Independent Submission
Internet-Draft
Intended status: Informational
Expires: July 22, 2018

K. Sriram, Ed.
USA NIST

January 18, 2018

BGPsec Design Choices and Summary of Supporting Discussions
draft-sriram-bgpsec-design-choices-15

Abstract

This document is written to capture the design rationale primarily for the individual draft-00 version of BGPsec protocol specification [I-D.lepinski-bgpsec-protocol]. However, where appropriate, the document also provides brief notes on design decisions that changed (relative to individual draft-00 version) in the finalized protocol specification [RFC8205]. The notes mention what the differences are and provide pointers to where the details and/or rationale can be found about those changes in design. The document lists the decisions that were made in favor of or against each design choice, and presents brief summaries of the arguments that aided the decision process.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 22, 2018.

Table of Contents

1. Introduction 3
2. Creating Signatures and the Structure of BGPsec Update Messages 4
2.1. Origin Validation Using ROA 4
2.2. Attributes Signed by an Originating AS 5
2.3. Attributes Signed by an Upstream AS 6
2.4. What Attributes Are Not Signed 7
2.5. Receiving Router Actions 8
2.6. Prepending of ASes in AS Path 9
2.7. What RPKI Data Need be Included in Updates 9
3. Withdrawal Protection 10
3.1. Withdrawals Not Signed 10
3.2. Signature Expire Time for Withdrawal Protection (a.k.a. Mitigation of Replay Attacks) 10
3.3. Should Route Expire Time be Communicated in a Separate Message 12
3.4. Effect of Expire-Time Updates in BGPsec on RFD 13
4. Signature Algorithms and Router Keys 14
4.1. Signature Algorithms 14
4.2. Agility of Signature Algorithms 15
4.3. Sequential Aggregate Signatures 16
4.4. Protocol Extensibility 17
4.5. Key Per Router (Rogue Router Problem) 18
4.6. Router ID 18
5. Optimizations and Resource Sizing 18
5.1. Update Packing and Repacking 19
5.2. Signature Per Prefix vs. Signature Per Update 19
5.3. Maximum BGPsec Update PDU Size 20
5.4. Temporary Suspension of Attestations and Validations . . 21
6. Incremental Deployment and Negotiation of BGPsec 22
6.1. Downgrade Attacks 22
6.2. Inclusion of Address Family in Capability Advertisement . 22
6.3. Incremental Deployment: Capability Negotiation 23
6.4. Partial Path Signing 23
6.5. Consideration of Stub ASes with Resource Constraints: Encouraging Early Adoption 24
6.6. Proxy Signing 25
6.7. Multiple Peering Sessions Between ASes 26
7. Interaction of BGPsec with Common BGP Features 26

7.1. Peer Groups 26
7.2. Communities 27
7.3. Consideration of iBGP Speakers and Confederations 28
7.4. Consideration of Route Servers in IXPs 28
7.5. Proxy Aggregation (a.k.a. AS_SETs) 29
7.6. 4-Byte AS Numbers 30
8. BGPsec Validation 30
8.1. Sequence of BGPsec Validation Processing in a Receiver . 30
8.2. Signing and Forwarding Updates when Signatures Failed Validation 32
8.3. Enumeration of Error Conditions 32
8.4. Procedure for Processing Unsigned Updates 33
8.5. Response to Syntactic Errors in Signatures and Recommendation for Reaction 34
8.6. Enumeration of Validation States 35
8.7. Mechanism for Transporting Validation State through iBGP 36
9. Operational Considerations 38
9.1. Interworking with BGP Graceful Restart 38
9.2. BCP Recommendations for Minimizing Churn: Certificate Expiry/Revocation and Signature Expire Time 39
9.3. Outsourcing Update Validation 39
9.4. New Hardware Capability 40
9.5. Signed Peering Registrations 40
10. Security Considerations 41
11. IANA Considerations 41
12. Informative References 41
Acknowledgements 46
Contributors 47
Author's Address 48

1. Introduction

The goal of BGPsec effort is to enhance the security of BGP by enabling full AS path validation based on cryptographic principles. Standards work on route origin validation based on a Resource certificate PKI (RPKI) is already completed or nearing completion in the IETF SIDR WG. The BGPsec effort is aimed at taking advantage of the same RPKI infrastructure developed in the SIDR WG to add cryptographic signatures to BGP updates, so that routers can perform full AS path validation [RFC7132] [RFC7353] [RFC8205]. The BGPsec protocol specification RFC was published recently [RFC8205]. The key high-level design goals of the BGPsec protocol are as follow [RFC7353]:

- Rigorous path validation for all announced prefixes; not merely showing that a path is not impossible.

- Incremental deployment capability; no flag-day requirement for global deployment.

- Protection of AS paths only in inter-domain routing (eBGP); not applicable to iBGP (or to IGPs).

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

This document is a companion to the earliest version of the BGPsec protocol specification submitted as individual draft-00 [I-D.lepinski-bgpsec-protocol], and is intended to provide design justifications for this initial BGPsec specification. However, where appropriate, the document also provides a brief rationale and/or pointer for design decisions that are reflected in the finalized BGPsec protocol specification [RFC8205]. This document lists the decisions that were made in favor of or against various design choices, and presents brief summaries of the discussions that weighed in the pros and cons and aided the decision process.

