one prefix.

The signature data is carried in an optional, non-transitive BGP attribute. The attribute is optional because this is the standard mechanism available in BGP to propagate new types of data. It was decided that the attribute should be non-transitive because we were concerned about the impact of sending the (potentially large) signatures to routers who don't understand them. Also, if a router that doesn't understand BGPSEC somehow gets a message with the signatures attribute then it would be undesirable for that router to forward the signatures on to all of its neighbors, especially those who do not understand BGPSEC, and who may choke badly if they receive a very large optional BGP attribute.

2.3. Attributes Signed by an Upstream AS

In the context of BGPSEC and throughout this document, an "upstream AS" simply refers to an AS that is further along in an AS path (origin AS being the nearest to a prefix). In principle, an AS that is upstream from an originating AS would sign the combined information including the prefix, Expire Time, AS path, and next ASN. There are multiple choices for what is actually signed by an upstream AS: (1) Sign over the combination of prefix, Expire Time, AS path, and next ASN; or (2) Sign over just the combination of previous signature (i.e., signature of the neighbor AS who forwarded the update) and next ASN; or (3) Sign over everything that was received from preceding AS; thus, ASi signs over prefix, Expire Time, {ASi, AS(i-1), AS(i-2), ..., AS2, AS1}, AS(i+1)(i.e., next ASN), and {Sig(i-1), Sig(i-2), ..., Sig2, Sig1}.

2.3.1. Decision

It was decided that that Method 2 will be used. Please see [I-D.ietf-sidr-bgpsec-protocol] for additional protocol details and syntax.

2.3.2. Discussion

The rationale for this choice (Method 2) was as follows. Signatures are performed over hash blocks. When the number of bytes to be signed exceeds one hash block, then the remaining bytes will overflow into a second hash block, which results in performance penalty. So it is advantageous to minimize the number of bytes being hashed. Also, an analysis of the three options noted above did not indentify any vulnerabilities associated with this approach.

2.4. What Attributes Are Not Signed

2.4.1. Decision

Any attributes other than those identified in Section 2.2 and Section 2.3 are not signed. Examples of such attributes are Community Attribute, NO-EXPORT Attribute, Local_Pref, etc.

2.4.2. Discussion

The above stated attributes that are not signed are viewed as local (e.g., do not need to propagate beyond next hop) or lack clear security needs. NO-EXPORT is sent over a secured next-hop and does not need signing. BGPSEC design should work with any transport layer protections. Certainly, we are assuming that the transport layer must be protected hop by hop (if only to prevent malicious session termination).

2.5. Receiving Router Actions

2.5.1. Decision

The expected router actions on receipt of a signed update are described by the following example. Consider an update that was originated by AS1 with prefix p and has traversed the AS path [AS(i-1) AS(i-2) .... AS2 AS1] before arriving at ASi. Let the Expire Time (inserted by AS1) for the signature in this update be denoted as Te. The update is to be processed at ASi and possibly forwarded to AS(i+1). Let the attestations (signatures) inserted by each router in the AS path be denoted by Sig1, Sig2, ..., Sig(i-2), and Sig(i-1) corresponding to AS1, AS2, ... , AS(i-2), and AS(i-1), respectively.

The method (#2 in Section 2.3) selected for signing requires a receiving router in ASi to perform the following actions:

- Validate the prefix-origin pair (p, AS1) by performing a ROA match.
- Verify that Te is greater than the clock time at the router performing these checks.
- Check Sig1 with inputs {AS2, AS1, p, Te}.
- Check Sig2 with inputs {Sig1, AS3}.
- Check Sig3 with inputs {Sig2, AS4}.

- o ...
- o ...
- o Check Sig(i-2) with inputs {Sig(i-3), AS(i-1)}.
- o Check Sig(i-1) with inputs {Sig(i-2), ASi}.
- o If the route that has been verified is selected as the best path (for prefix p), then generate Sig(i) with inputs {Sig(i-1), AS(i+1)}, and generate an update including Sig(i) to AS(i+1).

2.5.2. Discussion

See Section 8.1 for suggestions regarding efficient sequencing of BGPSEC validation processing in a receiving router. Some or all of the validation actions may be performed by an off-board server (see Section 9.3).

2.6. Prepending of ASes in AS Path

2.6.1. Decision

Prepending will be allowed. Prepending is defined as including more than one instance of the AS number of the router that is signing the update.

2.6.2. Discussion

For now, the protocol calls for a signature to be associated with each prepended AS. The optimization of having just one signature for multiple prepended ASes will be pursued later. If such optimization is used, a replication count could be included (in the signed object) to specify how many times an AS was prepended.

2.7. What RPKI Data Need be Included in Updates

2.7.1. Decision

Concerning inclusion of RPKI data in an update, it was decided that only the Subject Key Identifier (SKI) of the router cert must be included in a signed update. This info identifies the router certificate, based on the SKI generation criteria defined in [I-D.ietf-sidr-res-certs].

2.7.2. Discussion

We discussed if each router public key certificate should be included in a signed update. Inclusion of this information might be helpful for routers that do not have access to RPKI servers or temporarily lose connectivity to them. But we can safely assume that in majority of network environments, intermittent connectivity would not be a problem. So it is best to avoid this complexity because majority of the use environments do not have connectivity constraints. Because the SKI of a router certificate is a hash of the public key of that certificate, it suffices to select the public key from that certificate. This design assumes that each BGPSEC router has access to a cache containing the relevant data from (validated) router certificates.

3. Withdrawal Protection

3.1. Withdrawals Not Signed

3.1.1. Decision

Withdrawals are not signed.

3.1.2. Discussion

In the current BGP protocol, any AS can withdraw, at any time, any prefix it previously announced. The rationale for not signing withdrawals is that BGPSEC assumes use of transport security between neighboring BGPSEC routers. Thus no external entity can inject an update that withdraws a route, or replay a previously transmitted update containing a withdrawal. Because the rationale for withdrawing a route is not visible to a neighboring BGPSEC router, there are residual vulnerabilities associated with withdrawals. For example, a router that advertised a (valid) route may fail to withdraw that route when it is no longer viable. A router also might re-advertise a route that it previously withdrew, before the route is again viable. This latter vulnerability is mitigated by the Expire Time value in an AS path signature (see Section 3.2).

Repeated withdrawals and announcements for a prefix can run up the BGP RFD penalty and may result in unreachability for that prefix at upstream routers. But what can the attacker gain from doing so? This phenomenon is intrinsic to the design and operation of RFD. We do not concern ourselves about this in the BGPSEC design.

3.2. Signature Expire Time for Withdrawal Protection (a.k.a. Mitigation of Replay Attacks)

3.2.1. Decision

Only the originating AS inserts a signature Expire Time in the update; all other ASes along an AS path do not insert Expire Times associated with their respective signatures. Further, the originating AS will re-originate a route sufficiently in advance of the Expire Time of its signature so that other ASes along an AS path will typically receive the re-originated route well ahead of the current Expire Time for that route.

The duration of the signature Expire Time is recommended to be on the order of days (preferably) but it may be on the order of hours (about 4 to 8 hours) in some cases, where extra replay protection is percieved to be critical.

Each AS should stagger the Expire Time values in the routes it originates. Re-origination will be done, say, at time Tb after origination or the last re-origination, where Tb will equal a certain percentage of the Expire Time, Te (for example, Tb = 0.75 x Te). The percentage will be configurable and additional guidance can be provided via an operational considerations document later. Further, the actual re-origination time ought to be jittered with a uniform random distribution over a short interval {Tb1, Tb2} centered at Tb.

It is also recommended that a receiving BGPSEC router should detect if the only attribute change in an announcement (relative to the current best path) is the expire time (besides, of course, the signatures). In that case, assuming that the update is found valid, the route processor should not re-announce the route to BGP-4 only (i.e., non-BGPSEC) peers. (It still has to sign and re-announce the route to BGPSEC speakers.) This procedure will reduce BGP chattiness for the non-BGPSEC border routers.

3.2.2. Discussion

Mitigation of (update) replay attacks can be thought of as protection against malicious re-advertisement of withdrawn routes. If each AS along a path were to insert its own signature Expire Time, then there would be much additional BGP chattiness and increase in BGP processing load due to the need to detect and react to multiple (possibly redundant) signature Expire Times. Furthermore, there would be no extra benefit from the point of view of mitigation of replay attacks as compared to having a single Expire Time corresponding to the signature of the originating AS.

The recommended Expire Time value is on the order of days but 4 to 8 hours may used in some cases on the basis of percieved need for extra protection from replay attacks. Thus, different ASes may choose different values based on the perceived need to protect against route replays. (A shorter Expire Time reduces the window during which an AS can replay the route, even if the route has been withdrawn by a downstream AS. However, shorter Expire Time values cause routes to be refreshed more often, and thus causes more BGP chatter.) Even a 4 hours duration seems adequate to keep the re-origination workload manageable. For example, if 500K routes are re-originated every 4 hours, it amounts to an increase in BGP update load of at least 35 updates per second; this can be considered reasonable. However, further analysis is needed to confirm these recommendations.

It was stated above that originating AS will re-originate a route sufficiently in advance of its Expire Time. What is considered sufficiently in advance? For this, modeling should be performed to determine the 95th-percentile convergence time of update propagation in BGPSEC enabled Internet.

Each BGPSEC router should stagger the Expire Time values in the updates it originates, especially during table dumps to a neighbor or during its own recovery from a BGP session failure. By doing this, the re-origination workload at the router will be dispersed.

3.3. Should Route Expire Time be Communicated in a Separate Message

3.3.1. Decision

The idea of sending a new signature expire time in a special message (rather than re-transmitting the entire update with signatures) was considered. However, it was decided not to do this. Re-origination to communicate a new signature Expire Time will be done by propagation of a normal update message; no special type of message will be required.

3.3.2. Discussion

It was suggested that if re-beaconing of signature Expire Time is carried in a separate special message, then update processing load may be reduced. But it was recognized that such re-beaconing message necessarily entails AS path and prefix information, and hence cannot be separated from the update.

It was observed that at the edge of the Internet, there are frequent updates that may result from simple situations like BGP session being switched from one interface to another (e.g., from primary to backup) between two peering ASes (e.g., customer and provider). With BGP-4,

these updates do not propagate beyond the two ASes involved. But with BGPSEC, the customer AS will put in a new signature Expire Time each time such an event happens, and hence the update will need to propagate throughout the Internet (limited only by best path selection process). It was accepted that this cost of added churn will be unavoidable.

3.4. Effect of Expire-Time Updates in BGPSEC on RFD

3.4.1. Decision

With regard to the Route Flap Damping (RFD) protocol [RFC2439][JunOS][CiscoIOS], no differential treatment is required for Expire-Time triggered (re-beaconed) BGPSEC updates.

However, it was noted that it would be preferable if these updates did not cause route churn (and perhaps not even require any RFD related processing), since they are identical except for the change in the Expire Time value. The way this can be accomplished is by not assigning RFD penalty to Expire-Time triggered updates. If the community agrees, this could be accommodated, but a change to the BGP-RFD protocol specification will be required.

3.4.2. Discussion

Summary:

The decision is supported by the following observations: (1) Expire Time-triggered updates are generally not preceded by withdrawals, and hence the path hunting and associated RFD exacerbation [Mao02][RIPE378] problems are not anticipated; (2) Such updates would not normally change the best path (unless another concurrent event impacts the best path); (3) Expire Time-triggered updates would have negligible impact on RFD penalty accumulation because the re-advertisement interval is much longer relative to the half-time of decay of RFD penalty. Elaborating further on reason #4 above, we note that the re-advertisement of a route for a given address prefix from a given peer will be received at intervals of a few or several hours (see Section 3.2). During that time period, any incremental contribution to RFD penalty due to a Expire Time-triggered update would decay sufficiently to have negligible (if any) impact on damping of said address prefix. Additional details of this analysis and justification can be found below.

Further Details of the Analysis and Justification:

The frequency with which RFD penalty increments may be triggered for a given prefix from a given peer is the same as the re-beaconing

frequency for that prefix from its origin AS. The re-beaconing frequency is on the order of once every few or several hours (see Section 3.2). The incremental RFD penalty assigned to a prefix due to a re-beaconed update varies depending on the implementation. For example, it appears that JunOS implementation [JunOS] would assign a penalty of 1000 or 500 depending on whether the re-beaconed update is regarded as a re-advertisement or an attribute change, respectively. Normally, a re-beaconed update would be treated as a case of attribute change. The Cisco implementation [CiscoIOS] on the other hand assigns an RFD penalty only in the case of an actual flap (i.e., a route is available, then unavailable, or vice versa). So it appears that Cisco implementation of RFD would not assign any penalty for a re-beaconed update (i.e., a route was already advertised previously; not withdrawn; and the re-beaconed update is merely updating the expire time attribute). Even if we assume that an RFD penalty of 500 is assigned (corresponding to attribute change in JunOS RFD implementation), it can be illustrated that the incremental affect it would have on damping the prefix in consideration would be negligible. The reason for this is as follows. The half-time of RFD penalty decay is normally set to 15 minutes, whereas the re-beaconing frequency is on the order of once every few or several hours. An incremental penalty of 500 would decay to 31.25 in one hour; 0.12 in two hours; 3x10^(-5) in three hours. It may also be noted that the threshold for route suppression is 3000 in JunOS and 2000 in Cisco IOS. Based on the foregoing analysis, it may be concluded that routine re-beaconing by itself would not result in RFD suppression of routes in the BGPSEC protocol.

4. Signature Algorithms and Router Keys

4.1. Signature Algorithms

4.1.1. Decision

Initially, 256-bit ECDSA with SHA-256 will be used. One other algorithm, e.g., 256-bit DSA also will be used during prototyping and testing. The use of a second algorithm is needed to verify the ability of the BGPSEC implementations to change from a current algorithm to the next algorithm.

4.1.2. Discussion

Initially, we anticipated using 2048-bit RSA algorithm for BGPSEC update signatures because it is being used ubiquitously in the RPKI system. However, we elected to change to using ECDSA-256 because it yields a smaller signature size, so that the RIB sizes needed for BGPSEC would be much smaller [RIB_size].

Testing with two different signature algorithms (256-bit ECDSA and 256-bit RSA) for transition from one to the other will increase confidence in the prototyped protocol.

For Elliptic Curve Cryptography (ECC) algorithms, according to [RFC6090], optimizations and specialized algorithms (e.g., for speed-ups) have active IPR, but the basic (un-optimized) algorithms do not have IPR encumbrances.

4.2. Agility of Signature Algorithms

4.2.1. Decision

During the transition period from one algorithm, i.e., current algorithm, to the next (new) algorithm, the updates will carry two sets of signatures (i.e., two Signature-List Blocks), one corresponding to each algorithm. Each Signature-List Block will be preceded by its type-length field and an algorithm-suite identifier. A BGPSEC speaker that has been upgraded to handle the new algorithm should validate both Signature-List Blocks, and then add its corresponding signature to each Signature-List Block for forwarding the update to the next AS. A BGPSEC speaker that has not been upgraded to handle the new algorithm will strip off the Signature-List Block of the new algorithm, and forward the update after adding its own sig to the Signature-List Block of the current algorithm.

