         BGPsec Design Choices and Summary of Supporting Discussions

Abstract

   This document captures the design rationale of the initial draft
   version of what became RFC 8205 (the BGPsec protocol specification).
   The designers needed to balance many competing factors, and this
   document lists the decisions that were made in favor of or against
   each design choice. This document also presents brief summaries of
   the arguments that aided the decision process. Where appropriate,
   this document also provides brief notes on design decisions that
   changed as the specification was reviewed and updated by the IETF
   SIDR Working Group and that resulted in RFC 8205. These notes
   highlight the differences and provide pointers to details and
   rationale regarding those design changes.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   The goal of the BGPsec effort is to enhance the security of BGP by
   enabling full Autonomous System (AS) path validation based on
   cryptographic principles. Standards work on route origin validation
   based on a Resource PKI (RPKI) is already completed or nearing
   completion in the IETF SIDR WG [RFC6480] [RFC6482] [RFC6483]
   [RFC6487] [RFC6811]. 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] [RFC8205]. The BGPsec
   protocol specification, [RFC8205], was published recently. The key
   high-level design goals of the BGPsec protocol are as follows
   [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 (External BGP
      (eBGP)) -- not applicable to Internal BGP (iBGP) (or to IGPs).

   o Aiming for no increase in a provider's data exposure (e.g., not
      requiring any disclosure of peering relations).

   This document provides design justifications for the initial draft
   version of the BGPsec protocol specification [BGPsec-Initial]. The
   designers needed to balance many competing factors, and this document
   lists the decisions that were made in favor of or against each design
   choice. This document also presents brief summaries of the
   discussions that weighed in on the pros and cons and aided the
   decision process. Where appropriate, this document provides brief
   notes (starting with "Note:") on design decisions that changed from
   the approach taken in the initial draft version of the BGPsec
   protocol specification as the specification was reviewed and updated
   by the IETF SIDR WG. (These design decisions resulted in RFC 8205
   [RFC8205].) The notes provide pointers to the details and/or
   discussions about the design changes.

The design choices and discussions are presented in the following
sections (under the following eight broad categories, with many
subtopics within each category):

o Section 2 ("Creating Signatures and the Structure of BGPsec Update
   Messages")

o Section 3 ("Withdrawal Protection")

o Section 4 ("Signature Algorithms and Router Keys")

o Section 5 ("Optimizations and Resource Sizing")

o Section 6 ("Incremental Deployment and Negotiation of BGPsec")

o Section 7 ("Interaction of BGPsec with Common BGP Features")

o Section 8 ("BGPsec Validation")

o Section 9 ("Operational Considerations")

2. Creating Signatures and the Structure of BGPsec Update Messages

2.1. Origin Validation Using ROAs

2.1.1. Decision

Route origin validation using Route Origin Authorizations (ROAs)
[RFC6482] [RFC6811] is necessary and complements AS path attestation
based on signed updates. Thus, the BGPsec design makes use of the
origin validation capability facilitated by the ROAs in the RPKI.

Note: In the finalized BGPsec protocol specification [RFC8205],
BGPsec is synonymous with cryptographic AS path attestation. Origin
validation and BGPsec (path signatures) are the two key pieces of the
SIDR WG solution for BGP security.

2.1.2. Discussion

Route origin validation using RPKI constructs, as developed in the
IETF SIDR WG, is a necessary component of BGP security. 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 Network Layer Reachability
Information (NLRI) length, NLRI prefix, its own AS number (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.

Note: The finalized BGPsec protocol specification [RFC8205] differs
from the above. There is no mention in RFC 8205 of a signature
Expire Time field in the BGPsec update. Further, there are some
additional details concerning attributes signed by the origin AS that
can be found in Figure 8 in Section 4.2 of RFC 8205 [RFC8205]. In
particular, the signed data also includes the Address Family
Identifier (AFI) as described in RFC 8205. By adding the AFI in the
data covered by a signature, a specific security concern was
alleviated; see [Mandelberg1] (post to the SIDR WG Mailing List) and
the discussion thread that followed on the topic. The AFI is
obtained from the MP_REACH_NLRI attribute in the BGPsec update. As
stated in Section 4.1 of RFC 8205, a BGPsec update message "MUST use
the MP_REACH_NLRI attribute [RFC4760] to encode the prefix."

2.2.2. Discussion

The next-hop ASN is included in the data covered by the signature.
Without this inclusion, the AS path cannot be secured; for example,
the path can be shortened (by a MITM (man in the middle)) 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 the 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.)

Note: Expire Time and beaconing were eventually replaced by router
key rollover. The BGPsec protocol [RFC8205] is expected to make use
of router key rollover to mitigate replay attacks and withdrawal
suppression [BGPsec-Rollover] [Replay-Protection].

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
Sections 5.1 and 5.2). If a mechanism were employed to preserve

prefixes that were dropped, this would reveal information to
subsequent ASes that is not revealed in normal BGP operation. Thus,
a trade-off was made to preserve the level of route information
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 about the impact of sending the (potentially large)
signatures to routers that don't understand them. Also, if a router
that does not understand BGPsec somehow gets an update message with
path signatures (i.e., the update includes the BGPsec_PATH attribute
(see Section 3 of RFC 8205)), then it would be undesirable for that
router to forward the update to all of its neighbors, especially
those who do not understand BGPsec and may choke if they receive many
updates with large optional BGP attributes. It is envisioned that
BGPsec and traditional BGP will coexist while BGPsec is deployed
incrementally.

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 (the
origin AS being the nearest to a prefix). In principle, an AS that
is upstream from an originating AS would digitally 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 regarding what is signed by an upstream AS, as
follows:

o Method 1: The signature protects the combination of the NLRI
   length, NLRI prefix, AS path, next ASN, signature algorithm suite
   ID, and Expire Time,

o Method 2: The signature protects just the combination of the
   previous signature (i.e., the signature of the neighbor AS who
   forwarded the update) and the next ASN, or

o Method 3: The signature protects everything that was received from
   the preceding AS plus the next (i.e., target) ASN; thus, AS_i signs
   over the NLRI length, NLRI prefix, signature algorithm suite ID,
   Expire Time, {AS_i, $AS_{(i-1)}$, $AS_{(i-2)}$, ..., AS_2, AS_1}, $AS_{(i+1)}$
   (i.e., the next ASN), and {$Sig_{(i-1)}$, $Sig_{(i-2)}$, ..., Sig_2, Sig_1}.