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

An explanation about "Note:" in this document: Wherever a paragraph starting with "Note:" is seen in this document, it is pointing to a change in design that occurred from the time of submission of the individual draft [I-D.lepinski-bgpsec-protocol] to the time of publication of the finalized BGPsec protocol specification [RFC8205]. The paragraph also provides pointers to where details and/or discussion about the change can be found.

2. Creating Signatures and the Structure of BGPsec Update Messages

2.1. Origin Validation Using ROA

2.1.1. Decision

Route origin validation using Route Origin Authorization (ROA) [RFC6482] [RFC6811] is necessary and complements AS path attestation based on signed updates. Thus, BGPsec design makes use of the origin validation capability facilitated by the ROAs in 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 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 of a signature Expire Time field in the BGPsec update in RFC 8205. 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) in RFC 8205. By adding the AFI in the data covered by signature, a specific security concern was alleviated; see SIDR list post [Mandelberg1] and the discussion thread that followed on the topic. The AFI is obtained from the MP_REACH_NLRI attribute in the BGPsec update. It is stated in Section 4.1 of RFC 8205 that 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 that the AS path cannot be secured; for example, it can be shortened (by a MITM) without being detected.

It was decided that only the originating AS needs to insert a signature Expire Time in the update, as it is the originator of the route. The origin AS also will re-originate, i.e., beacon, the update prior to the Expire Time of 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 against replay attacks and withdrawal suppression [I-D.ietf-sidrops-bgpsec-rollover] [I-D.sriram-replay-protection-design-discussion].

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

The signature data is carried in an optional, non-transitive BGP attribute. The attribute is optional because this is the standard mechanism available in BGP to propagate new types of data. It was decided that the attribute should be non-transitive because of concern about the impact of sending the (potentially large) signatures to routers that don't understand them. Also, if a router that doesn't understand BGPsec somehow gets a message with the signatures attribute then it would be undesirable for that router to forward the signatures to all 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 co-exist 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 (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 for what is signed by an upstream AS as follows. Method 1: Signature protects the combination of NLRI length, NLRI prefix, AS path, next ASN, signature algorithm suite ID, and Expire Time; or Method 2: Signature protects just the combination of previous signature (i.e., signature of the neighbor AS who forwarded the update) and next ASN; or Method 3: Signature protects everything that was received from preceding AS plus next (i.e., target) ASN; thus, ASi signs over NLRI length, NLRI prefix, signature algorithm suite ID, Expire Time, {ASi, AS(i-1), AS(i-2), ..., AS2, AS1}, AS(i+1)(i.e., next ASN), and {Sig(i-1), Sig(i-2), ..., Sig2, Sig1}.

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

2.3.1. Decision

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

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 SIDR list post [Mandelberg2] and the discussion thread that followed on the topic. Also, there is a strong rationale for the sequence octets to be hashed (as shown in Figure 8 in Section 4.2 of RFC 8205) and this sequencing of data is motivated by implementation efficiency considerations; see SIDR list post [Borchert] 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, then the remaining bytes will overflow into a second hash block, which results in performance penalty. So it is advantageous to minimize the number of bytes being hashed. Also, an analysis of the three options noted above did not identify any vulnerabilities associated with this approach.

2.4. What Attributes Are Not Signed

2.4.1. Decision

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

2.4.2. Discussion

The above stated attributes that are not signed are viewed as local (e.g., do not need to propagate beyond next hop) or lack clear security needs. NO-EXPORT is sent over a secured next-hop and does not need signing. BGPsec design should work with any transport layer protections. It is well understood that the transport layer must be

protected hop by hop (if only to prevent malicious session termination).

2.5. Receiving Router Actions

2.5.1. Decision

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

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

- Validate the route origin pair (p, AS1) by performing a ROA match.
- Verify that Te is greater than the clock time at the router performing these checks.
- Check Sig1 with inputs {NLRI length, p, AlgID, Te, AS1, AS2}.
- Check Sig2 with inputs {Sig1, AS3}.
- Check Sig3 with inputs {Sig2, AS4}.
- ...
- ...
- Check Sig(i-2) with inputs {Sig(i-3), AS(i-1)}.
- Check Sig(i-1) with inputs {Sig(i-2), ASi}.
- 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 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 protocol specification uses a pCount field associated with each AS in the path to indicate the number of prepends for that AS (see Figure 5, Section 3.1 of [RFC8205]).

2.6.2. Discussion

The draft-00 version of the protocol specification calls for a signature to be associated with each prepended AS. The optimization of having just one signature for multiple prepended ASes will be pursued later (i.e., beyond draft-00 specification). If such optimization is used, a replication count would be included (in the signed update) to specify how many times an AS was prepended.