It was decided that there will be at most two Signature-List Blocks per update.

4.2.2. Discussion

A length field in the Signature-List Block allows for delineation of the two signature blocks. Hence, a BGPSEC router that doesn’t know about a particular algorithm suite (and hence doesn’t know how long signatures were for that algorithm suite) could still skip over the corresponding Signature-List Block when parsing the message.

The overlap period between the two algorithms is expected to last two to four years. The RIB memory and cryptographic processing capacity will have to be sized to cope with such overlap periods when updates would contain two sets of sigs [RIB_size].

The lifetime of a signature algorithm is anticipated to be much longer than the duration of a transition period from current to new algorithm. It is fully expected that all ASes will have converted to the required new algorithm within a certain amount of time that is much shorter than the interval in which a subsequent newer algorithm may be investigated and standardized for BGPSEC. Hence, the need for

more than two Signature-List Blocks per update is not envisioned.

4.3. Sequential Aggregate Signatures

4.3.1. Decision

There is currently weak or no support for the Sequential Aggregate Signature (SAS) approach. Please see in the discussion section below for a brief description of what SAS is and what its pros and cons are.

4.3.2. Discussion

In Sequential Aggregate Signature (SAS) method, there would be only one (aggregated) signature per signature block, irrespective of the number of AS hops. For example, ASn (nth AS) takes as input the signatures of all previous ASes [AS1, ..., AS(n-1)] and produces a single composite signature. This composite signature has the property that a recipient who has the public keys for AS1, ..., ASn can verify (using only the single composite signature) that all of the ASes actually signed the message. SAS could potentially result in savings in bandwidth, PDU size, and maybe in RIB size but the signature generation and validation costs will be higher as compared to one signature per AS hop.

SAS schemes exist in the literature, typically based on RSA or equivalent. In order to do SAS with RSA, and based on the algorithm choices already adopted for the RPKI, a 2048-bit signature size would be required. Without SAS, a DSA with 320- bit signature (1024-bit key) or ECDSA with 512-bit signature (256-bit key) would suffice, for equivalent cryptographic strength. The larger signature size of RSA used with SAS undermines the advantages of SAS, because the average hop count, i.e., number of ASes, for a route is about 3.8. In the end, it may turn out that SAS has more complexity and does not provide sufficient savings in PDU size or RIB size to merit its use. Further exploration of this is needed to better understand SAS properties and applicability for BGPSEC. There is also a concern that SAS is not a time-tested cryptographic technique and thus its adoption is potentially risky.

4.4. Protocol Extensibility

There is a clearly a need to specify a transition path from a current protocol specification to a new version. When changes to the processing of the BGPSEC_Path_Signatures are required, that will require for a new version of BGPSEC. Examples of this include changes to the data that is protected by the BGPSEC signatures or adoption of a signature algorithm in which the number of signatures

in the Signature-List Block may not correspond to one signature per AS in the AS-PATH (e.g., aggregate signatures).

4.4.1. Decision

The protocol-version transition mechanism here is analogous to the algorithm transition discussed in Section 4.2. During the transition period from one protocol version (i.e., current version) to the next (new) version, updates will carry two sets of signatures (i.e., two Signature-List Blocks), one corresponding to each version. A protocol-version identifier is included with each Signature-List Block. Hence, each Signature-List Block will be preceded by its type-length field and a protocol-version identifier. A BGPSEC speaker that has been upgraded to handle the new version should validate both Signature-List Blocks, and then add its corresponding signature to each Signature-List Block for forwarding the update to the next AS. A BGPSEC speaker that has not been upgraded to handle the new protocol version will strip off the Signature-List Block of the new version, and forward the update with an attachment of its own signature to the Signature-List Block of the current version.

4.4.2. Discussion

In the case that change to BGPSEC is 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. At this point a transition would begin which is analogous to the algorithm transition discussed in Section 4.2. During the transition period all BGPSEC speakers will simultaneously include both the BGPSEC_PATH_SIGNATURES (curent) 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 will 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.

4.5. Key Per Router (Rouge Router Problem)

4.5.1. Decision

Within each AS, each individual BGPSEC router can have a unique pair of private and public keys.

4.5.2. Discussion

If a router is compromised, its key pair can be revoked independently, without disrupting the other routers in the AS. Each

per-router key-pair will be represented in an end-entity certificate issued under the CA cert of the AS. The Subject Key Identifier (SKI) in the signature points to the router certificate (and thus the unique public key) of the router that affixed its signature, so that a validating router can reliably identify the public key to use for signature verification.

4.6. Router ID

4.6.1. Decision

The router certificate Subject name will be the string "router" followed by a decimal representation of a 4-byte AS number followed by the router ID. See the current RFCs for preferred standard textual representations for 4-byte ASNs [RFC5396] and router IDs [RFC2673].

4.6.2. Discussion

Every X.509 certificate requires a Subject name. The stylized Subject name adopted here is intended to facilitate debugging, by including the ASN and router ID.

5. Optimizations and Resource Sizing

5.1. Update Packing and Repacking

In the current BGP protocol (BGP-4) operation [RFC4271], a BGP router normally packs multiple prefix (NLRI) announcements into one update if the prefixes all share a common AS path and other BGP attributes. When a router forwards updates to other peers, it can pack multiple prefixes (based on shared AS path and attributes) into one update. (The new update may be include a subset of the prefixes that were packed in a received update.)

5.1.1. Decision

The initial BGPSEC specification [I-D.ietf-sidr-bgpsec-protocol] does not accommodate update packing. Each update contains exactly one prefix. This avoids the complexity that would arise if an upstream AS decided to generate an update based on only a subset of the received prefixes. BGPSEC recommendation regarding packing and repacking will be revisited when optimizations are considered in the future.

5.1.2. Discussion

Currently, with BGP-4, there are, on average, approximately 4 prefixes announced per update [RIB_size]. So the number of BGP updates (carrying announcements) is about 4 times fewer, on average, as compared to the number of prefixes announced.

The current decision is to include only one prefix per secured update (see Section 2.2 and Section 2.3). When optimizations are considered in the future, the possibility of packing multiple prefixes into an update can be considered. (Please see Section 5.2 for a discussion of signature per prefix vs. signature per update.) Repacking could be performed if signatures were generated on a per prefix basis. However, one problem regarding this approach, i.e., multiple prefixes in a BGP update but with separate signatures for each prefix, is that the resuting BGP update violates the basic definition of a BGP update. That is becuase the different prefixes will have different signature and expire-time attibutes, while a BGP update (by definition) must have the same set of shared attributes for all prefixes it carries.

Note that if we choose to supporting packing, unpacking, and repacking of prefixes, each prefix will need to have its own signature Expire Time. This is because if an AS elects to extract and propagate only one prefix from a packed, secured, update, an Expire Time needs to be present for that prefix.

5.2. Signature Per Prefix vs. Signature Per Update

5.2.1. Decision

The initial design calls for including exactly one prefix per update, hence there is only one signature in each secured update (modulo algorithm transition conditions). Optimizations will be examined later.

5.2.2. Discussion

Some notes to assist in future optimization discussions: In the general case of one signature per update, multiple prefixes may be signed with one signature together with their shared AS path, next ASN, and Expire Time. If signature per update is used, then there are potentially savings in update PDU size as well as RIB memory size. But if there are any changes made to the announced prefix set along the AS path, then the AS where the change occurs would need to insert an Explicit Path Attribute (EPA)[I-D.draft-clynn-s-bgp]. The EPA conveys information regarding what the prefix set contained prior to the change. There would be one EPA for each AS that made such a

modification, and there would be a way to associate each EPA with its corresponding AS. This enables an upstream AS to be able to know and to verify what was announced and signed by prior ASs in the AS path (in spite of changes made to the announced prefix set along the way). The EPA adds complexity to processing (signature generation and validation), further increases the size of updates and, thus of the RIB, and exposes data to downstream ASes that would not otherwise be exposed. Not all the pros and cons of packing and repacking in the context of signature per prefix vs. signature per update (with packing) have been evaluated. But the current recommendation is for having only one prefix per update (no packing); so there is no need for the EPA attribute.

5.3. Max PDU Size and PDU Negotiation

The current BGP-4 update PDU size is limited to 4096 bytes (4KB). The probability of exceeding the current max PDU size of 4KB will be higher for BGPSEC as compared to that for BGP-4 [RIB_size]. Hence, there is need for adopting a higher max PDU size for BGPSEC.

5.3.1. Decision

The current thinking is that the max PDU size should be increased to 64 KB [I-D.ietf-idr-bgp-extended-messages] so that there is sufficient room to accommodate two signature-list blocks (i.e., one block with a current algorithm and another block with a new algorithm during transition periods) for long paths. The larger max PDU also may be required to accommodate multiple prefix announcements in an update (each with up to two signature blocks) if we decide to adopt some optimizations in future versions of the BGPSEC specification.

It was decided that the max PDU size negotiation will be done explicitly (rather than implicitly as part of BGPSEC peering initiation).

5.3.2. Discussion

It was argued that if BGPSEC negotiation included negotiation of the larger max PDU size also, then it eliminates the need for checking a new error condition (regarding max PDU size). But then it was viewed as inadvisable to have two ways of doing something (i.e., implicit in BGPSEC and also as a separate negotiation capability). We decided that having the larger max PDU size will be a separate (explicit) capability negotiation.

5.4. Temporary Suspension of Attestations and Validations

5.4.1. Decision

A BGPSEC-capable router can temporarily suspend signing and/or validation of updates during periods of route processor overload. The router should later send signed updates corresponding to the updates for which validation and signing were skipped. The router also may choose to skip only validation but still sign and forward updates during periods of congestion.

5.4.2. Discussion

In some situations, a BGPSEC router may be unable to keep up with the workload of performing signing and/or validation. This can happen, for example, during BGP session recovery when a router has to send the entire routing table to a recovering router in a neighboring AS. So it is not mandatory that a BGPSEC router perform validation or signing of updates at all times. When the work load eases, the BGPSEC router should play catch up, sending signed updates corresponding to the updates for which validation and signing were skipped. During periods of overload, the router may simply send unsigned updates (with signatures dropped), or may sign and forward the updates with signatures (while the router itself has not yet validated the sigs it received).

6. Incremental Deployment and Negotiation of BGPSEC

6.1. Downgrade Attacks

6.1.1. Decision

No attempt will be made in BGPSEC design to prevent downgrade attacks, i.e., a BGPSEC-capable router sending unsigned updates when it is capable of sending signed updates.

6.1.2. Discussion

BGPSEC allows routers to temporarily suspend signing updates (see Section 5.4). Therefore, it would be contradictory if we were to try to incorporate in the BGPSEC protocol a way to detect and reject downgrade attacks. One proposed way for detecting downgrade attacks was considered, based on signed peering registrations (see Section 9.5).

6.2. Inclusion of Address Family in Capability Advertisement

6.2.1. Decision

It was decided that during capability negotiation, the address family for which the BGPSEC speaker is advertising support for BGPSEC will be shared using the Address Family Identifier (AFI). Initially, two address families would be included, namely, IPv4 and IPv6. BGPSEC for use with other address families may be specified in the future. Simultaneous use of the two (i.e., IPv4 and IPv6) address families for the same BGPSEC session will require that the BGPSEC speaker must include two instances of this capability (one for each address family) in the BGPSEC OPEN message.

6.2.2. Discussion

If new address families are supported in the future, they will be added in future versions of the specification. A comment was made that too many version numbers are bad for interoperability; Re-negotiation on the fly to add a new address family (i.e., without changeover to new version number) is desirable.

6.3. Incremental Deployment: Capability Negotiation

6.3.1. Decision

BGPSEC will be incrementally deployable. BGPSEC routers will use capability negotiation to agree to run BGPSEC between them. If a BGPSEC router's peer does not agree to run BGPSEC, then the BGPSEC router will run only BGP-4 with that peer, i.e., it will not send BGPSEC (i.e., signed) updates to the peer.

6.3.2. Discussion

During partial deployment, there will be BGPSEC islands as a result of this approach to incremental deployment. Updates that originate within a BGPSEC island will generally propagate with signed AS paths to the edges of that island.

An explicit capability negotiation (outside of the BGPSEC protocol initiation) will allow for negotiating a larger max PDU size (than the current 4KB) between BGPSEC peers (see Section 5.3).

6.4. Partial Path Signing

We discussed partial path signing which means that a BGPSEC AS can be permitted to sign an update that was received unsigned from a downstream neighbor. That is, the AS would add its ASN to the AS

path and sign the (previously unsigned) update to other neighboring (upstream) BGPSEC ASes. It was decided that this should not be permitted.

6.4.1. Decision

It was decided that partial path signing in BGPSEC will not be allowed. A BGPSEC update must be fully signed, i.e., each AS in the AS-PATH must sign the update. So in a signed update there must be a signature corresponding each AS in the AS path.

6.4.2. Discussion

Partial path signing (as described above) implies that the AS path is not rigorously protected. Rigorous AS path protection is a key requirement of BGPSEC [I-D.ietf-sidr-bgpsec-reqs]. Partial path signing clearly re-introduces the following attack vulnerability: If a BGPSEC speaker can sign an unsigned update, and if we presume that signed (i.e., partially or fully signed) updates will be preferred to unsigned updates, then a faulty, misconfigured or subverted BGPSEC speaker can manufacture any unsigned update it wants (with insertion of a valid origin AS) and add a signature to it to increase the chance that its update will be preferred.

6.5. Consideration of Stub ASes with Resource Constraints: Encouraging Early Adoption

6.5.1. Decision

The protocol permits each pair of BGPSEC-capable ASes to negotiate BGPSEC use asymmetrically. Thus a stub AS (or downstream customer AS) can agree to perform BGPSEC only in the transmit direction and speak BGP-4 in the receive direction. In this arrangement, the ISP's (upstream) AS will not send signed updates to this stub or customer AS. Thus the stub AS can avoid the need to upgrade its route processor and RIB memory to support BGPSEC update validation.

6.5.2. Discussion

Various other options were also considered for accommodating a resource-constrained stub AS:

1. An arrangement that can be effected outside of BGPSEC specification is as follows. Through a private arrangement (invisible to other ASes), an ISP's AS (upstream AS) can truncate the stub AS (or downstream AS) from the path and sign the update as if the prefix is originating from ISP's AS (even though the update originated unsigned from the customer AS). This way the

path will appear fully signed to the rest of the network. This alternative will require the owner of the prefix at the stub AS to issue a ROA for the upstream AS, so that the upstream AS is authorized to originate routes for said prefix.