Note: Please see the notes in Sections 2.2.1 and 2.2.2 regarding the elimination of the Expire Time field in the finalized BGPsec protocol specification [RFC8205].

2.3.1. Decision

It was decided that Method 2 will be used. Please see [BGPsec-Initial] for additional protocol details and syntax.

Note: The finalized BGPsec protocol specification [RFC8205] essentially uses Method 3 (except for Expire Time). Additional details concerning attributes signed by an upstream AS can be found in Figure 8 in Section 4.2 of RFC 8205 [RFC8205]. The decision to go with Method 3 (with suitable additions to the data signed) was motivated by a security concern that was associated with Method 2; see [Mandelberg2] (post to the SIDR WG Mailing List) and the discussion thread that followed on the topic. Also, there is a strong rationale for the sequence of octets to be hashed (as shown in Figure 8 in Section 4.2 of RFC 8205); this sequencing of data is motivated by implementation efficiency considerations. See [Borchert] (post to the SIDR WG Mailing List) for an explanation.

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, the remaining bytes will overflow into a second hash block, resulting in a 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 identify any vulnerabilities associated with this approach.

2.4. Attributes That Are Not Signed

2.4.1. Decision

Any attributes other than those identified in Sections 2.2 and 2.3 are not signed. Examples of such attributes include the community attribute, the NO-EXPORT attribute, and Local_Pref.

2.4.2. Discussion

Any of the above-mentioned attributes that are not signed are viewed
as local (e.g., do not need to propagate beyond the next hop) or lack
clear security needs. NO-EXPORT is sent over a secured next hop and
does not need signing. The 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 following example describes the expected router actions on
receipt of a signed update. 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 (Method 2 in Section 2.3) selected for signing requires a
receiving router in ASi to perform the following actions:

o Validate the route 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).

Note: The above description of BGPsec update validation and
forwarding differs in its details from the published BGPsec protocol
specification [RFC8205]. Please see Sections 4 and 5 of [RFC8205].

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 (ASN) of the router that is
signing the update.

Note: The finalized BGPsec protocol specification [RFC8205] uses a
pCount field associated with each AS in the path to indicate the
number of prepends for that AS (see Figure 5 in Section 3.1 of
[RFC8205]).

2.6.2. Discussion

The initial version [BGPsec-Initial] of the BGPsec specification
calls for a signature to be associated with each prepended AS. The
optimization of having just one signature for multiple prepended ASes
was pursued later. The pCount field is now used to represent AS
prepends; see Section 3.1 in RFC 8205.

2.7. RPKI Data That Needs to Be Included in Updates

2.7.1. Decision

Concerning the inclusion of RPKI data in an update, it was decided
that only the Subject Key Identifier (SKI) of the router certificate
must be included in a signed update. This information identifies the
router certificate, based on the SKI generation criteria defined in
[RFC6487].

2.7.2. Discussion

   Whether or not each router public key certificate should be included
   in a signed update was discussed. 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 the majority of network environments, intermittent connectivity
   would not be a problem. So, it is best to avoid this complexity,
   because the 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 the 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 associated with the origin AS's
   signature (see Section 3.2).

   Repeated withdrawals and announcements for a prefix can run up the
   BGP Route Flap Damping (RFD) penalty [RFC2439] 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

   Note: As mentioned earlier (Section 2.2.2), the Expire Time approach
   to mitigation of replay attacks and withdrawal suppression was
   subsequently changed to an approach based on router key rollover
   [BGPsec-Rollover] [Replay-Protection].

   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.

   It is recommended that the duration of the signature Expire Time be
   on the order of days (preferably), but it may be on the order of
   hours (about 4 to 8 hours) in some cases on the basis of perceived
   need for extra protection from replay attacks (i.e., where extra
   replay protection is perceived 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. Additional guidance can be provided
   via an operational considerations document later. Further, the
   actual re-origination time should 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 detect that 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 non-BGPsec peers. (It
   should sign and re-announce the route to BGPsec speakers only.) This
   procedure will reduce BGP chattiness for the non-BGPsec border
   routers.

3.2.2. Discussion

Mitigation of BGPsec update replay attacks can be thought of as
protection against malicious re-advertisements 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 an 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.

As noted in Section 3.2.1, the recommended Expire Time value is on
the order of days, but 4 to 8 hours may be used in some cases on the
basis of perceived need for extra protection from replay attacks.
Thus, different ASes may choose different values based on the
perceived need to protect against malicious route replays. (A
shorter Expire Time reduces the window during which an AS can
maliciously replay the route. However, shorter Expire Time values
cause routes to be refreshed more often, thus causing more BGP
chatter.) Even a 4-hour duration seems long enough 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 35 updates per second; this can be considered reasonable.
However, further analysis is needed to confirm these recommendations.

As stated in Section 3.2.1, the originating AS will re-originate a
route sufficiently in advance of its Expire Time. What is considered
"sufficiently in advance"? To answer this question, modeling should
be performed to determine the 95th-percentile convergence time of
update propagation in a 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 retransmitting the entire update with signatures) was
considered. However, the decision was made to not do this.
Re-origination to communicate a new signature Expire Time will be
done by propagating a normal update message; no special type of
message will be required.

3.3.2. Discussion

It was suggested that if the re-beaconing of the signature
Expire Time is carried in a separate special message, then any
processing load related to the update may be reduced. But it was
recognized that such a re-beaconing message by necessity 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 such simple situations as a 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
traditional BGP, 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; hence, the
update will need to propagate throughout the Internet (limited only
by the process of best-path selection). 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 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 did not even require any
RFD-related processing), since they are identical except for the
change in the Expire Time value. This can be accomplished by not
assigning an RFD penalty to Expire-Time-triggered updates. If the
community agrees, this could be accommodated, but a change to the
BGP-RFD protocol will be required.

