        BGPSEC Design Choices and Summary of Supporting Discussions
              draft-sriram-bgpsec-design-choices-10

Abstract

   This document has been written to capture the design rationale for
   the individual draft-00 version of BGPSEC protocol specification (I-
   D.lepinski-bgpsec-protocol-00). It lists the decisions that were
   made in favor of or against each design choice, and presents brief
   summaries of the arguments that aided the decision process. A
   similar document can be published in the future as the BGPSEC design
   discussions make further progress and additional design
   considerations are discussed and 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 5, 2017.

Copyright Notice

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
   [RFC7132] [RFC7353] [I-D.ietf-sidr-bgpsec-overview]
   [I-D.ietf-sidr-bgpsec-protocol]. The key high-level design goals of
   BGPSEC protocol are as follow [RFC7353]:

   o Rigorous path validation for all announced prefixes; not merely
      showing that a path is not impossible.
   o Incremental deployment capability; no flag-day requirement for
      global deployment.
   o Protection of AS paths only in inter-domain routing (eBGP); not
      applicable to iBGP (or to IGPs).

o Aim for no increase in provider's data exposure (e.g., require no
   disclosure of peering relations, etc).

This document is a companion to the earliest version of the BGPSEC
protocol specification submitted as individual draft-00
[I-D.lepinski-bgpsec-protocol], and is intended to provide design
justifications for this initial BGPSEC specification. This document
lists the decisions that were 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. A
similar document can be published in the future as the BGPSEC design
discussions make further progress and additional design
considerations are discussed and 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 NLRI length, NLRI prefix, its
own ASN, the next ASN, the signature algorithm suite ID, 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 data covered by the signature.
Without that the AS path cannot be secured; for example, it can be
shortened (by a MITM) without being detected.

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 NLRI
prefix. If more than one NLRI prefix were included, and an upstream
AS elected to propagate the advertisement for a subset of the
prefixes, then the signature(s) 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 over 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 of
concern that the impact of sending the (potentially large) signatures
to routers that 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 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 NLRI length, NLRI prefix, AS path, next
ASN, signature algorithm suite ID, and Expire Time. There are
multiple choices for what is actually signed by an upstream AS: (1)

Sign over the combination of NLRI length, NLRI prefix, AS path, next
ASN, signature algorithm suite ID, and Expire Time; 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 plus next ASN; thus,
ASi signs over NLRI length, NLRI prefix, signature algorithm suite
ID, 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.lepinski-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. It is well understood 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 NLRI 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. Let AlgID represent the ID of the signature algorithm
suite that is in use. 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:

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

   The draft-00 version of the protocol specification 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 (i.e., beyond draft-00 specification). If such
   optimization is used, a replication count would be included (in the
   signed update) 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
   [RFC6487].

2.7.2. **Discussion**

   It was 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. It is safe to 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.

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 (i.e., beaconing) 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][RIPE580] 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, it may
be noted that the re-advertisements (i.e., beacons) 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 one assumes 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, choice of 2048-bit RSA algorithm for BGPSEC update
   signatures was considered because it is being used ubiquitously in
   the RPKI system. However, use of ECDSA-256 algorithm was decided
   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
[RFC6891].

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], an originating BGP router normally packs multiple prefix (NLRI) announcements into one update if the prefixes all share the same BGP attributes. When an upstream BGP router forwards eBGP updates to its peers, it can also pack multiple prefixes (based on shared AS path and attributes) into one update. The update propagated by the upstream BGP router may include only a subset of the prefixes that were packed in a received update.

5.1.1. Decision

The initial draft-00 BGPSEC specification [I-D.lepinski-bgpsec-protocol] does not accommodate update packing. Each update contains exactly one prefix. This avoids the complexity that would be otherwise inevitable if the origin had packed and signed multiple prefixes in an update and an upstream AS decided to propagate an update containing only a subset of the prefixes in that update. 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 a separate signature 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.

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 if some optimizations such as update packing are adopted 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). It was
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
(see [CPUworkload]). 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 (even though the router itself has not
yet verified the signatures 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. As BGPsec adoption grows, the BGPsec
   islands will expand outward (subsuming non-BGPsec portions of the
   Internet) and/or pairs of islands may join together to form larger
   BGPsec islands.

   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**

   Partial path signing 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 [RFC7353]. Partial path signing clearly re-
   introduces the following attack vulnerability: If a BGPSEC speaker
   can sign an unsigned update, and if signed (i.e., partially or fully
   signed) updates would 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 as discussed below:

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 (Section 6.5.1). A stub AS that does asymmetric
BGPSEC has the advantage that it needs to minimally upgrade to BGPSEC
so it can sign updates to its upstream while it receives only
unsigned updates. Thus it can avoid the cost of increased processing
and memory needed to perform update validations and to store signed
updates in the RIBs, respectively.

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 policies 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 to the data that needs to be
   signed and signs the update for each peer AS in the group
   individually.

If updates were to be signed per peer group, that 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

The need to provide protection in BGPSEC for the community attribute
was discussed.

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 to
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 (draft-00 specification) 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 and this case is trivially supported in BGPSEC.

However, the draft-00 version of BGPSEC specification 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 [ASset]. 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. A recently
published BCP [RFC6472] recommends against the use of AS_SETs in
updates, so it is anticipated that the 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-byte ASN in a
path into a reserved 2-byte ASN (AS 23456) before forwarding the
update. This mechanism is incompatible with use of BGPSEC, since the
ASN translation is equivalent to a route modification attack and will
cause signatures corresponding to the translated 4-byte ASNs to fail
validation.

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 are included in the initial
   draft-00 BGPSEC specification [I-D.lepinski-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 the focus is 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 would 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, four validation states were considered: (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 and that was ruled out earlier. 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.