2. Another type of arrangement that can also be effected outside of the BGPSEC specification is as follows. Stub AS does not sign updates but obtains an RPKI (CA) certificate, issues a router certificate under that CA certificate. It passes on the private key for the router certificate to its upstream provider. That ISP (i.e., the second hop AS) would insert a signature on behalf the stub AS using said private key obtained from the stub AS.

3. An extended ROA is created that includes the stub AS as the originator of the prefix and the upstream provider as the second hop AS, and partial signatures would be allowed (i.e., stub AS need not sign the updates). It is recognized that this approach is also authoritative and not trust based. It was observed that the extended ROA is not much different from what is done with ROA (in its current form) when a PI address is originated from a provider's AS. This approach was rejected due to possible complications with creation and use of a new RPKI object, namely, the extended ROA. Also, the validating BGPSEC router has to perform a level of indirection with approach, i.e., it has to detect if an update is not fully signed and then look for the extended ROA to validate.

4. Another method based on a different form of indirection would be as follows: Customer (stub) AS registers something like a Proxy Signer Authorization, which authorizes the second hop (i.e., provider) AS to sign on behalf of the customer AS using the provider's own key [Dynamics]. This method allows for fully signed updates (unlike the Extended ROA based approach). But this approach also requires the creation of a new RPKI object, namely, the Proxy Signer Authorization. In this approach the second hop AS has to perform a level of indirection. This approach was also rejected.

The various inputs regarding ISP preferences were taken into consideration, and eventually the decision in favor of asymmetric BGPSEC was reached. A stub AS that does asymmetric BGPSEC has the advantage that it needs to minimally upgrade to BGPSEC so it can sign updates in the transmit direction and can avoid the increased processing and memory burden needed to perform validations and to store signed updates in the RIBs.

6.6. Proxy Signing

6.6.1. Decision

An ISP's AS (or upstream AS) can proxy sign BGP announcements for a customer (downstream) AS provided that the customer AS obtains an RPKI (CA) certificate, issues a router certificate under that CA certificate, and it passes on the private key for that certificate to its upstream provider. That ISP (i.e., the second hop AS) would insert a signature on behalf the customer AS using the private key provided by the customer AS. This is a private arrangement between said parties and is invisible to other ASes. Thus, this arrangement is not part of the BGPSEC protocol specification

BGPSEC will not make any special provisions for an ISP to use its own private key to proxy sign updates for a customer's AS. This type of proxy signing is considered a bad idea.

6.6.2. Discussion

Consider a scenario when a customer's AS (say, AS8) is multi-homed to two ISPs, i.e., AS8 peers with AS1 and AS2 of ISP-1 and ISP-2, respectively. In this case AS8 would have an RPKI (CA) certificate; it issues two separate router certificates (corresponding to AS1 and AS2) under that CA certificate; and it passes on the respective private keys for those two certificates to its upstream providers AS1 and AS2. Thus AS8 has proxy signing service from both its upstream ASes. In the future, if the customer AS8 disconnects from ISP-2, then it would revoke the router certificate corresponding to AS2.

6.7. Multiple Peering Sessions Between ASes

6.7.1. Decision

No problems are anticipated when BGPSEC capable ASes have multiple peering sessions between them (between distinct routers).

6.7.2. Discussion

As with BGP-4 ASes, BGPSEC capable ASes can also have multiple peering sessions between them. Because routers in an AS (can) have distinct private keys, the same update when propagated over these multiple peering sessions will result in multiple updates that will differ in their signatures. The peer (upstream) AS will apply its normal procedures for selecting a best path from those multiple updates (and updates from other peers).

Multiple peering sessions, between different pairs of routers

(between two neighboring ASes), may be simultaneously used for load sharing. This decision regarding load balancing (vs. using one peering as primary for carrying data and another as backup) is entirely local and is up to the two neighboring ASes.

7. Interaction of BGPSEC with Common BGP Features

7.1. Peer Groups

In the current BGP-4, the idea of peer groups is used in BGP routers to save on processing when generating and sending updates. Multiple peers for whom the same policy apply can be organized into peer groups. A peer group can typically have tens (maybe as high as 300) of ASes in it.

7.1.1. Decision

It was decided that BGPSEC updates are generated to target unique AS peers, so there is no support for peer groups in BGPSEC.

7.1.2. Discussion

BGPSEC routers can use peer groups. Some of the update processing prior to forwarding to members of a peer group can be done only once per update as is done in BGP-4. Prior to forwarding the update, a BGPSEC speaker adds the peer’s ASN and signs the update for each peer AS in the group individually.

If updates were to be signed per peer group, that it would require divulging information about the forward AS-set that constitutes a peer group (since the ASN of each peer would have to be included in the update). Some ISPs do not like to share this kind of information globally.

7.2. Communities

We discussed whether there is a need to provide protection in BGPSEC for the community attribute.

7.2.1. Decision

Community attribute(s) will not be included in what is signed in BGPSEC.

7.2.2. Discussion

The community attribute - in its current definition - may be inherently defective, from a security standpoint. A substantial amount of work is needed on semantics of the community attribute, and additional work on its security aspects also needs to be done. The community attribute is not necessarily transitive; it is often used only between neighbors. In those contexts, transport security mechanisms suffice to provide integrity and authentication. (There is no need to sign data when it is passed only between peers.) It was suggested that one could include only the transitive community attributes in what is signed and propagated (across the AS path). It was noted that there is a flag available (i.e., unused) in the community attribute, and it might be used by BGPSEC (in some fashion). However, little information is available at this point about the use and function of this flag. It was speculated that potentially this flag could be used to indicate to BGPSEC if the community attribute needs protection. For now, community attributes will not be secured by BGPSEC path signatures.

7.3. Consideration of iBGP Speakers and Confederations

7.3.1. Decision

An iBGP speaker that is also an eBGP speaker, and that executes BGPSEC, will necessarily carry BGPSEC data and perform eBGPSEC functions. Confederations are eBGP clouds for administrative purposes and contain multiple sub-ASs. A sub-AS is not required sign updates sent to the main AS; only the main AS will sign and propagate BGPSEC updates to eBGPSEC peer ASes.

If updates are not signed (i.e., BGPSEC is not used) within a confederation boundary, then everything will work fine at a BGPSEC speaker in the confederation that is executing BGPSEC with external peers. If updates are signed (i.e., BGPSEC is used) within a confederation boundary, then the BGPSEC speaker will be required to remove any signatures applied within the confederation, and replace them with a single signature representing the (main) AS, which will be appropriate for external BGPSEC peers. The BGPSEC specification will not specify how to perform this process.

7.3.2. Discussion

This topic may need to be revisited to flesh out the details carefully.

7.4. Consideration of Route Servers in IXPs

7.4.1. Decision

BGPSEC makes no special provisions to accommodate route servers in Internet Exchange Points (IXPs) .

7.4.2. Discussion

There are basically three methods that an IXP may use to propagate routes: (A) Direct bilateral peering through the IXP, (B) BGP peering between clients via a peering with a route server at the IXP (without IXP inserting its ASN in the path), and (C) BGP peering with an IXP route server, where the IXP inserts its ASN in the path. (Note: IXP's route server does not change the NEXT_HOP attribute even if it inserts its ASN in the path.) It is very rare for an IXP to use Method C because it is less attractive for the clients if their AS path length increases by one due to the IXP. A measure of the extent of use of Method A vs. Method B is given in terms of the corresponding IP traffic load percentages. As an example, at a major European IXP, these percentages are about 80% and 20% for Methods A and B, respectively. However, as the IXP grows (in terms of number of clients), it tends to migrate more towards Method B, because of the difficulties of managing up to n x (n-1)/2 direct inter-connections between n peers in Method A.

To the extent an IXP is providing direct bilateral peering between clients (Method A), that model works naturally with BGPSEC. Also, if the route server in the IXP plays the role of a regular BGPSEC speaker (minus the routing part for payload) and inserts its own ASN in the path (Method C), then that model would also work well in the BGPSEC Internet. However, the initial version of BGPSEC does not accommodate the "transparent" route server model of Method B.

7.5. Proxy Aggregation (a.k.a. AS_SETs)

7.5.1. Decision

Proxy aggregation (i.e., use of AS_SETs in the AS path) will not be supported in BGPSEC. That is to say that there is no provision in BGPSEC to sign an update when an AS_SET is part of an AS path. If a BGPSEC capable router receives an update that contains an AS_SET and also finds that the update is signed, then the router will strip the signatures and interpret the update as unsigned. If the update (with AS_SET) is selected as best path, it will be forwarded unsigned.

7.5.2. Discussion

Proxy aggregation does occur in the Internet today, but is it very rare. Only a very small fraction (about 0.1%) of observed updates contain AS_SETs in the AS path. Since BGP-4 currently allows for proxy aggregation with inclusion of AS_SETs in the AS path, it is necessary that BGPSEC specify what action a receiving router must take in case such an update is received with attestation. The IDR WG has approved a document that deprecates use of AS_SETs [I-D.ietf-idr-deprecate-as-sets], so we anticipate that use of AS_SETs will diminish over time.

7.6. 4-Byte AS Numbers

Not all (currently deployed) BGP speakers are capable of dealing with 4-byte ASNs [RFC4893]. The standard mechanism used to accommodate such speakers requires a peer AS to translate each 4-bye ASN in a path into a reserved 2-byte ASN before forwarding the update. This mechanism is incompatible with use of BGPSEC, since the ASN translation is equivalent to a route modification attack.

7.6.1. Decision

BGP speakers that are BGPSEC-capable are required to process 4-byte ASNs.

7.6.2. Discussion

It is reasonable to assume that upgrades for 4-byte ASN support will be in place prior to deployment of BGPSEC.

8. BGPSEC Validation

8.1. Sequence of BGPSEC Validation Processing in a Receiver

It is natural to ask in what sequence a receiver must perform BGPSEC update validation so that if a failure were to occur (i.e., update was determined to be invalid) the processor would have spent the least amount of processing or other resources.

8.1.1. Decision

There was agreement that the following sequence of receiver operations is quite meaningful, and will be included in the BGPSEC specification [I-D.ietf-sidr-bgpsec-protocol]. However, the ordering of validation processing steps is not a normative part of the BGPSEC specification.

1. Verify that the signed update is syntactically correct. For example, check if the number of sigs match with the number of ASes in the AS path (after duly accounting for AS prepending).

2. Verify that the origin AS is authorized to advertise the prefix in question. This verification is based on data from ROAs, and does not require any crypto operations.

3. Verify that the advertisement has not yet expired.

4. Verify that the target ASN in the signature data matches the ASN of the router that is processing the advertisement. Note that the target ASN check is also a non-crypto operation and is fast. It is suggested that signature data be checked from the most recent AS to the origin.

5. Locate the public key for the router from which the advertisement was received, using the SKI from the signature data.

6. Hash the data covered by the signature algorithm. Invoke the signature validation algorithm on the following three inputs: the locally computed hash, the received signature, and the public key. There will be one output: valid or invalid.

7. Repeat steps 5 and 6 for each preceding signature in the Signature-List Block, until the signature data for the origin AS is encountered and processed, or until either of these steps fails.

8.1.2. Discussion

The suggested sequence of receiver operations described above were discussed and are viewed as appropriate, if the goal is to minimize computational costs associated with cryptographic operations. One additional interesting suggestion was that when there are two Signature-List Blocks in an update, the validating router can first verify whichever of the two algorithms is cheaper to save on processing. If that Signature-List Block verifies, then the router can skip validating the other Signature- List Block. Of course, at the end of an algorithm transition period, many routers would support only the new algorithm because their old credentials would have expired.

8.2. Signing and Forwarding Updates when Signatures Failed Validation

8.2.1. Decision

A BGPSEC router should sign and forward a signed update to upstream peers if it selected the update as the best path, regardless of whether the update passed or failed validation (at this router). (Note: The BGPSEC protocol specification or a companion BCP may later specify some conditions of failed update validation (TBD) under which a BGPSEC router must not select the AS path in the update.)

8.2.2. Discussion

The availability of RPKI data at different routers (in the same or different ASes) may differ, depending on the sources used to acquire RPKI data. Hence an update may fail validation in one AS and the same update may pass validation in another AS. Thus an update may fail validation at one router in an AS and the same update may pass validation at another router in the same AS. A BCP may be published later in which some conditions of update failure are identified which may be unambiguous cases for rejecting the update, in which case the router must not select the AS path in the update. These cases are TBD.

8.3. Enumeration of Error Conditions

Enumeration of error conditions and the recommendations for reactions to them are still under discussion.

8.3.1. Decision

TBD. Also, please see Section 8.5 for the decision and discussion specifically related to syntactic errors in signatures.

8.3.2. Discussion

The list here is a first cut at some possible error conditions and recommended receiver reactions in response to detection of those errors. Refinements will follow after further discussions.

E1 Abnormalities that a peer (i.e., preceding AS) should definitely not have propagated to a receiving eBGPSEC router. Examples: (A) The number of signatures does not match the number of ASes in the AS path (after accounting for AS prepending); (B) There is an AS_SET in the received update and the update has signatures; (C) Other syntactic errors with sigs.

Reaction: See Section 8.5.

E2 Situations where a receiving eBGPSEC router can't find the cert for an AS in the AS_PATH.

Reaction: Mark the update as "Invalid". It is acceptable to consider the update in best path selection. If it is chosen, then the router should sign and propagate the update.

E3 Situations where a receiving eBGPSEC router can't find a ROA for the {prefix, origin} pair.

Reaction: Same as in (E2) above.

E4 The receiving eBGPSEC router verifies signatures and finds that the update is Invalid even though its peer might not have known (e.g., due to RPKI skew).

Reaction: Same as in (E2) above.

Note: Best route choice may involve choosing an unsigned update over one with "Invalid" signature(s). Hence, the signatures must not be stripped even if the update is "Invalid". No evil bit is set in the update (when it is Invalid) because an upstream peer may not get that same answer when it tries to validate.

8.4. Procedure for Processing Unsigned Updates

An update may come in unsigned from an eBGP peer or internally (e.g., as an iBGP update). In the latter case, the route is possibly being originated from within the AS in consideration, or from within an AS confederation.

8.4.1. Decision

If an unsigned route is received from an eBGP peer, and if it is selected, then the route will be forwarded unsigned to other eBGP peers, even BGPSEC-capable peers. If the route originated in this AS (IGP or iBGP) and is unsigned, then it should be signed and announced to external BGPSEC-capable peers. If the route originated in IGP (or iBGP) and is signed, then it was likely signed by ASes within a confederation. In this case, signatures from within the confederation would be processed and they would be deleted, and an origin AS signature will be added prior to announcement to eBGP (BGPSEC capable) peers (also see Section 7.3).

8.4.2. Discussion

There is also a possibility that an update received in IGP (or iBGP) may have private ASNs in the AS path. These private ASNs would

normally appear in the right most portion of the AS path. It was noted that in this case, the private ASNs to the right would be removed (as done in BGP-4 currently?), and then the update will be signed by the originating AS and announced to eBGP (BGPSEC capable) peers.