3.4.2. Discussion

   To summarize, this decision is supported by the following
   observations:

   1. Expire-Time-triggered updates are generally not preceded by
       withdrawals; 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 a negligible impact on
       RFD penalty accumulation because the re-advertisement interval is
       much longer relative to the half-time of RFD penalty decay.

   Elaborating further on the third observation 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
   several hours (see Section 3.2). During that time period, any
   incremental contribution to the RFD penalty due to an Expire-Time-
   triggered update would decay sufficiently to have negligible (if any)
   impact on damping the address prefix in question.

   Additional details regarding this analysis and justification are as
   follows:

   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 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 the 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 an
   attribute change. On the other hand, the Cisco implementation
   [CiscoIOS] 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's implementation of RFD would not assign any
   penalty for a re-beaconed update (i.e., a route was already
   advertised previously and was 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 an
   attribute change according to the JunOS RFD implementation), it can
   be illustrated that the incremental effect it would have on damping
   the prefix in question would be negligible: 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 several hours. An
incremental penalty of 500 would decay to 31.25 in 1 hour, 0.12 in
2 hours, and 3x10^(-5) in 3 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, the Elliptic Curve Digital Signature Algorithm (ECDSA)
with curve P-256 and SHA-256 will be used for generating BGPsec path
signatures. One other signature algorithm, e.g., RSA-2048, will also
be used during prototyping and testing. The use of a second
signature algorithm is needed to verify the ability of the BGPsec
implementations to change from a current algorithm to the next
algorithm.

Note: The BGPsec cryptographic algorithms document [RFC8208]
specifies only the ECDSA with curve P-256 and SHA-256.

4.1.2. Discussion

Initially, the RSA-2048 algorithm for BGPsec update signatures was
considered as a choice because it is being used ubiquitously in the
RPKI system. However, the use of ECDSA P-256 was decided upon
because it yields a smaller signature size; hence, the update size
and (in turn) the RIB size needed in BGPsec routers would be much
smaller [RIB_size].

Using two different signature algorithms (e.g., ECDSA P-256 and
RSA-2048) to test the transition from one algorithm to the other will
increase confidence in prototype implementations.

Optimizations and specialized algorithms (e.g., for speedups) built
on Elliptic Curve Cryptography (ECC) algorithms may have active IPR
(intellectual property rights), but at the time of publication of
this document no IPR had been disclosed to the IETF for the basic
(unoptimized) algorithms. (To understand this better, [RFC6090] can
be useful as a starting point.)

Note: Recently, even open-source implementations have incorporated
certain cryptographic optimizations and demonstrated significant
performance speedup [Gueron]. Researchers continue to devote
significant effort toward demonstrating substantial speedup for the
ECDSA as part of BGPsec implementations [Mehmet1] [Mehmet2].

4.2. Agility of Signature Algorithms

4.2.1. Decision

During the transition period from one algorithm (i.e., the current
algorithm) to the next (new) algorithm, the updates will carry two
sets of signatures (i.e., two Signature_Blocks), one corresponding to
each algorithm. Each Signature_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_Blocks and then add its corresponding
signature to each Signature_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_Block of the new algorithm
and then will forward the update after adding its own signature to
the Signature_Block of the current algorithm.

It was decided that there will be at most two Signature_Blocks per
update.

Note: BGPsec path signatures are carried in the Signature_Block,
which is an attribute contained in the BGPsec_PATH attribute (see
Section 3.2 in [RFC8205]). The algorithm agility scheme described in
the published BGPsec protocol specification is consistent with the
above; see Section 6.1 of [RFC8205].

4.2.2. Discussion

A length field in the Signature_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_Block when parsing the message.

The overlap period between the two algorithms is expected to last
2 to 4 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 signatures [RIB_size].

The lifetime of a signature algorithm is anticipated to be much
longer than the duration of a transition period from the current
algorithm to a 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_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 Section 4.3.2 for a brief
description of what the SAS is and what its pros and cons are.

4.3.2. Discussion

In the SAS method, there would be only one (aggregated) signature per
signature block, irrespective of the number of AS hops. For example,
ASn (the 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. The SAS could potentially result in savings in bandwidth
and in Protocol Data Unit (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 its
equivalent. For a SAS with RSA and for the cryptographic strength
needed for BGPsec signatures, a 2048-bit signature size (RSA-2048)
would be required. However, without a SAS, the ECDSA with a 512-bit
signature (256-bit key) would suffice for equivalent cryptographic
strength. The larger signature size of RSA used with a SAS
undermines the advantages of the SAS, because the average hop count,
i.e., the number of ASes, for a route is about 3.8. In the end, it
may turn out that the 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 the SAS is
not a time-tested cryptographic technique, and thus its adoption is
potentially risky.

4.4. Protocol Extensibility

There is 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, a new version
of BGPsec will be required. 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 block may not correspond to one signature per AS in the
AS path (e.g., aggregate signatures).

4.4.1. Decision

This protocol-version transition mechanism is analogous to the
algorithm transition discussed in Section 4.2. During the transition
period from one protocol version (i.e., the current version) to the
next (new) version, updates will carry two sets of signatures (i.e.,
two Signature_Blocks), one corresponding to each version. A
protocol-version identifier is associated with each Signature_Block.
Hence, each Signature_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_Blocks and then add its corresponding signature to each
Signature_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_Block of the new version and then will
forward the update with an attachment of its own signature to the
Signature_Block of the current version.

Note: The details of protocol extensibility (i.e., transition to a
new version of BGPsec) in the published BGPsec protocol specification
(see Section 6.3 in [RFC8205]) differ somewhat from the above. In
particular, the protocol-version identifier is not part of the BGPsec
update. Instead, it is negotiated during the BGPsec capability
exchange portion of BGPsec session negotiation.

4.4.2. Discussion

In the case that a 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_TWO") that is designed to accommodate the
desired changes to BGPsec. At this point, a transition would begin
that 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 (current) attribute (see
Section 3 of RFC 8205) and the new BGPsec_PATH_TWO attribute. Once
the transition is complete, the use of BGPsec_PATH could then be

   deprecated, at which point BGPsec speakers will include only the new
   BGPsec_PATH_TWO attribute. Such a process could facilitate a
   transition to new BGPsec semantics in a backwards-compatible fashion.