. **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: Should validation state
   propagation 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:

   o 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.
   o 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. There are potentially 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

   Possibly an off-router box (one or more per AS) can be used that
   performs 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 mlepinski@ncf.edu
        New College of Florida

        Doug Montgomery dougm@nist.gov
        USA NIST

        Kotikalapudi Sriram ksriram@nist.gov
        USA NIST

        Samuel Weiler weiler@watson.org
        Cobham

11. Acknowledgements

   The authors would like to thank John Scudder, Ed Kern, Pradosh
   Mohapatra, Keyur Patel, David Ward, Rudiger Volk, Heather Schiller,
   Jason Schiller, Chris Morrow, Sandy Murphy, Russ Mundy, Mark
   Reynolds, Sean Turner, Sharon Goldberg, Chris Hall, Shane Amante,
   Luke Berndt, and Doug Maughan 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

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

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

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

   [RFC4893] Vohra, Q. and E. Chen, "BGP Support for Four-octet AS
              Number Space", RFC 4893, DOI 10.17487/RFC4893, May 2007,
              <http://www.rfc-editor.org/info/rfc4893>.

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

   [RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms
              for DNS (EDNS(0))", STD 75, RFC 6891,
              DOI 10.17487/RFC6891, April 2013,
              <http://www.rfc-editor.org/info/rfc6891>.

14.2. Informative References

   [ASset]   Sriram, K. and D. Montgomery, "Measurement Data on AS_SET
              and AGGREGATOR: Implications for {Prefix, Origin}
              Validation Algorithms", IETF SIDR WG presentation, IETF
              78, July 2010, <http://www.nist.gov/itl/antd/upload/
              AS_SET_Aggregator_Stats.pdf>.

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

   [CPUworkload]
              Sriram, K. and R. Bush, "Estimating CPU Cost of BGPSEC on
              a Router", Presented at RIPE-63; also at IETF-83 SIDR WG
              Meeting, March 2012,
              <http://www.ietf.org/proceedings/83/slides/
              slides-83-sidr-7.pdf>.

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

   [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]
              Bush, R., Patel, K., and D. Ward, "Extended Message
              support for BGP", draft-ietf-idr-bgp-extended-messages-13
              (work in progress), June 2016.

   [I-D.ietf-sidr-bgpsec-overview]
              Lepinski, M. and S. Turner, "An Overview of BGPsec",
              draft-ietf-sidr-bgpsec-overview-08 (work in progress),
              June 2016.

   [I-D.ietf-sidr-bgpsec-protocol]
              Lepinski, M. and K. Sriram, "BGPsec Protocol
              Specification", draft-ietf-sidr-bgpsec-protocol-17 (work
              in progress), June 2016.

   [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, DOI 10.17487/RFC2439, November
              1998, <http://www.rfc-editor.org/info/rfc2439>.

   [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,
              DOI 10.17487/RFC4055, June 2005,
              <http://www.rfc-editor.org/info/rfc4055>.

   [RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y.
              Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724,
              DOI 10.17487/RFC4724, January 2007,
              <http://www.rfc-editor.org/info/rfc4724>.

   [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, DOI 10.17487/RFC5280, May 2008,
              <http://www.rfc-editor.org/info/rfc5280>.

   [RFC5396] Huston, G. and G. Michaelson, "Textual Representation of
              Autonomous System (AS) Numbers", RFC 5396,
              DOI 10.17487/RFC5396, December 2008,
              <http://www.rfc-editor.org/info/rfc5396>.

   [RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic
              Curve Cryptography Algorithms", RFC 6090,
              DOI 10.17487/RFC6090, February 2011,
              <http://www.rfc-editor.org/info/rfc6090>.

   [RFC6472] Kumari, W. and K. Sriram, "Recommendation for Not Using
              AS_SET and AS_CONFED_SET in BGP", BCP 172, RFC 6472,
              DOI 10.17487/RFC6472, December 2011,
              <http://www.rfc-editor.org/info/rfc6472>.

   [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support
              Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480,
              February 2012, <http://www.rfc-editor.org/info/rfc6480>.

   [RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route
              Origin Authorizations (ROAs)", RFC 6482,
              DOI 10.17487/RFC6482, February 2012,
              <http://www.rfc-editor.org/info/rfc6482>.

   [RFC6483] Huston, G. and G. Michaelson, "Validation of Route
              Origination Using the Resource Certificate Public Key
              Infrastructure (PKI) and Route Origin Authorizations
              (ROAs)", RFC 6483, DOI 10.17487/RFC6483, February 2012,
              <http://www.rfc-editor.org/info/rfc6483>.

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

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

   [RFC7132] Kent, S. and A. Chi, "Threat Model for BGP Path Security",
              RFC 7132, DOI 10.17487/RFC7132, February 2014,
              <http://www.rfc-editor.org/info/rfc7132>.

   [RFC7353] Bellovin, S., Bush, R., and D. Ward, "Security
              Requirements for BGP Path Validation", RFC 7353,
              DOI 10.17487/RFC7353, August 2014,
              <http://www.rfc-editor.org/info/rfc7353>.

   [RIB_size]
              Sriram, K. and et al., "RIB Size Estimation for BGPSEC",
              June 2011, <http://www.nist.gov/itl/antd/upload/
              BGPSEC_RIB_Estimation.pdf>.

   [RIPE580] Bush, R. and et al., "RIPE-580: RIPE Routing Working Group
              Recommendations on Route-flap Damping", January 2013,
              <http://www.ripe.net/ripe/docs/ripe-580>.

Author's Address

   Kotikalapudi Sriram (editor)
   USA NIST
   100 Bureau Drive
   Gaithersburg, MD 20899
   USA

   Email: ksriram@nist.gov