8.5. Response to Syntactic Errors in Signatures and Recommendation for Reaction

Different types of error conditions were discussed in Section 8.3. Here we focus only on syntactic error conditions in signatures.

8.5.1. Decision

If there are syntactic error conditions such as (a) AS_SET and Signature-List Block both appear in an update, or (b) the number of signatures does not match the number of ASes (after accounting for any AS prepending), or (c) a parsing issue occurs with the BGPSEC_Path_Signatures attribute, then the update (with the signatures stripped) will still be considered in the best path selection algorithm. If the update is selected as the best path, then the update will be propagated unsigned. The error condition will be logged locally.

A BGPSEC router will follow whatever the current IETF (IDR WG) recommendations are for notifying a peer that it is sending malformed messages.

In the case when there are two Signature-List Blocks in an update, and one or more syntactic errors are found to occur within one of the Signature-List Blocks but the other Signature-List Block is free of any syntactic errors, then the update will still be considered in the best path selection algorithm after the syntactically bad Signature-List Block has been removed. If the update is selected as the best path, then the update will be propagated with only one (i.e., the error-free) Signature-List Block. The error condition will be logged locally.

8.5.2. Discussion

As stated above, a BGPSEC router will follow whatever the current IETF (IDR WG) recommendations are for notifying a peer that it is sending malformed messages. Question: If the error is persistent, and there is a full BGP table dump occurring, then would there be 500K such errors resulting in 500K notify messages sent to the erring peer? The answer was that rate limiting would be applied to the notify messages which should prevent any overload due to these messages.

8.6. Enumeration of Validation States

Various validation conditions (i.e., situations) are possible which can be mapped to validation states for possible input to BGPSEC decision process. These conditions can be related to whether or not an update is signed, Expire Time checked, AS origin validation checked against a ROA, signatures verification passed, etc.

8.6.1. Decision

It was decided that BGPSEC validation outcomes will be mapped to one of only two validation states: (1) Valid - passed all validation checks (i.e., Expire Time check, prefix-origin and Signature-List Block validation), and (2) Invalid - all other possibilities.

It was decided subsequently that the terms "Valid" and "Invalid" will be generally not used in the context of update validation in BGPSEC. Instead the terms "Verified" and "Unverified" will be used. The term "Verified" would connote the same as "Valid" described above. The term "Unverified" would include all other situations such as (1) unverified due to lack of or insufficient RPKI data, (2) signature Expire-Time check failed, (3) prefix-origin validation failed, (4) signature checks were performed and one or more of them failed, (5) insufficient resources to process the signature blocks at this time, etc.

The text in this document will be modified at a future date to consistently reflect this decision regarding the terminology change. For now we continue to use the terms "Valid" and "Invalid" in the document.

8.6.2. Discussion

It may be noted that the result of update validation is just an additional input for the BGP decision process. The router configuration ultimately has control over what action (regarding BGP path selection) is taken.

Initially, we had considered four validation states: (1) Update is not signed; (2) Update is signed but router does not have corresponding RPKI data to perform validation check; (3) Invalid (validation check performed and failed); (4) Valid (validation check performed and passed). Later, it was decided that BGPSEC validation outcomes will be mapped to one of only two validation states as stated above. It was observed that an update can be invalid for many different reasons. To begin to differentiate these numerous reasons and to try to enumerate different flavors of the Invalid state is not likely to be constructive in route selection decision, and may even

introduce to new vulnerability in the system. However, some questions remain such as the following.

Question: Is there a need to define a separate validation state for the case when update is not signed but {prefix, origin} pair matched with ROA information? This question was discussed, and a tentative conclusion was that this is in principle similar to validation based on partial signatures, which we have decided against. So there is no need to add another validation state for this case; treat it as "Unverified" (i.e., "Invalid"). Questions still remain, e.g., would the relying party want to give said update a higher preference over another unsigned update that failed ROA validation or over a signed update that failed both signature and ROA validation?

8.7. Mechanism for Transporting Validation State through iBGP

8.7.1. Decision

BGPSEC validation need be performed only at eBGP edges. 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 will determine the means by which the validation status is conveyed internally, using various pre-existing mechanisms, e.g., setting a BGP community, or modifying a metric value such as Local_Pref or MED. A signed update that cannot be validated (except those with syntax errors) should be forwarded with signatures from the ingress to the egress router, where it is signed when propagated towards other eBGPSEC speakers in neighboring ASs. 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. The latter approach may be used at an operator’s discretion, under circumstances when RPKI skew is known to happen at different routers within an AS.

8.7.2. Discussion

The attribute used to represent the validation state can be carried between ASes if desired. ISPs may like to carry it over their eBGP links between their own ASes (e.g., AS701, AS702). A peer (or customer) may receive it over an eBGP link from a provider, and may want to use it to shortcut their own validation check. However, the peer (or customer) should be aware that this validation-state attribute is just a preview of a neighbor’s validation and must perform their own validation check in order to be sure of the actual state of update’s validation. Question: Do we want this validation state propagation to be protected by attestation in case it has utility for diagnostics purposes? It was decided not to protect the validation state information using signatures.

The following are meant to be only as suggestions for the AS operator; none of what follows is part of the BGPSEC specification as such.

The following Validation states may be needed for propagation via iBGP between edge routers in an AS:

- Validation states communicated in iBGP for an unsigned update (Origin validation result): (1) Valid, (2) Invalid, (3) Unknown, (4) Validation Deferred.
  - An update could be unsigned for two reasons but they need not be distinguished: (a) Because it had no signatures (came in unsigned from an eBGP peer), or (b) Signatures were present but stripped due to syntax errors.
- Validation states communicated in iBGP for a Signed update: (1) Valid, (2) Invalid, (3) Validation Deferred.

The reason for conveying the additional "Validation Deferred" state may be stated as follows. An ingress edge Router A receiving an update from an eBGPSEC peer may not attempt to validate signatures (e.g., in a processor overload situation), and in that case Router A should convey "Validation Deferred" state for that signed update (if selected for best path) in iBGP to other edge routers. Then an egress edge Router B upon receiving the update from ingress Router A would be able to perform its own validation (origin validation for unsigned or signature validation for signed update). As stated before, the egress Router B always may choose to perform its own validation when it receives an update from iBGP (independent of the validation status conveyed in iBGP) to account for the possibility of RPKI data skew at different routers. These various choices are local and entirely up to operator discretion.

9. Operational Considerations

9.1. Interworking with BGP Graceful Restart

BGP Graceful Restart (BGP-GR) [RFC4724] is a mechanism currently used to facilitate non-stop packet forwarding when the control plane is recovering from a fault (i.e., BGP session is restarted), but the data plane is functioning. A question was asked regarding if there are any special concerns about how BGP-GR works while BGPSEC is operational? Also, what happens if the BGP router operation transitions from BGP-4 to BGP-GR to BGPSEC, in that order?

9.1.1. Decision

No decision was made relative to this issue.

9.1.2. Discussion

BGP-GR can be implemented with BGPSEC just as it is currently implemented with BGP-4. The Restart State bit, Forwarding State bit, End-of-RIB marker, Staleness marker (in RIB-in), and Selection_Deferral_Timer are key parameters associated with BGP-GR [RFC4724]. These parameters would need to be incorporated into the BGPSEC session negotiation and/or operation just as the routers do now with the current BGP-4.

Regarding what happens if the BGP router transitions from BGP-4 to BGP-GR to BGPSEC, the answer would simply be as follows. If there is software upgrade from BGP-4 to BGPSEC during BGP-GR (assuming upgrade is being done on a live BGP speaker), then the BGP-GR session would (should) be terminated before a BGPSEC session is initiated. Once the eBGPSEC peering session is established, then the receiving eBGPSEC speaker will see signed updates from the sending (newly upgraded) eBGPSEC speaker. There is no apparent harm (it may, in fact, be desirable) if the receiving speaker continues to use previously-learned BGP-4 routes from the sending speaker until they are replaced by new BGPSEC routes. However, if the Forwarding State bit is set to zero by the sending speaker (i.e., the newly upgraded speaker) during BGPSEC session negotiation, then the receiving speaker would mark all previously-learned BGP-4 routes from that sending speaker as "Stale" in its RIB-in. Then, as fresh BGPSEC updates (possibly mixed with some unsigned BGP-4 updates) come in, the "Stale" routes will be replaced or refreshed.

9.2. BCP Recommendations for Minimizing Churn: Certificate Expiry/ Revocation and Signature Expire Time

9.2.1. Decision

This is still work in progress.

9.2.2. Discussion

BCP recommendations for minimizing churn in BGPSEC have been discussed. We have discussed various strategies on how routers should react in the events of certificate expiry/revocation and signature Expire Time exhaustion [Dynamics]. The details will be documented in the near future after additional work is completed.

9.3. Outsourcing Update Validation

9.3.1. Decision

Update signature validation and signing can be outsourced to an off-board server or processor.

9.3.2. Discussion

We can potentially have an off-router box (one or more per AS) that can be performing path validation. For example, these capabilities might be incorporated into a route reflector. At ingress, one needs the RIB-in entries validated; not the RIB-out entries. So the off-router box is probably unlike the traditional route reflector; it sits at net edge and validates all incoming BGPSEC updates. Thus it appears that each router passes each BGPSEC update it receives to the off-router box and receives a validation result before it stores the route in the RIB-in. Question: What about failure modes here? They would be dependent on (1) How much of the control plane is outsourced; (2) Reliability of the off-router box (or, equivalently communication to it); and (3) How centralized vs. distributed is this arrangement? When any kind of outsourcing is done, the user needs to be watchful and ensure that the outsourcing does not cross trust/ security boundaries.

9.4. New Hardware Capability

9.4.1. Decision

It is assumed that BGPSEC routers (PE routers and route reflectors) will have significantly upgraded hardware - much more memory for RIBs and hardware crypto assistance. However, stub ASes would not need to make such upgrades because they can negotiate asymmetric BGPSEC capability with their upstream ASes, i.e., they sign updates to the upstream AS but receive only BGP-4 (unsigned) updates (see Section 6.5).

9.4.2. Discussion

It is accepted that it might take several years to go beyond test deployment, because of the need for additional memory and processing capability. However, because BGPSEC deployment will be incremental, and because signed updates are not sent outside of a set of contiguous BGPSEC-enabled ASes, it is not clear how much additional (RIB) memory will be required during initial deployment. See (see [RIB_size]) for preliminary results on modeling and estimation of BGPSEC RIB size and its projected growth. Hardware cryptographic support reduces the computation burden on the route processor, and

offers good security for router private keys. However, given the incremental deployment model, it also is not clear how substantial a cryptographic processing load will be incurred, initially.

9.5. Signed Peering Registrations

9.5.1. Decision

The idea of signed BGP peering registrations (for the purpose of path validation) was rejected.

9.5.2. Discussion

The idea of using a secure map of AS relationships to "validate" updates was discussed and rejected. The reason for not pursuing such solutions was that they can’t provide strong guarantees about the validity of updates. Using these techniques, one can say only that an update is ’plausible’, but cannot say it is ’definitely’ valid (based on signed peering relations alone).

10. Co-authors

Rob Austein sra@hactrn.net
Internet Systems Consortium

Steven Bellovin smb@cs.columbia.edu
Columbia University

Randy Bush randy@psg.com
Internet Initiative Japan, Inc.

Russ Housley housley@vigilsec.com
Vigil Security

Stephen Kent kent@bbn.com
BBN Technologies

Warren Kumari warren@kumari.net
Google

Matt Lepinski mlepinsk@bbn.com
BBN Technologies

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

Samuel Weiler weiler@watson.org
Cobham

11. 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.

12. IANA Considerations

This memo includes no request to IANA.

13. Security Considerations

This memo requires no security considerations. See [I-D.ietf-sidr-bgpsec-protocol] for security considerations for the BGPSEC protocol.

14. References

14.1. Normative References

[CiscoIOS]
"Cisco IOS RFD implementation", <http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html#wp1002395>.

[I-D.draft-clynn-s-bgp]
Lynn, C., Mukkelson, J., and K. Seo, "Secure BGP (S-BGP)", June 2003, <http://tools.ietf.org/html/draft-clynn-s-bgp-protocol-01>.

[I-D.ietf-idr-bgp-extended-messages]
Patel, K., Ward, D., and R. Bush, "Extended Message support for BGP", draft-ietf-idr-bgp-extended-messages-01 (work in progress), June 2011.

[I-D.ietf-idr-deprecate-as-sets]
Kumari, W. and K. Sriram, "Deprecation of the use of BGP AS_SET, AS_CONFED_SET.",

draft-ietf-idr-deprecate-as-sets-04 (work in progress), May 2011.

[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-bgpsec-overview]
Lepinski, M. and S. Turner, "An Overview of BGPSEC", draft-ietf-sidr-bgpsec-overview-00 (work in progress), June 2011.

[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-bgpsec-reqs]
Bellovin, S., Bush, R., and D. Ward, "Security Requirements for BGP Path Validation", draft-ietf-sidr-bgpsec-reqs-00 (work in progress), June 2011.

[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.

[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-01 (work in progress), February 2011.

[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-22 (work in progress), May 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-roa-validation]
Huston, G. and G. Michaelson, "Validation of Route

Origination using the Resource Certificate PKI and ROAs", draft-ietf-sidr-roa-validation-10 (work in progress), November 2010.

[JunOS] "Juniper JunOS RFD implementation", <http://www.juniper.net/techpubs/en_US/junos10.4/topics/usage-guidelines/policy-using-routing-policies-to-damp-bgp-route-flapping.html>.

[Mao02] Mao, Z. and et al., "Route-flap Damping Exacerbates Internet Routing Convergence", August 2002, <http://www.eecs.umich.edu/~zmao/Papers/sig02.pdf>.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route Flap Damping", RFC 2439, November 1998.

[RFC2673] Crawford, M., "Binary Labels in the Domain Name System", RFC 2673, August 1999.

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

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, June 2005.

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

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, January 2007.

[RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS Number Space", RFC 4893, May 2007.

[RFC5396] Huston, G. and G. Michaelson, "Textual Representation of Autonomous System (AS) Numbers", RFC 5396, December 2008.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

[RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic Curve Cryptography Algorithms", RFC 6090, February 2011.

[RIPE378] Smith, P. and C. Panigl, "RIPE-378: RIPE Routing Working Group Recommendations on Route-flap Damping", <http://www.ripe.net/docs/ripe-378.html>.

14.2. Informative References

[Dynamics] Sriram, K., "Potential Impact of BGPSEC Mechanisms on Global BGP Dynamics", December 2009, <Work in progress, Presentation slides available on request.>.

[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.

[RIB_size] Sriram, K., "RIB Size Estimation for BGPSEC", June 2011, < http://www.antd.nist.gov/~ksriram/ BGPSEC_RIB_Estimation.pdf>.