4.5. Key per Router (Rogue Router Problem)

4.5.1. Decision

   Within each AS, each individual BGPsec router can have a unique pair
   of private and public keys [RFC8207].

4.5.2. Discussion

   Given a unique key pair per router, 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 certification authority
   (CA) certificate 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 ASN followed by the
   router ID. (Note: The details are specified in Section 3.1 in
   [RFC8209].)

4.6.2. Discussion

   Every X.509 certificate requires a subject name [RFC6487]. 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

   With traditional BGP [RFC4271], an originating BGP router normally
   packs multiple prefix 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 the 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

   Each update contains exactly one prefix. This avoids a level of
   complexity that would otherwise be 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 recommendations regarding packing
   and repacking may be revisited when optimizations are considered in
   the future.

5.1.2. Discussion

   Currently, with traditional BGP, there are, on average, approximately
   four prefixes announced per update [RIB_size]. So, the number of BGP
   updates (carrying announcements) is about four 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 2.2.2). When optimizations are considered in the
   future, the possibility of packing multiple prefixes into an update
   can also 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 -- multiple prefixes in
   a BGP update but with a separate signature for each prefix -- is that
   the resulting BGP update violates the basic definition of a BGP
   update: the different prefixes will have different signatures and
   Expire Time attributes, 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).

5.2.2. Discussion

   Some notes to assist in future optimization discussions follow:

   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 the "signature per update" technique
   is used, then there are potential 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)
   [Secure-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 know and verify what was announced and signed by prior
   ASes 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 of 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.

5.3. Maximum BGPsec Update PDU Size

   The current BGP update message PDU size is limited to 4096 bytes
   [RFC4271]. The question was raised as to whether or not BGPsec would
   require a larger update PDU size.

5.3.1. Decision

   The current thinking is that the maximum PDU size should be increased
   to 64 KB [BGP-Ext-Msg] so that there is sufficient room to
   accommodate two Signature_Blocks (i.e., one block with a current
   algorithm and another block with a new signature algorithm during a
   future transition period) for long AS paths.

Note: RFC 8205 states the following: "All BGPsec UPDATE messages MUST
conform to BGP's maximum message size. If the resulting message
exceeds the maximum message size, then the guidelines in Section 9.2
of RFC 4271 [RFC4271] MUST be followed."

5.3.2. Discussion

The current maximum message size for BGP updates is 4096 octets. An
effort is underway in the IETF to extend it to a larger size
[BGP-Ext-Msg]. BGPsec will conform to whatever maximum message size
is available for BGP while adhering to the guidelines in Section 9.2
of RFC 4271 [RFC4271].

Note: Estimates for the average and maximum sizes anticipated for
BGPsec update messages are provided in [MsgSize].

5.4. Temporary Suspension of Attestations and Validations

5.4.1. Decision

If a BGPsec-capable router needs to temporarily suspend/defer signing
and/or validation of BGPsec updates during periods of route processor
overload, the router may do so even though such suspension/deferment
is not desirable; the specification does not forbid it. Following
any temporary suspension, the router should subsequently 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 possible that a BGPsec router
temporarily pauses performing the validation or signing of updates.
When the workload eases, the BGPsec router should clear the
validation or signing backlog and send 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).

A BGPsec-capable AS may request (out of band) that a BGPsec-capable peer AS never downgrade a signed update to an unsigned update. However, in partial-deployment scenarios, it is not possible for a BGPsec router to require a BGPsec-capable eBGP peer to send only signed updates, except for prefixes originated by the peer's AS.

Note: If BGPsec has not been negotiated with a peer, then a BGPsec router forwards only unsigned updates to that peer; the sending router does so by following the reconstruction procedure in Section 4.4 of [RFC8205] to generate an AS_PATH attribute corresponding to the BGPsec_PATH attribute in a received signed update. If the above-mentioned temporary suspension is ever applied, then the same AS_PATH reconstruction procedure should be utilized.

6. Incremental Deployment and Negotiation of BGPsec

6.1. Downgrade Attacks

6.1.1. Decision

No attempt will be made in the 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 to detect 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 include two instances of this capability (one for each address family) during BGPsec capability negotiation.

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.
   Renegotiation on the fly to add a new address family (i.e., without
   changeover to a 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 traditional BGP with that peer, i.e., it will
   not send BGPsec (i.e., signed) updates to the peer.

   Note: See Section 7.9 of [RFC8205] for a discussion of incremental /
   partial-deployment considerations. Also, Section 6 of [RFC8207]
   describes how edge sites (stub ASes) can sign updates that they
   originate but can receive only unsigned updates. This facilitates a
   less expensive upgrade to BGPsec in resource-limited stub ASes and
   expedites incremental deployment.

6.3.2. Discussion

   The partial-deployment approach to incremental deployment will result
   in "BGPsec islands". 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 to form larger BGPsec islands.

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.

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 to 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
reintroduces the following attack vulnerability: if a BGPsec speaker
is allowed to sign an unsigned update and if signed (i.e., partially
or fully signed) updates would be preferred over unsigned updates,
then a faulty, misconfigured, or subverted BGPsec speaker can
manufacture any unsigned update it wants (by inserting 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
asymmetrically negotiate the use of BGPsec. Thus, a stub AS (or
downstream customer AS) can agree to perform BGPsec only in the
transmit direction and speak traditional BGP 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 hardware-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 the 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 the 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 the prefix.

2. Another type of arrangement that can also be effected outside of
    the BGPsec specification is as follows. The stub AS does not
    sign updates, but it obtains an RPKI (CA) certificate and 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 of the stub AS using the private key obtained from the stub AS. This arrangement is called "proxy signing" (see Section 6.6).