2.7. What RPKI Data Need be Included in Updates

2.7.1. Decision

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

2.7.2. Discussion

It was discussed if each router public key certificate should be included in a signed update. Inclusion of this information might be helpful for routers that do not have access to RPKI servers or temporarily lose connectivity to them. It is safe to assume that in majority of network environments, intermittent connectivity would not be a problem. So it is best to avoid this complexity because majority of the use environments do not have connectivity constraints. Because the SKI of a router certificate is a hash of

the public key of that certificate, it suffices to select the public key from that certificate. This design assumes that each BGPsec router has access to a cache containing the relevant data from (validated) router certificates.

3. Withdrawal Protection

3.1. Withdrawals Not Signed

3.1.1. Decision

Withdrawals are not signed.

3.1.2. Discussion

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

Repeated withdrawals and announcements for a prefix can run up the BGP RFD penalty and may result in unreachability for that prefix at upstream routers. But what can the attacker gain from doing so? This phenomenon is intrinsic to the design and operation of RFD.

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

3.2.1. Decision

Note: As mentioned earlier in 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 [I-D.ietf-sidrops-bgpsec-rollover] [I-D.sriram-replay-protection-design-discussion].

Only the originating AS inserts a signature Expire Time in the update; all other ASes along an AS path do not insert Expire Times associated with their respective signatures. Further, the

originating AS will re-originate a route sufficiently in advance of the Expire Time of its signature so that other ASes along an AS path will typically receive the re-originated route well ahead of the current Expire Time for that route.

The duration of the signature Expire Time is recommended to be on the order of days (preferably) but it may be on the order of hours (about 4 to 8 hours) in some cases, where extra replay protection is 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 and additional guidance can be provided via an operational considerations document later. Further, the actual re-origination time should to be jittered with a uniform random distribution over a short interval {Tb1, Tb2} centered at Tb.

It is also recommended that a receiving BGPsec router should detect if the only attribute change in an announcement (relative to the current best path) is the expire time (besides, of course, the signatures). In that case, assuming that the update is found valid, the route processor should not re-announce the route to non-BGPsec peers. (It should sign and re-announce the route to only BGPsec speakers.) 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-advertisement of withdrawn routes. If each AS along a path were to insert its own signature Expire Time, then there would be much additional BGP chattiness and increase in BGP processing load due to the need to detect and react to multiple (possibly redundant) signature Expire Times. Furthermore, there would be no extra benefit from the point of view of mitigation of replay attacks as compared to having a single Expire Time corresponding to the signature of the originating AS.

The recommended Expire Time value is on the order of days but 4 to 8 hours may used in some cases on the basis of 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, and thus causes more BGP chatter.) Even a 4 hours 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.

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

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

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

3.3.1. Decision

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

3.3.2. Discussion

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

It was observed that at the edge of the Internet, there are frequent updates that may result from simple situations like BGP session being switched from one interface to another (e.g., from primary to backup) between two peering ASes (e.g., customer and provider). With 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, and hence the update will need to propagate throughout the Internet (limited only by best path selection process). It was accepted that this cost of added churn will be unavoidable.

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

3.4.1. Decision

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

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

3.4.2. Discussion

Summary:

The decision is supported by the following observations: (1) Expire Time-triggered updates are generally not preceded by withdrawals, and hence the path hunting and associated RFD exacerbation [Mao02][RIPE580] problems are not anticipated; (2) Such updates would not normally change the best path (unless another concurrent event impacts the best path); (3) Expire Time-triggered updates would have negligible impact on RFD penalty accumulation because the re-advertisement interval is much longer relative to the half-time of decay of RFD penalty. Elaborating further on reason #3 above, it may be noted that the re-advertisements (i.e., beacons) of a route for a given address prefix from a given peer will be received at intervals of a few or several hours (see Section 3.2). During that time period, any incremental contribution to RFD penalty due to a Expire Time-triggered update would decay sufficiently to have negligible (if any) impact on damping the address prefix in consideration. Additional details of this analysis and justification can be found below.

Further Details of the Analysis and Justification:

The frequency with which RFD penalty increments may be triggered for a given prefix from a given peer is the same as the re-beaconing frequency for that prefix from its origin AS. The re-beaconing frequency is on the order of once every few or several hours (see Section 3.2). The incremental RFD penalty assigned to a prefix due to a re-beaconed update varies depending on the implementation. For example, it appears that JunOS implementation [JunOS] would assign a penalty of 1000 or 500 depending on whether the re-beaconed update is

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

4. Signature Algorithms and Router Keys

4.1. Signature Algorithms

4.1.1. Decision

Initially, 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 ECDSA with Curve P-256 and SHA-256.

4.1.2. Discussion

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

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

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

Note: Recently, even open source implementations have incorporated certain cryptographic optimizations and demonstrated significant performance speedup [Gueron]. Researchers continue to devote significant efforts to demonstrate substantial speedup for 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., current algorithm, to the next (new) algorithm, the updates will carry two sets of signatures (i.e., two Signature-List Blocks), one corresponding to each algorithm. Each Signature-List Block will be preceded by its type-length field and an algorithm-suite identifier. A BGPsec speaker that has been upgraded to handle the new algorithm should validate both Signature-List Blocks, and then add its corresponding signature to each Signature-List Block for forwarding the update to the next AS. A BGPsec speaker that has not been upgraded to handle the new algorithm will strip off the Signature-List Block of the new algorithm, and forward the update after adding its own sig to the Signature-List Block of the current algorithm.