Author's Address

Kotikalapudi Sriram (editor)
USA National Institute of Standards and Technology (NIST)
100 Bureau Drive
Gaithersburg, MD 20899
USA

Email: ksriram@nist.gov

INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: January 3, 2012

Xia Yin
Tsinghua Univ.
Yang Xiang
Tsinghua Univ.
Zhiliang Wang
Tsinghua Univ.
Jianping Wu
Tsinghua Univ.
July 2, 2011

Efficient Secure BGP AS Path using FS-BGP
draft-xia-sidr-fsbgp-00.txt

Abstract

This draft proposes Fast Secure BGP (FS-BGP), an efficient mechanism for securing AS paths and preventing prefix hijacking by signing critical AS path segments (i.e., adjacent AS triples). FS-BGP can achieve similar level of security as S-BGP, but with much higher efficiency.

Status of this Memo

Table of Contents

1. Introduction

In order to improve the security of BGP, several extensions have been proposed, which fall into two categories: anomaly detection and cryptographic based authentication. However, anomaly detection approaches [Whisper] [PGBGP] can not guarantee security and correctness. Cryptographic approaches, which are being pursued by the SIDR WG, use the Public Key Infrastructure (PKI) to authenticate routing announcements. There are a bunch of solutions including S-BGP [S-BGP] [I-D.lepinski-bgpsec-protocol] and many others. However, S-BGP may consume significant resources of computation and storage. The other solutions either compromise in the security [IRV] [I-D. ng-sobgp-bgp-extensions] [psBGP] [SPV], or bring in more complexity on certification distribution [SA].

Towards these unsolved issues, we propose an efficient approach, FS-BGP (Fast Secure BGP), to secure AS path. Through signing critical AS path segments (i.e., adjacent AS triples), FS-BGP can achieve similar level of security as S-BGP, but with much higher efficiency. Analysis, evaluations, and more discussions can be found in our recent technical report [TR-FSBGP].

2. Terminology

(i):	AS i
<n, ..., 0>:	AS path from AS n to the origin AS 0
<n, ..., 0>f:	AS path of prefix f
<i+1, i, i-1>:	critical AS path segment, adjacent AS triple in a path
<1, 0, f>:	origin critical AS path segment in a path of prefix f
{msg}i:	signature on msg generated by AS i

3. Background

In BGP, UPDATE messages can not be validated, so neither the origin AS nor the AS path is guaranteed to be correct. Secure BGP (S-BGP) [SBGP] is the dominant solution to this problem, and it uses a PKI to help authenticating involved parties and messages. Specifically, S-BGP uses Route Attestations (RAs) for path authentication.

As shown in Figure 1, a RA is all signatures signed by ASes along the path to authenticate the existence and position of ASes in the path. We define {msg}i as the signature on msg generated with AS i’s private key. In Figure 1, each AS i equivalently signs the corresponding extended AS path <i+1, i, ..., 0> and the prefix f. The inclusion of the recipient AS i+1 in each signature is necessary to prevent cut-and-paste attack.

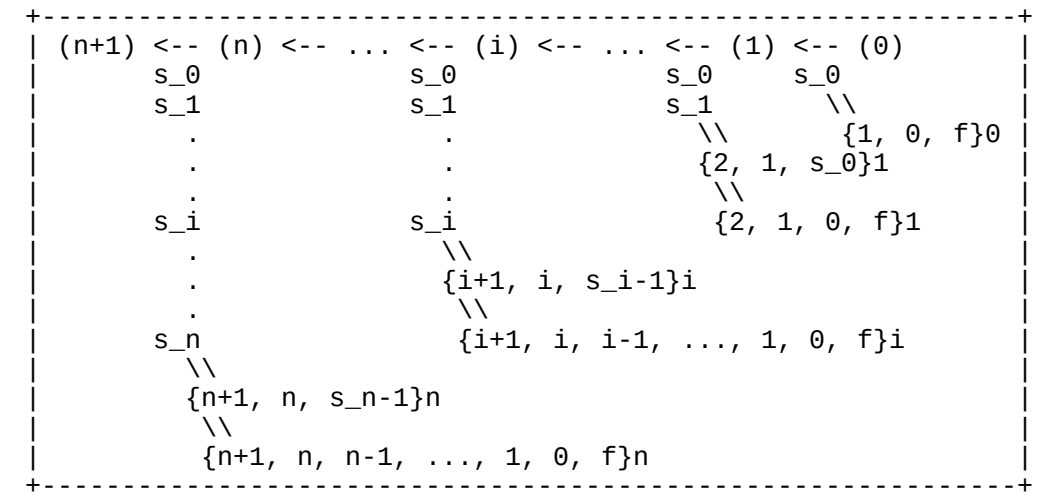


Figure 1. RAs in S-BGP.

The main concern about deploying S-BGP in practice is the huge computational cost for signing and verifying signatures. The dominating barrier for adopting S-BGP is the overhead of processing RAs, that is to authenticate paths. Toward this direction, there are a bunch of solutions for reducing the overhead of path authentication.

soBGP [I-D.ng-sobgp-bgp-extensions] maintains all authenticated AS edges in a database, but faces the problem of forged paths. IRV [IRV] builds an authentication server in each AS, but brings the problem of maintaining and inter-connecting these servers, and introduces query latencies. SPV [SPV] accelerates the signing process by pre-generated one-time signatures based on a single root value, but involves a significant amount of state information, and its security can only be guaranteed probabilistically. Signature Amortization (S-A) [SA] uses one bit vector for each neighbor of an AS to indicate the allowed recipients of a route, such that only one signing is needed for multiple recipients. However, each AS will need to pre-establish a neighbor list corresponding to the bit vector, and to distribute it to all other ASes.

As we can see, existing methods usually compromise security, and most of them only improve the performance of signing. However, verification happens more frequently than signing, since one signature often needs to be verified at multiple places.

According to the above analysis, it is important to design an efficient method to secure AS paths. Our solution, FS-BGP, builds on the assumption that a PKI is ready for use, and focuses on AS path

authentication.

4. Secure Feasible AS Paths

S-BGP can not prevent replay of outdated routes. It can only use expiration-date to roughly control the window of exposure to replay attack. As a result, though it only signs currently announcing path, it actually authenticates all announced feasible paths. Under a stable AS-level topology, we call a path feasible when the path satisfies the import and export policies of all ASes along the path.

Since failures often occur in the global routing system, many feasible paths can be easily announced and become authenticated. Thus, if a protocol can guarantee that all authenticated paths are feasible path, then it can achieve similar level of security as S-BGP. So we wonder that is it possible to efficiently secure feasible paths but not blindly sign every currently announcing path.

BGP is a policy-based routing protocol. An AS only exports a route to a neighbor if it is willing to forward traffic to the corresponding prefix from that neighbor. Although complex policies (i.e., route filters [RFC2622]) exist, AS usually does not differentiate between prefixes or nonadjacent ASes. For example, in Figure 2, when AS n decides whether routes learned from AS n-1 can be exported to AS n+1, it only considers its relation with the two neighbors, but does not consider other ASes along the path (<n-2, ..., 1, 0>). We call this the Neighbor Based Importing and Exporting (NBIE).

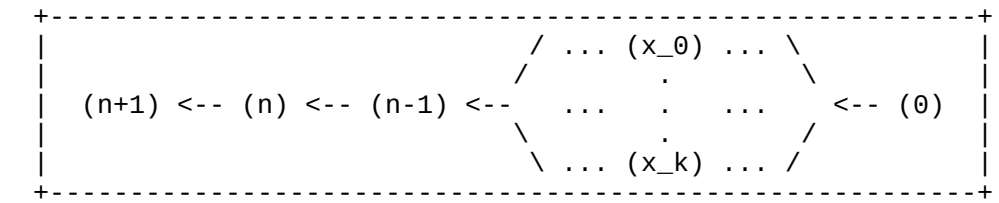


Figure 2. In S-BGP, AS n signs k paths which share a mutual AS path segment <n+1, n, n-1>.

NBIE abstracts the basic functionality of BGP. According to our measurement results in whois database, only a small portion of routing polices (route filters) violate the NBIE assumption. Nevertheless, the purpose of route filters is to protect the routing system against distribution of inaccurate routing information [RFC2622]. In other words, the use of route filters is mainly due to security considerations rather than policy requirements. We believe that under a security environment (i.e., FS-BGP or S-BGP), these filters are not needed any more. In deed, our schema can flexibly

support complicated routing polices [TR-FSBGP].

5. FS-BGP: Fast Secure BGP

5.1. Signing Critical AS Path Segments

Following our key observation above, we propose Fast Secure BGP (FS-BGP) to grantee the authentication of feasible paths. Given a feasible path p=<n+1, n, ..., 0>, we define its set of critical path segments as c_i, 0<i<=n, where

$$c_i = \begin{cases} <1,0,f> & \text{, for } i=0 \\ <i+1,i,i-1> & \text{, for } 0<i<=n \end{cases}$$

We call AS i the owner of c_i. Particularly, c_0 is called the originating critical path segment owned by AS 0. A critical path segment <i+1, i, i-1> actually describes an routing export policy of its owner AS i, and implies that AS i can export all routes imported from AS i-1 to AS i+1.

More specifically, FS-BGP uses Critical Segment Attestations (CSA) to authenticate paths. A CSA is simply the signature of the critical path segment signed by its owner. In a path p=<n+1, n, ..., 0>, the CSA s_i signed by AS i is defined as:

$$s_i = \begin{cases} \{1,0,f\}0 & \text{, for } i=0 \\ \{i+1,i,i-1\}i & \text{, for } 0<i<=n \end{cases}$$

The inclusion of the prefixes f in s_0 is necessary, because AS 0 might be multi-homing and only announces part of its prefixes to AS 1. Figure 3 and Figure 1 compare the signatures in FS-BGP and S-BGP. Obviously, the number of distinct critical path segments is far less than the number of distinct paths. As a result, the number of signing and verification operations in FS-BGP can be greatly reduced, after using a small cache. In Figure 2, AS n needs to sign each of the k paths individually in S-BGP. However, in FS-BGP, all the k different paths can reuse one signature of the common critical segment <n+1, n, n-1>.

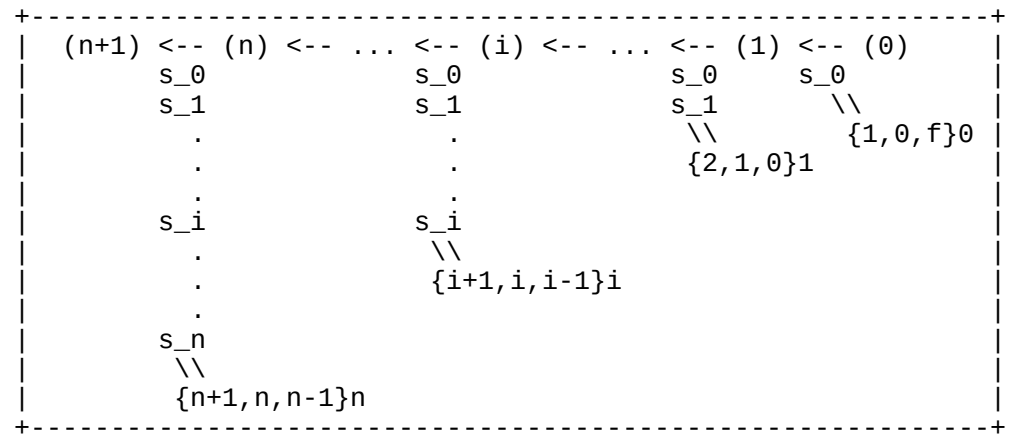


Figure 3. CSAs in FS-BGP.

We argue that, under the NBIE rule, if every AS along a path signs its critical path segment, then the path can be authenticated as a feasible path [TR-FSBGP]. However, since not all feasible paths are actually announced, it is possible to forge a path if the security mechanism relies on CSA only, as shown in Section 5.2. We will provide effective solution to this problem.

5.2. Prevent Effective Hijacking in FS-BGP

In FS-BGP, an AS using FS-BGP can forge paths that are not actually announced by others, but avoids CSA based detection. Forged paths can be constructed by concatenating critical segments, and used for prefix hijacking.

Although forging paths in FS-BGP is possible, there are still some restrictions on how paths can be forged. First, a path can only be forged by combining non-forged paths which share mutual segments. Second, some part of a forged path must be treated as sub-optimal or suppressed by some AS along the path. Third, forged paths are still feasible, and can only be used for prefixes associated with the originating critical segment in the path. Last, forged paths can not be very short [TR-FSBGP].

Although there are limitations on forged paths, prefix hijacking is still possible. In this section, we discuss solutions to prevent prefix hijacking. We only concern effective hijacking, in a sense that, the recipient of a forged route indeed changes its forwarding path.

Firstly, we divide all feasible paths into three categories:

Optimal path: the best path that passes all the decision steps in BGP.

Sub-optimal path: paths with the same Local Preference as the optimal path, but not chosen as the best one.

Suppressed path: paths with lower Local Preferences than the optimal and sub-optimal paths. For example, paths that are more expensive (i.e., through a provider), are often suppressed by a low preference.

We argue that, if a forged path is no shorter than the non-forged path BGP should announce, it can not be used for effective hijacking [TR-FSBGP]. Under a stable AS-level topology, a router will use its optimal path for every prefix. If BGP is purely a shortest path routing protocol (optimal path is always the shortest one), manipulator can not effectively hijack any prefix by forging paths. However, policy routing makes hijacking possible.

We know only suppressed path can be shorter than the optimal path (since a sub-optimal path has the same local preference as the optimal path, its length can not be shorter). Thus, if there is a mechanism to guarantee that all suppressed paths are no shorter than their corresponding optimal paths, manipulator can no longer effectively hijack a prefix either. This idea can be implemented by using AS Path Pre-pending (ASPP).

We call such a mechanism Suppressed Path Padding (SPP), and Figure 4 depicts the pseudo code for deciding how many times an AS i should pad itself in a path. If a path is imported from a neighbor AS i-1 with the highest local preference, AS i only appears once (line 1 and 2). Otherwise, the number of occurrences k_i must be large enough such that no suppressed path can be shorter than the corresponding optimal path. Given a path p, denote the optimal path to the same prefix as p by opt(p), then k_i is set as the largest Path Length difference between any suppressed path p imported from this neighbor and the corresponding opt(p) (line 4 to 7).

```
+-----------------------------------------------------------+
| Algorithm: Suppressed Path Padding                        |
| INPUT:  local AS i, neighbor AS i-1                       |
| OUTPUT: k_i: number of times that AS i needs to be padded |
|         in the paths import from AS i-1                   |
| 1:  IF AS i-1 has the highest local preference THEN       |
| 2:      RETURN 1                                          |
| 3:  k_i <- 1                                              |
| 4:  FOR ALL path p imported from AS i-1 DO                |
| 5:      opt(p) <- the optimal path corresponding to p     |
| 6:      IF length(p) - length(opt(p)) > k_i THEN          |
| 7:          k_i <- length(p) - length(opt(p))             |
| 8:  RETURN k_i                                            |
+-----------------------------------------------------------+
```

Figure 4. SPP (Suppressed Path Padding).