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., the 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 the ROA (in its current form) when a Provider-Independent (PI) address is originated from a provider's AS. This approach was rejected due to possible complications with the 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 this approach, i.e., it must detect that 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. The 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 approach based on the extended ROA). But this approach also requires the creation of a new RPKI object, namely, the Proxy Signer Authorization. In this approach, the second-hop AS and validating ASes have 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). An advantage for a stub AS that does asymmetric BGPsec is that it only needs to minimally upgrade to BGPsec so it can sign updates to its upstream AS 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 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 of the customer AS using the private key
provided by the customer AS. This is a private arrangement between
the two ASes 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 multihomed 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 a proxy-signing service from both of its
upstream ASes. In the future, if AS8 were to disconnect 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

In traditional BGP, multiple peering sessions between different pairs
of routers (between two neighboring ASes) may be simultaneously used
for load sharing. Similarly, 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
may 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).

This decision regarding load balancing (vs. using one peering session
as the primary for carrying data and another as the 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 traditional BGP, 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 of ASes (and maybe as
   many as 300) 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 router processing can make use of peer groups preceding the
   signing of updates to peers. 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 traditional BGP. 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, information about the
   forward AS set that constitutes a peer group would have to be
   divulged (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 any message that is
   signed in BGPsec.

7.2.2. Discussion

   From a security standpoint, the community attribute, as currently
   defined, may be inherently defective. A substantial amount of work
   on the semantics of the community attribute is needed, 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
any message that 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 this
flag could potentially be used to indicate to BGPsec whether or not
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 by necessity carry BGPsec data and perform eBGPsec functions.
Confederations are eBGP clouds for administrative purposes and
contain multiple Member-ASes. A Member-AS is not required to sign
updates sent to another Member-AS within the same confederation.
However, if BGPsec signing is applied in eBGP within a confederation,
i.e., each Member-AS signs to the next Member-AS in the path within
the confederation, then upon egress from the confederation, the
Member-AS at the boundary must remove any and all signatures applied
within the confederation. The Member-AS at the boundary of the
confederation will sign the update to an eBGPsec peer using the
public ASN of the confederation and its private key. The BGPsec
specification will not specify how to perform this process.

Note: In RFC 8205, signing a BGPsec update between Member-ASes within
a confederation is required if the update were to propagate with
signatures within the confederation. A Confed_Segment flag exists in
each Secure_Path segment, and when set, it indicates that the
corresponding signature belongs to a Member-AS. At the confederation
boundary, all signatures with Confed_Segment flags set are removed
from the update. RFC 8205 specifies in detail how all of this is
done. Please see Figure 5 in Section 3.1 of [RFC8205], as well as
Section 4.3 of [RFC8205], for details.

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-Initial] made no special provisions to accommodate route
   servers in Internet Exchange Points (IXPs).

   Note: The above decision subsequently changed: RFC 8205 allows the
   accommodation of IXPs, especially for transparent route servers. The
   pCount (AS prepend count) field is set to zero for transparent route
   servers (see Section 4.2 of [RFC8205]). The operational guidance for
   preventing the misuse of pCount=0 is given in Section 7.2 of
   RFC 8205. Also, see Section 8.4 of RFC 8205 for a discussion of
   security considerations concerning pCount=0.

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 peering with a route server at the IXP (without
   the 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: The 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 the 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 (this data is based on private
   communication with IXPs circa 2011). 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 interconnections between n peers in Method A.

   To the extent that 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.

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

7.5.1. Decision

   Proxy aggregation (i.e., the use of AS_SETs in the AS path) will not
   be supported in BGPsec. 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 consider the update
   malformed (i.e., a protocol error).

   Note: Section 5.2 of RFC 8205 specifies that a receiving BGPsec
   router "MUST handle any syntactical or protocol errors in the
   BGPsec_PATH attribute by using the 'treat-as-withdraw' approach as
   defined in RFC 7606 [RFC7606]."

7.5.2. Discussion

   Proxy aggregation does occur in the Internet today, but it is very
   rare. Only a very small fraction (about 0.1%) of observed updates
   contain AS_SETs in the AS path [ASset]. Since traditional BGP
   currently allows for proxy aggregation with the inclusion of AS_SETs
   in the AS path, it is necessary that BGPsec specify what action a
   receiving router must take if such an update is received with
   attestation. BCP 172 [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 [RFC6793]. The standard mechanism used to accommodate
   such speakers requires a peer AS to translate each 4-byte ASN in the
   AS path to a reserved 2-byte ASN (23456) before forwarding the
   update. This mechanism is incompatible with the 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 the 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., the 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; the following steps are included in [BGPsec-Initial]. However, the ordering of these validation-processing steps is not a normative part of the BGPsec specification.

1. Verify that the signed update is syntactically correct. For example, check to see if the number of signatures matches 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 cryptographic 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-cryptographic operation and is fast.

5. Validate the signature data starting from the most recent AS to the origin.

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

7. 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.

8. Repeat steps 5 and 6 for each preceding signature in the Signature_Block until (a) the signature data for the origin AS is encountered and processed or (b) either of these steps fails.

Note: Significant refinements to the above list occurred in the
progress towards RFC 8205. The detailed syntactic-error checklist is
presented and explained in Section 5.2 of [RFC8205]. Also, a logical
sequence of steps to be followed in the validation of
Signature_Blocks is described in Section 5.2 of [RFC8205].

8.1.2. Discussion

If the goal is to minimize computational costs associated with
cryptographic operations, the sequence of receiver operations that is
suggested above is viewed as appropriate. One additional interesting
suggestion was that when there are two Signature_Blocks in an update,
the validating router can first verify which of the two algorithms is
cheaper, to save on processing. If that Signature_Block verifies,
then the router can skip validating the other Signature_Block.

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).

8.2.2. Discussion

The availability of RPKI data at different routers (in the same AS 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. Also, 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 that will identify some update-failure
conditions that may present unambiguous cases for rejecting the
update (in which case the router would not select the AS path in the
update). These cases are "TBD" (to be determined).

8.3. Enumeration of Error Conditions

Enumeration of error conditions and the recommendations for how to
react 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.

Note: Section 5.2 of RFC 8205 describes the detection of syntactic
and protocol errors in BGPsec updates as well as how the updates with
such errors are to be handled.

8.3.2. Discussion

The following list is a first attempt to provide some possible error
conditions and recommended receiver reactions in response to the
detection of those errors. Refinements will follow after further
discussions.