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

Note: Signature-List Block is Signature_Block in RFC 8205. 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-List Block allows for delineation of the two signature blocks. Hence, a BGPsec router that doesn't know about a particular algorithm suite (and hence doesn't know how long signatures were for that algorithm suite) could still skip over the corresponding Signature-List Block when parsing the message.

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

The lifetime of a signature algorithm is anticipated to be much longer than the duration of a transition period from current to new algorithm. It is fully expected that all ASes will have converted to the required new algorithm within a certain amount of time that is much shorter than the interval in which a subsequent newer algorithm may be investigated and standardized for BGPsec. Hence, the need for more than two Signature-List Blocks per update is not envisioned.

4.3. Sequential Aggregate Signatures

4.3.1. Decision

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

4.3.2. Discussion

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

SAS schemes exist in the literature, typically based on RSA or equivalent. For SAS with RSA and for the cryptographic strength needed for BGPsec signatures, a 2048-bit signature size (RSA-2048) would be required. However, without SAS, ECDSA with 512-bit signature (256-bit key) would suffice for equivalent cryptographic strength. The larger signature size of RSA used with SAS undermines the advantages of SAS, because the average hop count, i.e., number of ASes, for a route is about 3.8. In the end, it may turn out that SAS has more complexity and does not provide sufficient savings in PDU size or RIB size to merit its use. Further exploration of this is needed to better understand SAS properties and applicability for

BGPsec. There is also a concern that SAS is not a time-tested cryptographic technique and thus its adoption is potentially risky.

4.4. Protocol Extensibility

There is a clearly a need to specify a transition path from a current protocol specification to a new version. When changes to the processing of the BGPsec path signatures are required, that will require a new version of BGPsec. Examples of this include changes to the data that is protected by the BGPsec signatures or adoption of a signature algorithm in which the number of signatures in the signature block may not correspond to one signature per AS in the AS-PATH (e.g., aggregate signatures).

4.4.1. Decision

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

Note: Signature-List Block is Signature_Block in RFC 8205. 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 BGPsec capability exchange during the BGPsec session negotiation.

4.4.2. Discussion

In the case that change to BGPsec is deemed desirable, it is expected that a subsequent version of BGPsec would be created and that this version of BGPsec would specify a new BGP Path Attribute, let's call it BGPsec_PATH_SIG_TWO, which is designed to accommodate the desired changes to BGPsec. At this point a transition would begin which is analogous to the algorithm transition discussed in Section 4.2.

During the transition period, all BGPsec speakers will simultaneously include both the BGPsec_Path_Signatures (current) attribute and the new BGPsec_PATH_SIG_TWO attribute. Once the transition is complete, the use of BGPsec_Path_Signatures could then be deprecated, at which point BGPsec speakers will include only the new BGPsec_PATH_SIG_TWO attribute. Such a process could facilitate a transition to a new BGPsec semantics in a backwards compatible fashion.

4.5. Key Per Router (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 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 CA cert of the AS. The Subject Key Identifier (SKI) in the signature points to the router certificate (and thus the unique public key) of the router that affixed its signature, so that a validating router can reliably identify the public key to use for signature verification.

4.6. Router ID

4.6.1. Decision

The router certificate Subject name will be the string "router" followed by a decimal representation of a 4-byte AS number followed by the router ID. See the current RFCs for preferred standard textual representations for 4-byte ASNs [RFC5396] and router IDs [RFC6891].

4.6.2. Discussion

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

5. Optimizations and Resource Sizing

5.1. Update Packing and Repacking

With traditional BGP protocol [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 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 the complexity that would be otherwise inevitable if the origin had packed and signed multiple prefixes in an update and an upstream AS decided to propagate an update containing only a subset of the prefixes in that update. BGPsec recommendation regarding packing and repacking may be be revisited when optimizations are considered in the future.

5.1.2. Discussion

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

The current decision is to include only one prefix per secured update (see Section 2.2 and Section 2.3). When optimizations are considered in the future, the possibility of packing multiple prefixes into an update can be considered. (Please see Section 5.2 for a discussion of signature per prefix vs. signature per update.) Repacking could be performed if signatures were generated on a per prefix basis. However, one problem regarding this approach, i.e., multiple prefixes in a BGP update but with a separate signature for each prefix, is that the resulting BGP update violates the basic definition of a BGP update. That is because the different prefixes will have different signature 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). Optimizations will be examined later.