It is worth noting that, SPP is quite general. When necessary, it can and also should be used even in S-BGP. Consider the case when the optimal route fails. At this time, S-BGP will announce a previously sub-optimal or suppressed path temporarily, and this path can be used later by the manipulator to launch an effective attack, if it is short enough. S-BGP can not prevent this attack, while our SPP works effectively.

6. Security Considerations

The entire document is about security consideration. More theoretical analysis and experiment results can be found in our technical report [TR-FSBGP].

7. IANA Considerations

This document requires no IANA actions.

8. Conclusions

This draft proposes Fast Secure BGP (FS-BGP), an efficient mechanism for securing feasible AS paths and preventing prefix hijacking by signing critical AS path segments. We believe that FS-BGP can achieve similar level of security as S-BGP. Our experiment results show that, FS-BGP has a much higher efficiency.

9 References

9.1 Normative References

[RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, June 1999.

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

[S-BGP] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure Border Gateway Protocol (S-BGP)", IEEE Journal on Selected Areas in Communications, 18:103-116, 2000.

[I-D.lepinski-bgpsec-protocol] M. Lepinski, "BGPSEC Protocol Specification", draft-lepinski-bgpsec-protocol, work-in-progress, 2011.

[I-D.ng-sobgp-bgp-extensions] J. Ng, "Extensions to BGP to Support Secure Origin BGP (soBGP)", draft-ng-sobgp-bgp-extensions, 2004.

[IRV] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. D. McDaniel, and A. D. Rubin, "Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing", In NDSS, 2003.

[psBGP] P. C. van Oorschot, T. Wan, and E. Kranakis, "On interdomain routing security and pretty secure BGP (psBGP)", ACM Trans. Inf. Syst. Secur., 10(3), 2007.

[SPV] Y.-C. Hu, A. Perrig, and M. A. Sirbu, "SPV: secure path vector routing for securing BGP", In SIGCOMM, pages 179-192, 2004.

[SA] D. M. Nicol, S. W. Smith, and M. Zhao, "Evaluation of efficient security for BGP route announcements using parallel simulation", Simulation Modeling Practice and Theory, 12(3-4):187-216, 2004.

[TR-FSBGP] Yang Xiang, Zhiliang Wang, Xia Yin, Xingang Shi, and Jianping Wu, "FS-BGP: An Efficient Approach to Securing AS Paths", Tsinghua University, Technical Report, THUTR-2011-FSBGP, 2011.

9.2 Informative References

[Whisper] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz, "Listen and Whisper: Security Mechanisms for BGP", In NSDI, pages 127-140, 2004.

[PGBGP] J. Karlin, S. Forrest, and J. Rexford, "Pretty Good BGP: Improving BGP by Cautiously Adopting Routes", In ICNP, pages 290-299, 2006.

Authors' Addresses

Xia Yin
Tsinghua University, Beijing, 100084 P.R. China

Email: yxia@csnet1.cs.tsinghua.edu.cn

Yang Xiang
Tsinghua University, Beijing, 100084 P.R. China

Email: xiangy08@csnet1.cs.tsinghua.edu.cn

Zhiliang Wang
Tsinghua University, Beijing, 100084 P.R. China

Email: wzl@csnet1.cs.tsinghua.edu.cn

Jianping Wu
Tsinghua University, Beijing, 100084 P.R. China

Email: jianping@csnet1.cs.tsinghua.edu.cn

Secure Inter-Domain Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2012

R. Bush
Internet Initiative Japan
B. Wijnen
RIPE/NCC
K. Patel
Cisco Systems
M. Baer
SPARTA
July 4, 2011

Definitions of Managed Objects for BGP Origin Validation
draft-ymbk-bgp-origin-validation-mib-00

Abstract

This document extends the current Management Information Base (MIB) defined for BGP in RFC 4273 to provide support for BGP Origin Validation. In particular, it describes manage objects used for managing BGP Origin validation state within BGP 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 January 5, 2012.

Table of Contents

1. Introduction

This document extends a portion of the BGP4 Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for managing BGP Origin validation within BGP protocol.

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. Overview

3. Definitions

```
BGP-ORIG-VAL-MIB DEFINITIONS ::= BEGIN

IMPORTS

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

    InetAddressType, InetAddress, InetAddressPrefixLength
            FROM INET-ADDRESS-MIB

    TEXTUAL-CONVENTION, DateAndTime, TruthValue
            FROM SNMPv2-TC;

bgpOrigVal  MODULE-IDENTITY
    LAST-UPDATED "201106280000Z"

    ORGANIZATION "IETF Secure Inter-Domain Routing Working Group
                  (SIDR)"
    CONTACT-INFO "N/A"

    DESCRIPTION  "This MIB contains management objects to support
                  the Border Gateway Protocol's (BGP) Origin
                  Validation.

                  Copyright (c) 2011 IETF Trust and the persons
                  identified as authors of the code. All rights
```

```
                    reserved.

                    Redistribution and use in source and binary forms,
                    with or without modification, is permitted pursuant
                    to, and subject to the license terms contained in,
                    the Simplified BSD License set forth in Section 4.c
                    of the IETF Trust's Legal Provisions Relating to
                    IETF Documents
                    (http://trustee.ietf.org/license-info)."

        REVISION     "201106280000Z"
        DESCRIPTION "Initial version of BGP Origin Validation MIB."

        ::= { mib-2 XXX }  -- XXX to be assigned by IANA

    bgpValMIB             OBJECT IDENTIFIER ::= { bgpOrigVal 1 }
    bgpValNotifications   OBJECT IDENTIFIER ::= { bgpValMIB  0 }
    bgpValGen             OBJECT IDENTIFIER ::= { bgpValMIB  1 }
    bgpValROA             OBJECT IDENTIFIER ::= { bgpValMIB  2 }
    bgpValRPKI            OBJECT IDENTIFIER ::= { bgpValMIB  3 }
    bgpValPFX             OBJECT IDENTIFIER ::= { bgpValMIB  4 }
    bgpValGhost           OBJECT IDENTIFIER ::= { bgpValMIB  5 }
    bgpValRepos           OBJECT IDENTIFIER ::= { bgpValMIB  6 }


    bgpValROATable OBJECT-TYPE
        SYNTAX      SEQUENCE OF BgpValROATableEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table lists the ROAs on this system."
        ::= { bgpValROA 1 }

    bgpValROATableEntry OBJECT-TYPE
        SYNTAX      BgpValROATableEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An entry in the bgpValROATable."
        INDEX   { bgpVRTasNum }
        ::= { bgpValROATable 1 }

    BgpValROATableEntry ::= SEQUENCE {
        bgpVRTasNum         Unsigned32,
        bgpVRTPrefixType    InetAddressType,
        bgpVRTPrefix        InetAddress,
        bgpVRTPrefixLength  InetAddressPrefixLength,
        bgpVRTValid         INTEGER
```

```
   }

   bgpVRTasNum OBJECT-TYPE
       SYNTAX      Unsigned32
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This value represents the AS number for this row in the
           table."
       ::= { bgpValROATableEntry 1 }

   bgpVRTPrefixType OBJECT-TYPE
       SYNTAX      InetAddressType
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "This is the address type of the prefix in this row."
       ::= { bgpValROATableEntry 2 }

   bgpVRTPrefix OBJECT-TYPE
       SYNTAX      InetAddress
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "This is the prefix of the ROA indicated by this row."
       ::= { bgpValROATableEntry 3 }

   bgpVRTPrefixLength OBJECT-TYPE
       SYNTAX      InetAddressPrefixLength
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "This is the length of the prefix for the ROA."
       ::= { bgpValROATableEntry 4 }

   bgpVRTValid OBJECT-TYPE
       SYNTAX      INTEGER { unknown(1), valid(2), invalid(3) }
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "This is indicates if the state of the roa associated with
           this row."
       DEFVAL { unknown }
       ::= { bgpValROATableEntry 5 }

   END
```

4. Contributors

5. Acknowledgements

6. 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
bgpOrigVal	{ mib-2 XXX }

7. Security Considerations

This extension to [RFC4273] does not change the underlying security issues inherent in the existing BGP and [RFC4273].

8. References

8.1. Normative References

[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-01 (work in progress), February 2011.

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

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "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.

[RFC2842] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 2842, May 2000.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC4273] Haas, J. and S. Hares, "Definitions of Managed Objects for BGP-4", RFC 4273, January 2006.

8.2. Informative References

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

Authors' Addresses

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: bwijnen@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

Secure Inter-Domain Routing Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2012

R. Bush
Internet Initiative Japan
B. Wijnen
RIPE NCC
K. Patel
Cisco Systems
M. Baer
SPARTA
July 11, 2011

Definitions of Managed Objects for RPKI Router Protocol
draft-ymbk-rpki-rtr-protocol-mib-01

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 managing 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 January 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document defines a portion of the BGP4 Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it extends SNMP MIB and defines objects used for managing the RPKI Router protocol.

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 control and manage RPKI Router protocol. The MIB defined in this draft is broken into two tables: the RPKI Router Peer Table, and the RPKI Router Rceived Record Table. The RPKI Router PeerTable contains information about state and current activity of connections with the RPKI Router peers. The RPKI Router Received Record Table contains IP prefixes and its record information received from all peers running RPKI Router protocol.

4. Definitions

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

IMPORTS
```

```
       MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Integer32,
       Unsigned32, mib-2, Counter64
               FROM SNMPv2-SMI

       InetAddressType, InetAddress, InetAddressPrefixLength
               FROM INET-ADDRESS-MIB

       TEXTUAL-CONVENTION, DateAndTime, TruthValue
               FROM SNMPv2-TC;

   rpkiRouter  MODULE-IDENTITY
       LAST-UPDATED "201107110000Z"

       ORGANIZATION "IETF Secure Inter-Domain Routing Working Group
                     (SIDR)"
       CONTACT-INFO "Work 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: bwijnen@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 contains management objects to support the
                    Resource Public Key Infrastructure (RPKI) protocol
```

```
                    on routers.

                    Copyright (c) 2011 IETF Trust and the persons
                    identified as authors of the code. All rights
                    reserved.

                    Redistribution and use in source and binary forms,
                    with or without modification, is permitted pursuant
                    to, and subject to the license terms contained in,
                    the Simplified BSD License set forth in Section 4.c
                    of the IETF Trust's Legal Provisions Relating to
                    IETF Documents
                    (http://trustee.ietf.org/license-info)."

        REVISION     "201107110000Z"
        DESCRIPTION "Decond version of RPKI Router MIB."

        ::= { mib-2 XXX }  -- XXX to be assigned by IANA

    rpkiRouterMIB            OBJECT IDENTIFIER ::= { rpkiRouter    1 }
    rpkiRouterNotifications  OBJECT IDENTIFIER ::= { rpkiRouterMIB 0 }


    rpkiRPeerTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF RpkiRPeerTableEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table lists the RPKI peers known to this system."
        ::= { rpkiRouterMIB 1 }

    rpkiRPeerTableEntry OBJECT-TYPE
        SYNTAX      RpkiRPeerTableEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An entry in the rpkiPeerTable.  It holds management objects
            associated with a RPKI Peer."
        INDEX { rpkiRPTRemoteAddressType, rpkiRPTRemoteAddress,
                rpkiRPTRemoteAddressPort }
        ::= { rpkiRPeerTable 1 }

    RpkiRPeerTableEntry ::= SEQUENCE {
        rpkiRPTRemoteAddressType  InetAddressType,
        rpkiRPTRemoteAddress      InetAddress,
        rpkiRPTRemoteAddressPort  Integer32,
        rpkiRPTLocalAddressType   InetAddressType,
        rpkiRPTLocalAddress       InetAddress,
```

```
        rpkiRPTLocalAddressPort   Integer32,
        rpkiRPTConnectProtocol    INTEGER,
        rpkiRPTConnectionStatus   INTEGER,
        rpkiRPTMsgsFromPeer       Counter64,
        rpkiRPTMsgsToPeer         Counter64,
        rpkiRPTActiveRecords      Unsigned32,
        rpkiRPTRecordsAnnounced   Unsigned32,
        rpkiRPTRecordssWithdrawn  Unsigned32,
        rpkiRPTHighestSerialNum   Unsigned32,
        rpkiRPTNonce              Integer32,
        rpkiRPTStalePathTimer     Integer32,
        rpkiRPTStalePathTimeRemaining Integer32,
        rpkiRPTRefreshTimer           Integer32,
        rpkiRPTRefreshTimeRemaining   Integer32
    }

    -- XXX needs more objects, see Keyurs email S:first cut, RFC4273

    rpkiRPTRemoteAddressType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network address type of the remote peer."
        ::= { rpkiRPeerTableEntry 1 }

    rpkiRPTRemoteAddress OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network address of the remote peer."
        ::= { rpkiRPeerTableEntry 2 }

    rpkiRPTRemoteAddressPort OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The network address port of the remote peer."
        ::= { rpkiRPeerTableEntry 3 }

    rpkiRPTLocalAddressType OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The network address type of the local connection to the
```

```
            remote peer."
        ::= { rpkiRPeerTableEntry 4 }

    rpkiRPTLocalAddress OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The network address of the local connection to the remote
            peer."
        ::= { rpkiRPeerTableEntry 5 }

    rpkiRPTLocalAddressPort OBJECT-TYPE
        SYNTAX      Integer32 (0..65535)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The network address port of local connection to the remote
            peer."
        ::= { rpkiRPeerTableEntry 6 }


    rpkiRPTConnectProtocol OBJECT-TYPE
        SYNTAX      INTEGER { ssh(1), tcp(2) }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The protocol used to connect to the peer"
        ::= { rpkiRPeerTableEntry 7 }

    rpkiRPTConnectionStatus OBJECT-TYPE
        SYNTAX      INTEGER { up(1), down(2) }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The RPKI connection status for this peer."
        ::= { rpkiRPeerTableEntry 8 }

    rpkiRPTMsgsFromPeer OBJECT-TYPE
        SYNTAX      Counter64
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Number of messages received from peer."
        ::= { rpkiRPeerTableEntry 9 }

    rpkiRPTMsgsToPeer OBJECT-TYPE
        SYNTAX      Counter64
```

```
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "Number of messages sent to the peer."
        ::= { rpkiRPeerTableEntry 10 }

    rpkiRPTActiveRecords OBJECT-TYPE
        SYNTAX      Unsigned32(0..4294967295)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "Number of active records received from the peer."
        ::= { rpkiRPeerTableEntry 11 }

    rpkiRPTRecordsAnnounced OBJECT-TYPE
        SYNTAX      Unsigned32(0..4294967295)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The total number of records announced by the peer."
        ::= { rpkiRPeerTableEntry 12 }

    rpkiRPTRecordssWithdrawn OBJECT-TYPE
        SYNTAX      Unsigned32(0..4294967295)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The total number of records withdrawn by the peer."
        ::= { rpkiRPeerTableEntry 13 }

    rpkiRPTHighestSerialNum OBJECT-TYPE
        SYNTAX      Unsigned32(0..4294967295)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The highest serial number of data received from the peer.
            Note: this value wraps back to zero."
        REFERENCE   "RFC1982"
        ::= { rpkiRPeerTableEntry 14 }

    rpkiRPTNonce OBJECT-TYPE
        SYNTAX      Integer32(0..65535)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The Nonce associated with the peer.  Note: "
        ::= { rpkiRPeerTableEntry 15 }
```