E1 Abnormalities where a peer (i.e., the preceding AS) should
    definitely not have propagated to a receiving eBGPsec router.
    For example, (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, or (C) other syntactic errors with
    signatures have occurred.

    Reaction: See Section 8.5.

E2 Situations where a receiving eBGPsec router cannot find the
    certificate for an AS in the AS path.

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

E3 Situations where a receiving eBGPsec router cannot find a ROA for
    the {prefix, origin} pair in the update.

    Reaction: Same as in (E2) above.

E4 Situations where 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.

In some networks, the best-path-selection policy may specify
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 being
originated from within the AS in question.

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.

8.4.2. Discussion

It is also possible that an update received in IGP (or iBGP) may have
private ASNs in the AS path. These private ASNs would normally
appear in the rightmost portion of the AS path. It was noted that in
this case the private ASNs to the right would be removed (as done in
traditional BGP), and then the update will be signed by the
originating AS and announced to BGPsec-capable eBGP peers.

Note: See Section 7.5 of [RFC8205] for operational considerations for
BGPsec in the context of private ASNs.

8.5. Response to Syntactic Errors in Signatures and Recommendations for How to React to Them

Note: The contents of this subsection (i.e., Section 8.5) differ
substantially from the recommendations in RFC 8205 regarding the
handling of syntactic errors and protocol errors. Hence, the reader
may skip this subsection and instead read Section 5.2 of [RFC8205].
This subsection (Section 8.5) is kept here for the sake of archival
value concerning design discussions.

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
   BGPsec_PATH both appearing in an update, (B) the number of signatures
   not matching the number of ASes (after accounting for any AS
   prepending), or (C) a parsing issue occurring with the BGPsec_PATH
   attribute, then the update (with the signatures stripped) will still
   be considered in the best-path-selection algorithm. (**Note: This is
   not true in RFC 8205**.) 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_Blocks in an update, and one
   or more syntactic errors are found to occur within one of them but
   the other one is free of any syntactic errors, then the update will
   still be considered in the best-path-selection algorithm after the
   syntactically bad Signature_Block has been removed. (**Note: This is
   not true in RFC 8205**.) If the update is selected as the best path,
   then the update will be propagated with only one (i.e., the
   error-free) Signature_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
   a full BGP table dump occurs, then would there be 500K such errors
   resulting in 500K "notify" messages sent to the peer that is
   generating the errors? Answer: Rate limiting would be applied to the
   notify messages and should prevent any overload due to these
   messages.

8.6. Enumeration of Validation States

   Various validation conditions are possible that can be mapped to
   validation states for possible input to the BGPsec decision process.
   These conditions can be related to whether an update is signed,
   Expire Time is checked, route origin validation is checked against a
   ROA, signature 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, route origin and Signature_Block
   validation) and (2) Invalid -- all other possibilities. "Invalid"
   would include situations such as the following:

   1. Due to a lack of RPKI data or insufficient RPKI data, validation
       was not performed.

   2. The signature Expire Time check failed.

   3. Route origin validation failed.

   4. Signature checks were performed, and one or more of them failed.

   Note: Expire Time is obsolete (see the notes in Sections 2.2.1 and
   2.2.2). RFC 8205 uses the states "Valid" and "Not Valid", but only
   with respect to AS path validation (i.e., not including the result of
   origin validation); see Section 5.1 of [RFC8205]. "Not Valid"
   includes all conditions in which path validation was attempted but a
   "Valid" result could not be reached. (Note: Path validation is not
   attempted in the case of syntactic or protocol errors in a BGPsec
   update; see Section 5.2 of [RFC8205].) Each Relying Party (RP) is
   expected to devise its own policy to suitably factor the results of
   origin validation [RFC6811] and path validation [RFC8205] into its
   path-selection decision.

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's local
   policy ultimately has control over what action (regarding BGP path
   selection) is taken.

   Initially, four validation states were considered:

   1. The update is not signed.

   2. The update is signed, but the router does not have corresponding
       RPKI data to perform a validation check.

   3. The validation check was performed, and the check failed
       (Invalid).

   4. The validation check was performed, and the check passed (Valid).

As stated above, it was later decided that BGPsec validation outcomes will be mapped to one of only two validation states. 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 will not likely be constructive in route-selection decisions and may even introduce new vulnerabilities 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 an update is not signed but the {prefix, origin} pair matches the ROA information? After some discussion, a tentative conclusion was reached: this is in principle similar to validation based on partial path signing (which was ruled out; see Section 6.4). So, there is no need to add another validation state for this case; treat it as "Invalid", considering that it is unsigned.

Another remaining question: Would the RP want to give the update a higher preference over another unsigned update that failed origin 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 how the validation status is conveyed internally, using various preexisting 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 router to the egress router, where it is signed when propagated towards other eBGPsec speakers in neighboring ASes. 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.

Note: An extended community for carrying the origin validation state in iBGP has been specified in RFC 8097 [RFC8097].

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., sibling ASes). 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 to be sure of the actual state of
   the update's validation. Question: Should validation-state
   propagation be protected by attestation in cases where it is useful
   for diagnostics purposes? The decision was made to not protect the
   validation-state information using signatures.

   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
      (route origin validation result): (1) Valid, (2) Invalid,
      (3) NotFound (see [RFC6811]), (4) Validation Deferred.

      * An update could be unsigned for either of the following two
         reasons, but they need not be distinguished: (a) it had no
         signatures (i.e., came in unsigned from an eBGP peer) or
         (b) signatures were present but stripped.

   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 illustrated 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. An egress
   edge Router B, upon receiving the update from ingress Router A, would
   then be able to perform its own validation (origin validation for an
   unsigned update or origin/signature validation for a signed update).
   As stated before, the egress router (Router B in this example) may
   always choose to perform its own validation when it receives an
   update from iBGP (independently of the update's validation status
   conveyed in iBGP) to account for the possibility of RPKI data skew at
   different routers. These various choices are local and entirely at
   the operator's discretion.

9. Operational Considerations

   Note: Significant thought has been devoted to operations and
   management considerations subsequent to the writing of
   [BGPsec-Initial]. The reader is referred to [RFC8207] and Section 7
   of [RFC8205] for details.