5.2.2. Discussion

Some notes to assist in future optimization discussions: In the general case of one signature per update, multiple prefixes may be signed with one signature together with their shared AS path, next ASN, and Expire Time. If signature per update is used, then there are potentially savings in update PDU size as well as RIB memory size. But if there are any changes made to the announced prefix set along the AS path, then the AS where the change occurs would need to insert an Explicit Path Attribute (EPA)[I-D.draft-clynn-s-bgp]. The EPA conveys information regarding what the prefix set contained prior to the change. There would be one EPA for each AS that made such a modification, and there would be a way to associate each EPA with its corresponding AS. This enables an upstream AS to be able to know and to verify what was announced and signed by prior 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 the pros and cons of packing and repacking in the context of signature per prefix vs. signature per update (with packing) have been evaluated. But the current recommendation is for having only one prefix per update (no packing); so there is no need for the EPA attribute.

5.3. Maximum BGPsec Update PDU Size

The current BGP update message PDU size is limited to 4096 bytes [RFC4271]. The question was raised if BGPsec would require a larger update PDU size.

5.3.1. Decision

The current thinking is that the max PDU size should be increased to 64 KB [I-D.ietf-idr-bgp-extended-messages] so that there is sufficient room to accommodate two signature-list blocks (i.e., one block with a current algorithm and another block with a new 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. There is effort underway in the IETF to extend it to a larger size [I-D.ietf-idr-bgp-extended-messages]. BGPsec will conform to whatever maximum message size that 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 that. 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 validation or signing of updates. When the work load 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) a BGPsec-capable peer AS never to 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. For this, the

sending router follows the reconstruction procedure of Section 4.4 in [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 BGPsec design to prevent downgrade attacks, i.e., a BGPsec-capable router sending unsigned updates when it is capable of sending signed updates.

6.1.2. Discussion

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

6.2. Inclusion of Address Family in Capability Advertisement

6.2.1. Decision

It was decided that during capability negotiation, the address family for which the BGPsec speaker is advertising support for BGPsec will be shared using the Address Family Identifier (AFI). Initially, two address families would be included, namely, IPv4 and IPv6. BGPsec for use with other address families may be specified in the future. Simultaneous use of the two (i.e., IPv4 and IPv6) address families for the same BGPsec session will require that the BGPsec speaker must include two instances of this capability (one for each address family) 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. Re-negotiation on the fly to add a new address family (i.e., without changeover to new version number) is desirable.

6.3. Incremental Deployment: Capability Negotiation

6.3.1. Decision

BGPsec will be incrementally deployable. BGPsec routers will use capability negotiation to agree to run BGPsec between them. If a BGPsec router's peer does not agree to run BGPsec, then the BGPsec router will run only 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, see Section 6 of [RFC8207] where it is described that edge sites (stub ASes) can sign updates that they originate but receive only unsigned updates. This facilitates less expensive upgrade to BGPsec in resource-limited stub ASes, and expedites incremental deployment.

6.3.2. Discussion

During partial deployment, there will be BGPsec islands as a result of this approach to incremental deployment. Updates that originate within a BGPsec island will generally propagate with signed AS paths to the edges of that island. As BGPsec adoption grows, the BGPsec islands will expand outward (subsuming non-BGPsec portions of the Internet) and/or pairs of islands may join 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. It was decided that this should not be permitted.

6.4.1. Decision

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

6.4.2. Discussion

Partial path signing (as described above) implies that the AS path is not rigorously protected. Rigorous AS path protection is a key requirement of BGPsec [RFC7353]. Partial path signing clearly re-introduces the following attack vulnerability: If a BGPsec speaker is

allowed to sign an unsigned update, and if signed (i.e., partially or fully signed) updates would be preferred to unsigned updates, then a faulty, misconfigured or subverted BGPsec speaker can manufacture any unsigned update it wants (with insertion of a valid origin AS) and add a signature to it to increase the chance that its update will be preferred.

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

6.5.1. Decision

The protocol permits each pair of BGPsec-capable ASes to negotiate BGPsec use asymmetrically. Thus, a stub AS (or downstream customer AS) can agree to perform BGPsec only in the transmit direction and speak 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 BGPsec specification is as follows. Through a private arrangement (invisible to other ASes), an ISP's AS (upstream AS) can truncate the stub AS (or downstream AS) from the path and sign the update as if the prefix is originating from ISP's AS (even though the update originated unsigned from the customer AS). This way the path will appear fully signed to the rest of the network. This alternative will require the owner of the prefix at the stub AS to issue a ROA for the upstream AS, so that the upstream AS is authorized to originate routes for the prefix.

2. Another type of arrangement that can also be effected outside of the BGPsec specification is as follows. Stub AS does not sign updates but obtains an RPKI (CA) certificate, issues a router certificate under that CA certificate. It passes on the private key for the router certificate to its upstream provider. That ISP (i.e., the second hop AS) would insert a signature on behalf the stub AS using 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., stub AS need not sign the updates). It is recognized that this approach is also authoritative and not trust based. It was observed that the extended ROA is not much different from what is done with ROA (in its current form) when a PI address is originated from a provider's AS. This approach was rejected due to possible complications with creation and use of a new RPKI object, namely, the extended ROA. Also, the validating BGPsec router has to perform a level of indirection with approach, i.e., it must detect if an update is not fully signed and then look for the extended ROA to validate.