```
   rpkiRPTStalePathTimer OBJECT-TYPE
       SYNTAX      Integer32
       UNITS      "seconds"
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "The number of seconds configured for the stale path timer."
       ::= { rpkiRPeerTableEntry 16 }

   rpkiRPTStalePathTimeRemaining OBJECT-TYPE
       SYNTAX      Integer32
       UNITS      "seconds"
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "The number of seconds remaining of the stale path timer for
           this peer."
       ::= { rpkiRPeerTableEntry 17 }

   rpkiRPTRefreshTimer OBJECT-TYPE
       SYNTAX      Integer32
       UNITS      "seconds"
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "The number of seconds configured for the refresh timer."
       ::= { rpkiRPeerTableEntry 18 }

   rpkiRPTRefreshTimeRemaining OBJECT-TYPE
       SYNTAX      Integer32
       UNITS      "seconds"
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
           "The number of seconds remaining of the refresh timer for
           this peer."
       ::= { rpkiRPeerTableEntry 19 }


   END
```

5. Contributors

6. Acknowledgements

7. 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 }

8. Security Considerations

9. References

9.1. Normative References

[I-D.ietf-sidr-rpki-rtr] Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-13 (work in progress), June 2011.

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

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "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.

[RFC2842] Chandra, R. and J. Scudder, "Capabilities Advertisement with BGP-4", RFC 2842, May 2000.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

9.2. Informative References

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

Authors' Addresses

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: bwijnen@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

INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: January 12, 2012

Mingui Zhang
Huawei
Bin Liu
Tsinghua University
Dacheng Zhang
Huawei
Beichuan Zhang
The University of Arizona

July 11, 2011

Secure Extension of BGP by Decoupling Path Propagation and Adoption
draft-zhang-idr-decoupling-03

Abstract

This draft proposes a novel mitigation scheme to protect the inter-domain data delivery during false routing announcements. A new path attribute is defined to Decouple propagation of a path and adoption of a path for data forwarding in BGP (DBGP). DBGP does not use suspicious paths for data forwarding, but still propagates them in the routing system to facilitate attack detection. It can extensively protect data delivery from routing announcements of false sub-prefixes, false origins, false nodes and false links, and works well with ongoing attack detection and prevention systems.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Acknowledgements

The helpful comments of the following are hereby acknowledged, in alphabetic order: Alvaro Retana, Xiaohu Xu.

Table of Contents

1. Introduction

False routing announcements cause serious security issues to the inter-domain routing system, which can lead to widespread service disruptions. A special case is prefix hijacking, in which a network announces an IP prefix that belongs to another network. Existing works such as Pretty Good BGP (PGBGP)[PGBGP] block suspicious routing updates to protect data delivery. However, such an approach has the side effect of blocking the view of detection systems at the control plane and data plane. As a result, an attack will not be detected, and operators will not be alerted to take actions to stop the attack. This draft proposes an extension of BGP, to solve this paradox by decoupling path propagation and adoption in BGP.

In current BGP, the path a router adopts for data forwarding is the same path being propagated to neighbors. That is why upon receiving a suspicious path, a router has to either accept it (no mitigation but good detection) or reject it (good mitigation but no detection). Our idea is for a router to use trusted paths for data forwarding, but still inform its neighbors about the suspicious path. The suspicious paths will be carried in an optional transitive attribute in BGP updates, while the routers still use trusted paths for data forwarding. This way the data traffic is protected while false routing announcements are being propagated to detection systems.

2. Terminology

DBGP: Decoupling path propagation and adoption in BGP

3. False Routing Announcements

3.1. Problem

False routing announcements can be caused by either inadvertent mis-configurations or malicious attacks. For the ease of exposition, we use "attacker" to refer to the network (or Autonomous System, AS) that makes the false routing announcements regardless of the intention. Based on which part of the routing path is false, such announcements can be classified into five types, each of which has different severity:

- Prefix origin: The attacker originates someone else’s prefix. Depending on where a network is in the Internet topology, some will choose paths leading to the true origin and some will choose paths leading to the false origin.

- Intermediate node: The attacker does not forge the prefix or its origin, but inserts itself into the path as an intermediate

AS. Similar to the case of false origin, some networks will choose the correct paths and some the false paths. But usually fewer networks will choose the false path since it is longer than that of false origin attack.

- Intermediate link: The attacker forges a new link to bypass some of the ASes to get a shorter path. The shortened path is expected to attract more networks' traffic.

- Sub-prefix: The attacker originates a sub-prefix of someone else's prefix. Due to the longest match in routing lookup, a false sub-prefix will win over the original prefix. Thus, all traffic destined to the sub-prefix range will be forwarded to the attacker.

- Super-prefix: Theoretically the attacker can also announce a false super-prefix, but that will not attract any traffic unless part of the prefix range is unused by the real owner, in which case it is equivalent to announcing a false origin of the unused prefix range.

3.2. Countermeasures

The current strategies proposed for the problem of false announcements fall in three categories: prevention, detection and mitigation.

3.2.1. Prevention

Prevention schemes (e.g., [SBGP], [SoBGP], and [SPV]) use cryptographic mechanisms to protect the routing updates and let routers reject any forged announcement. Unfortunately no prevention scheme has seen much deployment on the Internet due to the lack of incentives for those first movers. Since crypto-based schemes add significant computational load to routers and require upgrade on software or hardware, individual ISPs need to see immediate benefits to justify the deployment. On the current Internet, however, the first mover's routing announcements will be accepted by other networks without authentication, and adding authentication does not bring any immediate benefit.

3.2.2 Detection

The representative detection systems proposed in recent years include [Cyclops], [PHAS], [MyASN], [IAR], [iSPY], [NWatch], [OList], and [LWeight]. Such systems are designed to detect false routing by examining routing updates, probing data paths, cross-checking with registry databases, or a combination of these techniques. Once a

false routing case is detected, the owner of the prefix will be notified, and it is expected that the owner will take actions to resolve the problem, which, in today's Internet, usually involves contacting the offending network or its upstream provider to stop the false routing announcements. This process of detection, notification and resolution takes time, ranging from an hour to a day in some past incidents and varying from network to network [NWatch][RIPE]. In the meantime, the damage to data traffic has already been made and malicious attackers may have already achieved their objectives.

3.2.3. Traditional Mitigation

A mitigation schemes attempts to protect the data traffic while an attack is going on. The common approach is to somehow identify abnormal routing updates and block them. As proposed in [PGBGP] and [PGBGP++], a router can examine the content of incoming updates. If an AS path contains unexpected prefix origin or links, it will be suppressed from propagating for a period of time to wait for the network operator's validation. The length of the time is configurable by the network operators. Instead, an alternative path (via trusted prefix origin and links) will be employed for data delivery in the suppression period. After this period, if the path is not proved to be illegal, the router will adopt and announce this new path. PGBGP gives operators certain reaction time to resolve potential false announcements while protecting data delivery in the meantime. When a real attack is detected and resolved, the corresponding false announcement will be withdrawn from the routing system, whereas legitimate announcement will stay in the routing system and eventually be accepted.

But blocking false routing announcements can get in the way of detection systems. For instance, on September 22, 2008, a Russian ISP AS8997 hijacked a large number of prefixes as it leaked an entire table [ASN8997]. However, since the upstream ISP of AS8997 blocked the routing updates, detection systems such as MyASN and IAR did not pick up this incident. The attack mainly affected ISPs and users within Russia but largely went unnoticed by prefix owners.

Each existing solution has its drawbacks, and none is sufficiently effective by itself. In future, there may be several different solutions deployed on the Internet at the same time, complementary to each other, forming a multi-line defense to protect routing and data delivery before, during, and after attacks.

3.3. Paradox of Blocking Suspicious Updates and Attack Detection

It has been realized that a mitigation mechanism and a detection system that complement each other well can be integrated into an effective routing defense solution. For instance, the mitigation mechanism can help the detection system to confine the damage caused by an attack, as the affected data traffic may be vulnerable for hours before the attack can be actually detected by the detection mechanism and be eventually stopped. Also, the mitigation mechanism can also obtain benefit from the detection system because a mitigation mechanism normally cannot identity false routing information accurately with its limited knowledge, resource and time. The output from the detection system can be used to correct the many false positives generated by the mitigation mechanism and also inform

the prefix owner to resolve the attack.

However, there is a dilemma: the mitigation mechanism tries to render the attack ineffective while the detection system needs the attack to be effective in order to detect it.

4. DBGP's Mitigation: Decoupling Path Propagation and Adoption

The suspicious paths are serially blocked hop by hop for validation in traditional mitigation schemes, which gets in the way of detection systems. In order to solve this dilemma, this document proposes a solution which decouples path propagation and path adoption. The basic idea of this solution is to extend BGP's update message with a new optional transitive path attribute so that a router can inform its neighbor routers about the suspicious path and meanwhile the router uses another trusted path for data forwarding. In order to achieve this, a new BGP attribute, DAS_PATH, is defined in Section 5. Compared to the traditional mitigation schemes, the propagation of suspicious paths through DAS_PATHs in DBGP enables the parallel validation, which accelerates the adoption process of suspected legitimate paths (the false positive).

4.1. Quarantine Time

Based on operating experiences, a false routing announcement can be detected and corrected in a certain period (e.g., one day) after it is launched, and so an announcement will be trusted if it is not withdrawn in a pre-defined period. Therefore, in mitigation schemes, when a new path is identified as a suspicious one, it will be quarantined (or blocked) from being used for a period of time, which is called the quarantine time and noted as T_q. If a new path has stayed in a router's Adj-RIBs-In for more than T_q, it will be trusted by this router. If this path is the most preferred from the Adj-RIBs-In, the router will use this path for data delivery and announce this path to its neighbors.

A router MAY determine the quarantine time itself. Assume there are two routers, R1 and R2. R1 is the downstream of R2, and the quarantine time of R1 and R2 is T1 and T2 respectively. The suspicious path is PATH at R1. There are two possibilities.

- T1 is shorter than T2. When T1 expires, PATH becomes trusted by R1 and R1 begins to use it for data delivery. R1 will announce R1-PATH as a legitimate AS path to R2. However, at this time, the path R1-PATH is still being quarantined by R2.

- T1 is longer than T2. When T2 expires, R1-PATH becomes trusted by R2. However, R2 cannot use this path for data delivery as R1

has not announced R1-PATH as an AS path. It will be cached in the Adj-RIBs-In until the downstream router R1 has announced it as an AS path when T1 expires.

5. Protocol Descriptions

Take Figure 5.1 for example. A, B, C, and O are DBGP routers residing in different ASes, , X is the attacker and p is the prefix of interest. Before the attack, the preferred path for traffic is ABCO. Here, we use the notation "R1R2...Rn-p" to denote the AS path which is destined to prefix "p" via routers R1, R2, ..., Rn. When X makes a false announcement of X-p to B, B will regard this new path as suspicious because it would divert traffic to an AS(X) that previously was not on the data path (BCO). B will store the suspicious path in its Adj-RIBs-In (The routing tables of an AS router is comprised of three sub-tables: Adj-RIBs-In, Loc-RIB and Adj-RIBs-Out [BGP4]), but keep using the existing path in its Loc-RIB for data forwarding. At the same time, B re-announces its path (BCO) in an update message to A, and encapsulates the new, suspicious path (BX) as an optional transitive attribute which is defined in Section 5.1. After receiving the update message, router A learns this suspicious path, stores it, and propagates it further to its neighbors using the optional attribute in the same way. Therefore, the suspicious path is propagated to the Internet while not adopted for data forwarding. This approach enable the detection systems to intercept the suspicious path and notify the prefix owner to take actions. Once the attack is stopped, the false announcements will be withdrawn from the routing system, i.e., deleted or replaced in the Adj-RIBs-In of the involved routers. However, a router realizes that the quarantine time has been expired and the suspicious path is still in the Adj-RIBs-In, the router regards the path as a legitimate one. Hence the DBGP router will install the path in its Loc-RIB for data forwarding and re-announce the path using the regular ASPATH attribute in the update message. For example, if BX-p has been validated as legitimate, B will announce it to A as its AS_PATH. The rest of this section will discuss the design details in new BGP attribute definition, identifying suspicious paths, choosing alternative paths for data forwarding, propagating the paths, and releasing quarantined paths.

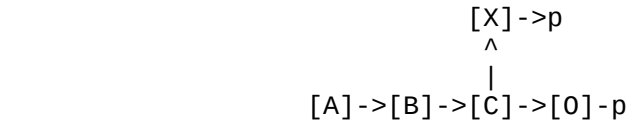


Figure 5.1: Attack Example 1

The rest of this section will discuss the design details in the

definition of the new DAS_PATH Attribute, identifying suspicious paths, choosing alternative paths for data forwarding, propagating the paths, and releasing quarantined paths.

5.1. DAS_PATH Attribute

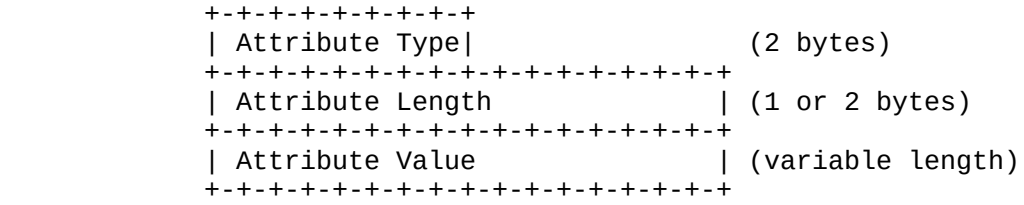


Figure 5.2: The DAS_PATH Attribute

DAS_PATH (Decoupled AS_PATH) is defined as a new optional transitive Path Attribute (Figure 5.2) to be included in BGP's UPDATE messages.

The first bit of the Attribute Type is set (1), therefore the attribute is optional. The second bit of Attribute Type is set (1), therefore the attribute is transitive and SHOULD be passed on to other BGP peers. The third and fourth bits are Partial bit and Extended Length bit respectively, which has already been defined in [BGP4]. The Attribute Type code is assigned by the IANA (Internet Assigned Numbers Authority). The definition of Attribute Length is the same as that in [BGP4].