9.1. Interworking with BGP Graceful Restart

   BGP Graceful Restart (BGP-GR) [RFC4724] is a mechanism currently used
   to facilitate nonstop packet forwarding when the control plane is
   recovering from a fault (i.e., the BGP session is restarted) but the
   data plane is functioning. Two questions were raised: Are there any
   special concerns about how BGP-GR works while BGPsec is operational?
   Also, what happens if the BGP router operation transitions from
   traditional BGP operation to BGP-GR to BGPsec, in that order?

9.1.1. Decision

   No decision was made relative to this issue (at the time that
   [BGPsec-Initial] was written).

   Note: See Section 7.7 of [RFC8205] for comments concerning the
   operation of BGP-GR with BGPsec. They are consistent with the
   discussion below.

9.1.2. Discussion

   BGP-GR can be implemented with BGPsec, just as it is currently
   implemented with traditional BGP. The Restart State bit, Forwarding
   State bit, End-of-RIB marker, staleness marker (in the Adj-RIB-In),
   and Selection_Deferral_Timer are key parameters associated with
   BGP-GR [RFC4724]. These parameters would apply to BGPsec, just as
   they apply to traditional BGP.

   Regarding what happens if the BGP router transitions from traditional
   BGP to BGP-GR to BGPsec, the answer would simply be as follows. If
   there is a software upgrade to BGPsec during BGP-GR (assuming that
   the upgrade is being done on a live BGP speaker), then the BGP-GR
   session should be terminated before a BGPsec session is initiated.
   Once the eBGPsec peering session is established, 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 unsigned BGP 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 unsigned BGP
routes from that sending speaker as "stale" in its Adj-RIB-In. Then,
as BGPsec updates are received (possibly interspersed with unsigned
BGP updates), 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

Work related to this topic is still in progress.

9.2.2. Discussion

BCP recommendations for minimizing churn in BGPsec have been
discussed. There are various potential strategies on how routers
should react to such events as 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 an ingress router, one needs
the Adj-RIB-In entries validated but not the RIB-out entries. So,
the off-router box is probably unlike the traditional route
reflector; it sits at the network 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 Adj-RIB-In. Question: What
about failure modes here? The failure modes would be dependent on
the following:

1. How much of the control plane is outsourced.

2. How reliable the off-router box is (or, equivalently,
    communication to and from it).

3. How centralized vs. distributed this arrangement is.

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 (Provider Edge (PE) routers and
route reflectors) will require significantly upgraded hardware --
much more memory for RIBs and hardware cryptographic 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 unsigned
BGP updates (see Section 6.5).

9.4.2. Discussion

It is accepted that it might take several years to go beyond test
deployment of BGPsec because of the need for additional route
processor CPU and memory. 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 [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 in the early phases of
deployment.

Note: There are recent detailed studies that considered software
optimizations for BGPsec. In [Mehmet1] and [Mehmet2], computational
optimizations for cryptographic processing (i.e., ECDSA speedup) are
considered for BGPsec implementations on general-purpose CPUs. In
[V_Sriram], software optimizations at the level of update processing
and path selection are proposed and quantified for BGPsec
implementations.

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: such solutions were not pursued
   because they cannot provide strong guarantees regarding the validity
   of updates. Using these techniques, one can say only that an update
   is "plausible"; one cannot say that it is "definitely" valid (based
   on signed peering relations alone).

10. Security Considerations

   This document requires no security considerations. See [RFC8205] for
   security considerations for the BGPsec protocol.

11. IANA Considerations

   This document has no IANA actions.

12. 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>.

   [BGP-Ext-Msg]
                Bush, R., Patel, K., and D. Ward, "Extended Message
                support for BGP", Work in Progress, draft-ietf-idr-bgp-
                extended-messages-24, November 2017.

   [BGPsec-Initial]
                Lepinski, M., "BGPSEC Protocol Specification", Work in
                Progress, draft-lepinski-bgpsec-protocol-00, March 2011.

   [BGPsec-Rollover]
                Weis, B., Gagliano, R., and K. Patel, "BGPsec Router
                Certificate Rollover", Work in Progress, draft-ietf-
                sidrops-bgpsec-rollover-04, December 2017.

   [Borchert]
              Borchert, O. and M. Baer, "Subject: Modifiation [sic]
              request: draft-ietf-sidr-bgpsec-protocol-14", message to
              the IETF SIDR WG Mailing List, 10 February 2016,
              <https://www.ietf.org/mail-archive/web/sidr/current/
              msg07509.html>.

   [CiscoIOS]
              "Cisco IOS: Configuring Route Dampening", February 2014,
              <https://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/
              configuration/guide/fipr_c/1cfbgp.html>.

   [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, <https://www.ietf.org/proceedings/
              83/slides/slides-83-sidr-7.pdf>.

   [Dynamics]
              Sriram, K., Montgomery, D., Borchert, O., Kim, O., and P.
              Gleichmann, "Potential Impact of BGPSEC Mechanisms on
              Global BGP Dynamics", Presentation to the BGPsec
              authors/designers team, October 2009,
              <https://www.nist.gov/file/448631>.

   [Gueron]   Gueron, S. and V. Krasnov, "Fast and side channel
              protected implementation of the NIST P-256 Elliptic Curve
              for x86-64 platforms", OpenSSL patch ID 3149,
              October 2013, <https://rt.openssl.org/Ticket/
              Display.html?id=3149&user=guest&pass=guest>.

   [JunOS]   "Juniper JunOS: Using Routing Policies to Damp BGP Route
              Flapping", November 2010, <http://www.juniper.net/
              techpubs/en_US/junos10.4/topics/usage-guidelines/
              policy-using-routing-policies-to-damp-bgp-route-
              flapping.html>.

   [Mandelberg1]
              Mandelberg, D., "Subject: wglc for draft-ietf-sidr-bgpsec-
              protocol-11 (Specific topic: Include Address Family
              Identifier in the data protected under signature -- to
              alleviate a security concern)", message to the IETF SIDR
              WG Mailing List, 10 February 2015, <https://www.ietf.org/
              mail-archive/web/sidr/current/msg06930.html>.