4. Another method based on a different form of indirection would be as follows: Customer (stub) AS registers something like a Proxy Signer Authorization, which authorizes the second hop (i.e., provider) AS to sign on behalf of the customer AS using the provider's own key [Dynamics]. This method allows for fully signed updates (unlike the Extended ROA based approach). But this approach also requires the creation of a new RPKI object, namely, the Proxy Signer Authorization. In this approach, the second hop AS 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). A stub AS that does asymmetric BGPsec has the advantage that it needs to minimally upgrade to BGPsec so it can sign updates to its upstream while it receives only unsigned updates. Thus,it can avoid the cost of increased processing and memory needed to perform update validations and to store signed updates in the RIBs, respectively.

6.6. Proxy Signing

6.6.1. Decision

An ISP's AS (or upstream AS) can proxy sign BGP announcements for a customer (downstream) AS provided that the customer AS obtains an RPKI (CA) certificate, issues a router certificate under that CA certificate, and it passes on the private key for that certificate to its upstream provider. That ISP (i.e., the second hop AS) would insert a signature on behalf the customer AS using the private key provided by the customer AS. This is a private arrangement between 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 multi-homed to two ISPs, i.e., AS8 peers with AS1 and AS2 of ISP-1 and ISP-2, respectively. In this case AS8 would have an RPKI (CA) certificate; it issues two separate router certificates (corresponding to AS1 and AS2) under that CA certificate; and it passes on the respective private keys for those two certificates to its upstream providers AS1 and AS2. Thus,AS8 has proxy signing service from both its upstream ASes. In the future, if the customer AS8 disconnects from ISP-2, then it would revoke the router certificate corresponding to AS2.

6.7. Multiple Peering Sessions Between ASes

6.7.1. Decision

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

6.7.2. Discussion

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 as primary for carrying data and another as backup) is entirely local and is up to the two neighboring ASes.

7. Interaction of BGPsec with Common BGP Features

7.1. Peer Groups

In the 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 (maybe as high as 300) of ASes in it.

7.1.1. Decision

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

7.1.2. Discussion

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

7.2. Communities

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

7.2.1. Decision

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

7.2.2. Discussion

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

community attribute needs protection. For now, community attributes will not be secured by BGPsec path signatures.

7.3. Consideration of iBGP Speakers and Confederations

7.3.1. Decision

An iBGP speaker that is also an eBGP speaker, and that executes BGPsec, will necessarily carry BGPsec data and perform eBGPsec functions. Confederations are eBGP clouds for administrative purposes and contain multiple 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 external eBGPsec peer using the public AS number 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 done. Please see Section 3.1 (Figure 5) and Section 4.3 in [RFC8205] for the 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 (individual draft-00) makes no special provisions to accommodate route servers in Internet Exchange Points (IXPs) .

Note: The above decision changed subsequently. RFC 8205 allows accommodation for IXPs, especially for the case of transparent route servers. The pCount (AS prepend count) field is set to 0 for transparent route servers (see Section 4.2 of [RFC8205]). The

operational guidance for preventing misuse of pCount=0 is given in Section 7.2 of RFC 8205. Also, see Section 8.4 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 a peering with a route server at the IXP (without IXP inserting its ASN in the path), and (C) BGP peering with an IXP route server, where the IXP inserts its ASN in the path. (Note: IXP's route server does not change the NEXT_HOP attribute even if it inserts its ASN in the path.) It is very rare for an IXP to use Method C because it is less attractive for the clients if their AS path length increases by one due to the IXP. A measure of the extent of use of Method A vs. Method B is given in terms of the corresponding IP traffic load percentages. As an example, at a major European IXP, these percentages are about 80% and 20% for Methods A and B, respectively (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 inter-connections between n peers in Method A.

To the extent an IXP is providing direct bilateral peering between clients (Method A), that model works naturally with BGPsec. Also, if the route server in the IXP plays the role of a regular BGPsec speaker (minus the routing part for payload) and inserts its own ASN in the path (Method C), then that model would also work well in the BGPsec Internet and this case is trivially supported in BGPsec.

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

7.5.1. Decision

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

Note: In Section 5.2 of RFC 8205, it is specified 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 is it 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 inclusion of AS_SETs in the AS path, it is necessary that BGPsec specify what action a receiving router must take in case such an update is received with attestation. 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 [RFC4893]. 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 use of BGPsec, since the ASN translation is equivalent to a route modification attack and will cause signatures corresponding to the translated 4-byte ASNs to fail validation.

7.6.1. Decision

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

7.6.2. Discussion

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

8. BGPsec Validation

8.1. Sequence of BGPsec Validation Processing in a Receiver

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

8.1.1. Decision

There was agreement that the following sequence of receiver operations is quite meaningful, and are included in the individual draft-00 BGPsec specification [I-D.lepinski-bgpsec-protocol].

However, the ordering of validation processing steps is not a normative part of the BGPsec specification.

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

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

3. Verify that the advertisement has not yet expired.

4. Verify that the target ASN in the signature data matches the ASN of the router that is processing the advertisement. Note that the target ASN check is also a non-crypto operation and is fast.