The Attribute Value is composed of a sequence of DAS path segments. Each DAS path segment is encoded as a triple <path segment type, path segment length, path segment value>.

The path segment type is a 1-octet field with the following two allowable values:

Value	Segment	Type
1	DAS_SET	unordered set of ASs a route in the UPDATE message has traversed
2	DAS_SEQUENCE	ordered set of ASs a route in the UPDATE message has traversed

The path segment length field is 1-octet long field and contains the number of ASs in the path segment value field.

The path segment value field contains one or more AS numbers, each encoded as a 2-octets long field.

When a DAS_PATH is propagated across the network, the operations on

DAS_PATH follows the well-known AS_PATH attribute only that DAS_PATH is non-mandatory. If a router which does not deploy DBGP receives the update messages containing DAS_PATH attribute, i.e., does not understand this attribute, it will just pass it on to the next router. If its downstream router is a DBGP router, it will be able to pick up the information from this attribute and continue DBGP operations. Therefore DBGP can be incrementally deployed over the Internet.

5.2. Identification of Suspicious Paths

For a given prefix p, a path is trusted if it has been staying in the Adj-RIBs-In continuously for the required quarantine time, T_q. All the nodes, links, and origins that appear in trusted paths are trusted components, and the set of them is denoted by trusted(p). This set of trusted components is derived from current contents of all Adj-RIBs-In without using a database to store historical information like PGBGP does. Nodes, links, and origins that do not belong to trusted(p) are said to be suspicious components for this particular prefix p. A new path is suspicious if it contains any suspicious component for its prefix. However, not all suspicious paths need to be explicitly quarantined. DBGP quarantines paths that satisfy the following condition:

- A new path is quarantined if and only if it is suspicious, more preferred than other alternative paths, and contains an AS that is not in the current data forwarding path.

If the new path is not better than alternative paths, it will not be able to divert any traffic. One may suggest that the attacker can first announce a less preferred path so that DBGP routers will take it as a backup path without suspicion, and then make the primary path fail to trick the router to use the false backup path. But in this case, if the attacker has the control of the primary path, it can already get the traffic without doing this. If the attacker does not have control of the primary path, it will not know when the primary path may fail and which backup path the router will choose, thus the attack will not be effective.

If the new path does not introduce any new AS on the data path, it is not quarantined since it does not divert any traffic. In Figure 5.3, when X launches an attack by announcing X-p, this path is not quarantined by B since B already sends its traffic to X. B will accept this path and announces it to A. Assuming ABX-p is more preferred than ACO-p, A will quarantine ABX-p since this new path would divert A’s traffic to a new place, AS X, and X is a suspicious origin to A.

To reduce the potential false positives in quarantining paths, we introduce an optimization rule. If the current best path is replaced by a suspicious path (i.e., the same neighbor that sent the best path previously sends another path to replace it), then the current best path is cached for a short time period. This rule is used to accommodate some unstable prefixes, which may get announced and withdrawn or oscillate between two paths frequently. With this rule, quick re-announcement of a path will not be treated as suspicious. Previous measurement work has shown that (1) most prefixes are stable, and only a small number of prefixes are very unstable; (2) the most popular prefixes are stable; and (3) the most preferred paths are being used by routers for most of the time [BGPpop][Stable][BGPHA]. Therefore, DBGP's criterion for suspicious paths will not generate excessive false positives in reality. The majority of the rest prefixes is only suspicious infrequently.

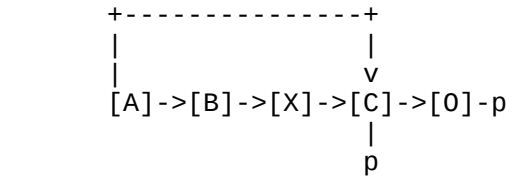


Figure 5.3: Attack Example 2

5.3. Propagating DAS_PATHs with Updates

DBGP uses the new BGP attribute, DAS_PATH, to carry a suspicious path in an update. In certain cases DBGP router may need to select a DAS_PATH from multiple ones to announce. For example, in Figure 5.4, assume C does not deploy DBGP and accepts the false announcement of X-p. When B receives BCX-p, B quarantines this new path and switches to its backup BDO-p. The update from B to A will have BDO as the AS_PATH, and BCX as the DAS_PATH attribute. However, since BDO is suspicious to A as well, A will quarantine BDO and switch to AEFO. Thus the update from A can contain either ABCX or ABDO. A router's neighbors may simultaneously send updates containing the DAS_PATH to it, which also makes the router have multiple DAS_PATHs to choose. There are two possible choices for DBGP implementations.

1. Dominate a router to select the best DAS_PATH from the multiple ones to announce, which is similar to the traditional AS_PATH selection process.

2. Encapsulate all the DAS_PATH attributes in tandem and sort them by their priorities. The priorities are determined according to the rules used in the AS_PATH selection process.

The first rule works well due to the following two facts. First, during attacks, the best DAS_PATH is most likely the bogus path aiming to attract data traffic. Second, during false positives, the best DAS_PATH will most likely become the best AS_PATH when the quarantine time ends.

The second rule allows an AS router to provide more information to its upstream, which is helpful for diagnose and correctness of attacks. Moreover, the announcement of multiple DAS_PATHs MAY help to reduce the convergence time. Take Figure 5.4 for example, A announces two DAS_PATHs, i.e., ABDO-p and ABCX-p, to its upstream node, says U. When U receives the additional DAS_APTH: ABDO-p, it will begin the validation process of this suspicious path. After A determines that BDO-p is legitimate and installs it to its Loc-RIB, the validation process MAY have been finished, therefore U can immediately start to use ABDO-p.

For the second rule, the number of DAS_PATHs in an update depends on the topology and policy of the network. Generally speaking, the number of DAS_PATHs in an update increases with the AS hop count of the AS_PATH. However, given AS paths are usually 4 to 5 hops and rarely goes to more than 10 hops, we do not expect the announcement of multiple DAS_PATHs will make DBGP message too large.

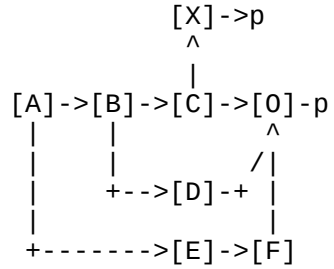


Figure 5.4: Attack Example 3

5.4. Choosing Alternative Paths

When a new path is the most preferred but suspicious, DBGP routers will use an alternative path for data delivery. The question is which alternative path to be chosen. First, if the existing path that is being used for data forwarding is still the best, then the router can stick to that path without any changes. Second, if the existing path in use will have been replaced by the suspicious path, then the router needs to pick an alternative. For example, in Figure 5.4, suppose C does not deploy DBGP and blindly accepts the false announcement X-p. B's existing path BCO-p will be replaced by a suspicious path BCX-p, therefore B needs to temporarily switch to a

backup path BDO-p from its Adj-RIBs-In. Third, if there is no alternative path or all alternative paths are labeled as suspicious, then the router err on data delivery by adopting a suspicious path to forward packets.

5.5 Releasing Quarantined Paths

If the quarantined paths are false announcements, it is likely that within T_q, the attack will be stopped and these paths being withdrawn from the routing system. In this case, there is no explicit release of the quarantined path. Just the upstream router will send an update with empty DAS_PATH attribute. If T_q has passed and the quarantined path is still in the Adj-RIBs-In, then it is more likely that this is a legitimate path. The router will treat the path as a regular path and make it go through the path selection process. If the path turns out to be the most preferred one, it will be used for data forwarding and trigger routing updates to neighbor routers.

6. Clarifications

6.1. Individual Historical Database

When DBGP is implemented in an AS router, the router does not have to purchase additional memory to store the trusted paths as that in [PGBGP]. By default, a DBGP router uses the simple rules defined in Section 5.2 to filter suspected components of an AS path based on the information stored in its Adj-RIBs-In. All a router need to do is to add a new column to its Adj-RIBs-In to record the elapse time after an AS path entered a Adj-RIB-In.

6.2. Difference from "add-path"

DBGP solution is different from the ongoing work of advertising multiple BGP paths in [add-path] where AS routers are also allowed to export multiple AS paths in one update. All advertised AS paths are available to upstream AS routers in [add-path]. Despite that DBGP allows multiple paths to be advertised in one update, except the AS path, all the other paths are actually unavailable. In other words, these paths only remain in Adj-RIBs-In of the AS router. They will not be put into either the Loc-RIB or Adj-RIBs-Out.

6.3 Detection Facilitation

Traditional mitigation mechanisms block the propagation of suspicious paths, which undermines the effectiveness of detection systems. DBGP is proposed to address this shortage. In DBGP, the data traffic is protected while the false routing announcements are spread out to be monitored by detection systems. If the first rule in Section 5.4 is

adopted, the capacity of propagating suspicious paths in DBGP is the same as that in BGP. If the second rule is adopted, this capacity is enhanced by DBGP instead.

6.4. Cooperation with Prevention

As a mitigation scheme, DBGP routers validate AS paths based on the limited information stored in local Adj-RIBs-In. This would cause some legitimate paths to be identified as suspicious and blocked from being used for data delivery (high false positive). If a down stream router would like their paths be adopted quickly rather than be suspected, it can include certificates in the update messages. For example, if AS routers adopt the solution in [pfx-va], the AS number claiming to originate an address prefix will be validated by the prefix holder. The authorized origin will not be suspected by DBGP routers. Further, if the validation can cover the whole AS path, all kinds of attacks that DBGP is trying to cope with SHOULD be prevented in the first place. In all, the deployment of DBGP actually creates the incentive for deploying prevention systems.

6.5. Discontiguous Deployment

DBGP does NOT require contiguous deployment in order to be effective. The key purpose of DBGP is to recognize and propagate the suspicious path segment via the DAS-PATH. When the AS router at the far side obtain the UPDATE message with the DAS-PATH missing some intermediate AS numbers, it doesn’t matter. This AS router can still use this path segment for detection/validation purpose. Let’s use the example in Figure 5.1 to explain. The starter AS router of the DAS-PATH must have deployed DBGP. In this figure, "B" is the starter. "B" can recognize "X" as suspicious node and propagates this via DAS-PATH to "A". We suppose "A" has not deployed DBGP. When A’s upstream AS router receives A’s update message, it will find that the AS-PATH is "ABCO" and the DAS-PATH is "BX". The detection system just uses "BX" as the input.

It is not recommended to complement the DAS-PATH according to the primary path contained in the same UDPATE message. One will easily find this kind of trick is in vain.

7. Security Considerations

The entire document is about security consideration.

8. IANA Considerations

The attribute type code of DAS_PATH should be assigned by the IANA, which identifies the attribute uniquely from all others.

9. References

9.1. Normative References

[PGBGP] J. Karlin, S. Forrest, and J. Rexford, "Pretty Good BGP: Improving BGP by Cautiously Adopting Routes," in Proceedings of IEEE ICNP, 2006.

[SBGP] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (SBGP)," IEEE Journal on Selected Areas in Communications, vol. 18, no. 4, pp. 582-592, 2000.

[SoBGP] J. Ng, "Extensions to BGP to Support Secure Origin BGP," April 2004, ftp://ftp-eng.cisco.com/sobgp/drafts/draft-ng-sobgp-bgp-extensions-02.txt.

[SPV] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure Path Vector Routing for Securing BGP," in Proceedings of ACM SIGCOMM, 2004.

[BGP4] J. W. Stewart, BGP4: Inter-Domain Routing in the Internet. Boston,MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[pfx-va] P. Mohapatra, J. Scudder, D. Ward, R. Bush, R. Austein, "BGP Prefix Origin Validation", draft-ietf-sidr-pfx-validate-01, working in progress.

9.2 Informative References

[Cyclops] Y.-J. Chi, R. Olivera, and L. Zhang, "Cyclops: the as-level connectivity observatory," SIGCOMM Comput. Commun. Rev., vol. 38, no. 5, pp. 5-16, 2008.

[PHAS] M. Lad, D. Massey, D. Pei, B. Zhang, and L. Zhang, "PHAS: A Prefix Hijack Alert System," in 15th USENIX Security Symposium, 2006, pp.153-166.

[MyASN] "RIPE myASN System," http://www.ris.ripe.net/myasn.html.

[IAR] [Online]. Available: http://iar.cs.unm.edu/

[iSPY] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP Prefix Hijacking on My Own," in Proceedings of ACM SIGCOMM, 2008.

[NWatch] G. Siganos and M. Faloutsos, "Neighborhood Watch for

Internet Routing: Can We Improve the Robustness of Internet Routing Today?" in Proceedings of IEEE INFOCOM, 2007.

[Olist] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang,"Dection of Invalid Routing Announcement in the Internet," in Proceedings of the IEEE DSN, June 2002.

[LWeight] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A Light-weight Distributed Scheme for Detecting IP Prefix Hijacks in Real-time," in Proceedings of ACM SIGCOMM, 2007.

[RIPE] [Online]. Available: http://www.ripe.net/news/study-youtubehijacking.html

[ASN8997] "Prefix hijack by ASN 8997." [Online]. Available: http://www.merit.edu/mail.archives/nanog/2008-09/msg00704.html

[BGPpop] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP Routing Stability of Popular Destinations," in Proceedings of ACM IMC 2002, pp. 197-202.

[Stable] K. Butler, P. McDaniel, and W. Aiello, "Optimizing BGP Security by Exploiting Path Stability," in Proceedings of ACM CCS, Alexandria, VA, United States, 2006, pp. 298-310.

[BGPHA] R. V. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang, "Measurement of Highly Active Prefixes in BGP," in Proceedings of IEEE Globecom,2005.

Appendix A: Empirical Evaluation

The following aspects of DBGP are tested on the SSFNet-2.0 simulation platform which has implemented BGP4.

- The ability to counter different types of attacks
- The ability to rectify the false positives
- The memory and message overhead

The evaluation proves that DBGP can be used to mitigate all types of attacks. Compared with previous mitigation approaches [PGBGP], it reduces the delay of legitimate announcements significantly, only incurs a small amount of messages and memory overhead.

Author's Addresses

Mingui Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di
Information Industry Base, Hai-Dian District,
Beijing, 100085 P.R. China

Email: zhangmingui@huawei.com

Bin Liu
Tsinghua University
East Main Building RM9-416
Tsinghua University, Hai-Dian District,
Beijing, 100084 P.R. China

Email: lmyujie@gmail.com

Dacheng Zhang
Huawei Technologies Co.,Ltd
HuaWei Building, No.3 Xinxi Rd., Shang-Di
Information Industry Base, Hai-Dian District,
Beijing, 100085 P.R. China

Email: zhangdacheng@huawei.com

Beichuan Zhang
University of Arizona
Computer Science Department,
The University of Arizona
Tucson, AZ 85721 U.S.A.

Email: bzhang@arizona.edu