[Mandelberg2]
          Mandelberg, D., "Subject: draft-ietf-sidr-bgpsec-
          protocol-13's security guarantees (Specific topic: Sign
          all of the preceding signed data (rather than just the
          immediate, previous signature) -- to alleviate a security
          concern)", message to the IETF SIDR WG Mailing List,
          26 August 2015, <https://www.ietf.org/mail-archive/
          web/sidr/current/msg07241.html>.

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

[Mehmet1] Adalier, M., "Efficient and Secure Elliptic Curve
          Cryptography Implementation of Curve P-256", NIST Workshop
          on ECC Standards, June 2015,
          <http://csrc.nist.gov/groups/ST/ecc-workshop-2015/papers/
          session6-adalier-Mehmet.pdf>.

[Mehmet2] Adalier, M., Sriram, K., Borchert, O., Lee, K., and D.
          Montgomery, "High Performance BGP Security: Algorithms and
          Architectures", North American Network Operators Group
          Meeting NANOG69, February 2017,
          <https://www.nanog.org/meetings/abstract?id=3043>.

[MsgSize] Sriram, K., "Decoupling BGPsec Documents and Extended
          Messages draft", Presented at the IETF SIDROPS WG
          Meeting, IETF 98, March 2017,
          <https://www.ietf.org/proceedings/98/slides/
          slides-98-sidrops-decoupling-bgpsec-documents-and-
          extended-messages-draft-00.pdf>.

[Replay-Protection]
          Sriram, K. and D. Montgomery, "Design Discussion and
          Comparison of Protection Mechanisms for Replay Attack and
          Withdrawal Suppression in BGPsec", Work in Progress,
          draft-sriram-replay-protection-design-discussion-10,
          April 2018.

[RFC2439] Villamizar, C., Chandra, R., and R. Govindan, "BGP Route
          Flap Damping", RFC 2439, DOI 10.17487/RFC2439,
          November 1998, <https://www.rfc-editor.org/info/rfc2439>.

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

   [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,
              <https://www.rfc-editor.org/info/rfc4724>.

   [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter,
              "Multiprotocol Extensions for BGP-4", RFC 4760,
              DOI 10.17487/RFC4760, January 2007,
              <https://www.rfc-editor.org/info/rfc4760>.

   [RFC6090] McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic
              Curve Cryptography Algorithms", RFC 6090,
              DOI 10.17487/RFC6090, February 2011,
              <https://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,
              <https://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, <https://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,
              <https://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,
              <https://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,
              <https://www.rfc-editor.org/info/rfc6487>.

   [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet
              Autonomous System (AS) Number Space", RFC 6793,
              DOI 10.17487/RFC6793, December 2012,
              <https://www.rfc-editor.org/info/rfc6793>.

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

   [RFC7132] Kent, S. and A. Chi, "Threat Model for BGP Path Security",
              RFC 7132, DOI 10.17487/RFC7132, February 2014,
              <https://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,
              <https://www.rfc-editor.org/info/rfc7353>.

   [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K.
              Patel, "Revised Error Handling for BGP UPDATE Messages",
              RFC 7606, DOI 10.17487/RFC7606, August 2015,
              <https://www.rfc-editor.org/info/rfc7606>.

   [RFC8097] Mohapatra, P., Patel, K., Scudder, J., Ward, D., and R.
              Bush, "BGP Prefix Origin Validation State Extended
              Community", RFC 8097, DOI 10.17487/RFC8097, March 2017,
              <https://www.rfc-editor.org/info/rfc8097>.

   [RFC8205] Lepinski, M., Ed., and K. Sriram, Ed., "BGPsec Protocol
              Specification", RFC 8205, DOI 10.17487/RFC8205,
              September 2017, <https://www.rfc-editor.org/info/rfc8205>.

   [RFC8207] Bush, R., "BGPsec Operational Considerations", BCP 211,
              RFC 8207, DOI 10.17487/RFC8207, September 2017,
              <https://www.rfc-editor.org/info/rfc8207>.

   [RFC8208] Turner, S. and O. Borchert, "BGPsec Algorithms, Key
              Formats, and Signature Formats", RFC 8208,
              DOI 10.17487/RFC8208, September 2017,
              <https://www.rfc-editor.org/info/rfc8208>.

   [RFC8209] Reynolds, M., Turner, S., and S. Kent, "A Profile for
              BGPsec Router Certificates, Certificate Revocation Lists,
              and Certification Requests", RFC 8209,
              DOI 10.17487/RFC8209, September 2017,
              <https://www.rfc-editor.org/info/rfc8209>.

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

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

   [Secure-BGP]
              Lynn, C., Mikkelson, J., and K. Seo, "Secure BGP (S-BGP)",
              Work in Progress, draft-clynn-s-bgp-protocol-01,
              June 2003.

   [V_Sriram]
              Sriram, V. and D. Montgomery, "Design and analysis of
              optimization algorithms to minimize cryptographic
              processing in BGP security protocols", Computer
              Communications, Vol. 106, pp. 75-85,
              DOI 10.1016/j.comcom.2017.03.007, July 2017,
              <https://www.sciencedirect.com/science/article/pii/
              S0140366417303365>.

Acknowledgements

Contributors

   The following people made significant contributions to this document
   and should be considered co-authors:

   Rob Austein
   Dragon Research Labs
   Email: sra@hactrn.net

   Steven Bellovin
   Columbia University
   Email: smb@cs.columbia.edu

   Russ Housley
   Vigil Security, LLC
   Email: housley@vigilsec.com

    Stephen Kent
    Independent
    Email: kent@alum.mit.edu

    Warren Kumari
    Google
    Email: warren@kumari.net

    Matt Lepinski
    New College of Florida
    Email: mlepinski@ncf.edu

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

    Chris Morrow
    Google, Inc.
    Email: morrowc@google.com

    Sandy Murphy
    Parsons, Inc.
    Email: sandy@tislabs.com

    Keyur Patel
    Arrcus
    Email: keyur@arrcus.com

    John Scudder
    Juniper Networks
    Email: jgs@juniper.net

    Samuel Weiler
    W3C/MIT
    Email: weiler@csail.mit.edu

Author's Address

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

    Email: ksriram@nist.gov