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-List Block, until the signature data for the origin AS is encountered and processed, or until 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

The suggested sequence of receiver operations described above were discussed and are viewed as appropriate, if the goal is to minimize computational costs associated with cryptographic operations. One additional interesting suggestion was that when there are two Signature-List Blocks in an update, the validating router can first verify whichever of the two algorithms is cheaper to save on

processing. If that Signature-List Block verifies, then the router can skip validating the other Signature-List Block.

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 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 in which some conditions of update failure are identified which may be unambiguous cases for rejecting the update, in which case the router must not select the AS path in the update. These cases are TBD.

8.3. Enumeration of Error Conditions

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

8.3.1. Decision

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

Note: Section 5.2 of RFC 8205 describes 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 list here is a first cut at some possible error conditions and recommended receiver reactions in response to detection of those errors. Refinements will follow after further discussions.

E1 Abnormalities that a peer (i.e., preceding AS) should definitely not have propagated to a receiving eBGPsec router. Examples: (A) The number of signatures does not match the number of ASes in the

AS path (after accounting for AS prepending); (B) There is an AS_SET in the received update and the update has signatures; (C) Other syntactic errors with signatures.

Reaction: See Section 8.5.

E2 Situations where a receiving eBGPsec router cannot find the cert for an AS in the AS_PATH.

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

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

Reaction: Same as in (E2) above.

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

Reaction: Same as in (E2) above.

In some networks, 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 consideration.

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

There is also a possibility that an update received in IGP (or iBGP) may have private AS numbers in the AS path. These private AS numbers would normally appear in the right most portion of the AS path. It was noted that in this case, the private AS numbers 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 [RFC8205] for operational considerations for BGPsec in the context of private AS numbers.

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

Note: The contents in this subsection (i.e., Section 8.5) differ substantially from the syntactic and protocol error handling recommendations for BGPsec in RFC 8205. Hence, the reader may skip reading this subsection and instead read Section 5.2 of [RFC8205]. This section (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 Signature-List Block (or Signature_Block per RFC 8205) both appear in an update, or (b) the number of signatures does not match the number of ASes (after accounting for any AS prepending), or (c) a parsing issue occurs with the BGPsec_Path_Signatures attribute, then the update (with the signatures stripped) will still be considered in the best path selection algorithm (**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-List 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-List 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-List Block. The error condition will be logged locally.

8.5.2. Discussion

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

8.6. Enumeration of Validation States

Various validation conditions are possible which can be mapped to validation states for possible input to BGPsec decision process. These conditions can be related to whether an update is signed, Expire Time checked, route origin validation checked against a ROA, signatures verification passed, etc.

8.6.1. Decision

It was decided that BGPsec validation outcomes will be mapped to one of only two validation states: (1) Valid - passed all validation checks (i.e., Expire Time check, route origin and Signature-List Block validation), and (2) Invalid - all other possibilities. "Invalid" would include situations such as (1) did not perform validation due to lack of or insufficient RPKI data, (2) signature Expire Time check failed, (3) route origin validation failed, and (4) signature checks were performed and one or more of them failed.

Note: Expire Time is obsolete (see the notes in Section 2.2.1 and Section 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 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 in the results from origin validation [RFC6811] and path validation [RFC8205] in 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) Update is not signed; (2) Update is signed but router does not have corresponding RPKI data to perform validation check; (3) Invalid (validation check performed and failed); (4) Valid (validation check performed and passed). Later, it was decided that BGPsec validation outcomes will be mapped to one of only two validation states as stated above. It was observed that an update can be invalid for many different reasons. To begin to differentiate these numerous reasons and to try to enumerate different flavors of the Invalid state is not likely to be constructive in route selection decision, and may even introduce to new vulnerability in the system. However, some questions remain such as the following.

Question: Is there a need to define a separate validation state for the case when update is not signed but {prefix, origin} pair matched with ROA information? This question was discussed, and a tentative conclusion was that this is in principle similar to validation based on partial path signatures and that was ruled out earlier (see Section 6.4). So there is no need to add another validation state for this case; treat it as "Unverified" (i.e., "Invalid") considering that it is unsigned. Questions still remain, e.g., would the relying party want to give the update in consideration 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 pre-existing mechanisms, e.g., setting a BGP community, or modifying a metric value such as Local_Pref or MED. A signed update that cannot be validated (except those with syntax errors) should be forwarded with signatures from the ingress to the egress router, where it is signed when propagated towards other eBGPsec speakers in neighboring 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 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 update's validation. Question: Should validation state propagation be protected by attestation in case it has utility for diagnostics purposes? It was decided not to protect the validation state information using signatures.

The following are intended only as suggestions to be considered by AS operators.

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

- Validation states communicated in iBGP for an unsigned update (route origin validation result): (1) Valid, (2) Invalid, (3) 'Not Found' (see [RFC6811]), (4) Validation Deferred.
  - An update could be unsigned for two reasons but they need not be distinguished: (a) Because it had no signatures (came in unsigned from an eBGP peer), or (b) Signatures were present but stripped.
- Validation states communicated in iBGP for a Signed update: (1) Valid, (2) Invalid, (3) Validation Deferred.

The reason for conveying the additional "Validation Deferred" state may be stated as follows. An ingress edge Router A receiving an update from an eBGPsec peer may not attempt to validate signatures (e.g., in a processor overload situation), and in that case Router A should convey "Validation Deferred" state for that signed update (if selected for best path) in iBGP to other edge routers. Then an egress edge Router B upon receiving the update from ingress Router A would be able to perform its own validation (origin validation for

unsigned update or origin/signature validation for signed update). As stated before, the egress Router B may always choose to perform its own validation when it receives an update from iBGP (independent of the validation status conveyed in iBGP) to account for the possibility of RPKI data skew at different routers. These various choices are local and entirely up to operator discretion.

9. Operational Considerations

Note: Significant thought has been devoted to operations and management considerations post publication of individual draft-00 of BGPsec specification. For this, the reader is referred to [RFC8207] and Section 7 of [RFC8205].

9.1. Interworking with BGP Graceful Restart

BGP Graceful Restart (BGP-GR) [RFC4724] is a mechanism currently used to facilitate non-stop packet forwarding when the control plane is recovering from a fault (i.e., BGP session is restarted), but the data plane is functioning. A question was asked regarding if there are any special concerns about how BGP-GR works while BGPsec is operational? Also, what happens if the BGP router operation transitions from 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 of publication of individual draft-00 of BGPsec specification).

Note: See Section 7.7 of [RFC8205] for comments concerning the operation of Graceful Restart 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 RIB-in), and Selection_Deferral_Timer are key parameters associated with BGP-GR [RFC4724]. These parameters would apply to BGPsec just as they do with 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 software upgrade to BGPsec during BGP-GR (assuming 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, then the receiving eBGPsec speaker will see signed updates from the sending (newly upgraded) eBGPsec speaker. There is no apparent harm (it may, in fact, be desirable) if the receiving speaker continues to use previously-learned 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 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

This is still work in progress.

9.2.2. Discussion

BCP recommendations for minimizing churn in BGPsec have been discussed. There are potentially various strategies on how routers should react to events such as certificate expiry/revocation, signature Expire Time exhaustion, etc. [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 RIB-in entries validated; 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 RIB-in. Question: What about failure modes here? They would be dependent on (1) How much of the control plane is outsourced; (2) Reliability of the off-router box

(or, equivalently communication to and from it); and (3) How centralized vs. distributed is this arrangement? When any kind of outsourcing is done, the user needs to be watchful and ensure that the outsourcing does not cross trust/security boundaries.

9.4. New Hardware Capability

9.4.1. Decision

It is assumed that BGPsec routers (PE routers and route reflectors) will require significantly upgraded hardware - much more memory for RIBs and hardware crypto assistance. However, stub ASes would not need to make such upgrades because they can negotiate asymmetric BGPsec capability with their upstream ASes, i.e., they sign updates to the upstream AS but receive only 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 (see [RIB_size]) for preliminary results on modeling and estimation of BGPsec RIB size and its projected growth. Hardware cryptographic support reduces the computation burden on the route processor, and offers good security for router private keys. However, given the incremental deployment model, it also is not clear how substantial a cryptographic processing load will be incurred, initially.

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. The reason for not pursuing such solutions was that they cannot provide strong guarantees about the validity of updates. Using these techniques, one can say only that an update is 'plausible', but cannot say it is 'definitely' valid (based on signed peering relations alone).

10. Security Considerations

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

11. IANA Considerations

This document includes no request to IANA.

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>.

[Borchert] Borchert, O. and M. Baer, "Subject: Modification 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 RFD implementation", <http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html#wp1002395>.

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

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

[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/>.

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

[I-D.ietf-idr-bgp-extended-messages]
Bush, R., Patel, K., and D. Ward, "Extended Message support for BGP", draft-ietf-idr-bgp-extended-messages-24 (work in progress), November 2017.

[I-D.ietf-sidrops-bgpsec-rollover]
Weis, B., Gagliano, R., and K. Patel, "BGPsec Router Certificate Rollover", draft-ietf-sidrops-bgpsec-rollover-04 (work in progress), December 2017.

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

[I-D.sriram-replay-protection-design-discussion]
Sriram, K. and D. Montgomery, "Design Discussion and Comparison of Protection Mechanisms for Replay Attack and Withdrawal Suppression in BGPsec", draft-sriram-replay-protection-design-discussion-09 (work in progress), October 2017.

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

[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 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 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. and 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 Operator Group Meeting NANOG-69, February 2017, <https://www.nanog.org/meetings/abstract?id=3043>.

[MsgSize] Sriram, K., "Decoupling BGPsec Documents and Extended Messages draft", Presented in 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>.

[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>.

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

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 4055, DOI 10.17487/RFC4055, June 2005, <https://www.rfc-editor.org/info/rfc4055>.

[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>.

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

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

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

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

[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>.

[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>.

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

[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>.

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

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

[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, July 2017, <http://www.sciencedirect.com/science/article/pii/S0140366417303365>.

Acknowledgements

The authors would like to thank Jeff Haas and Wes George for serving as reviewers for this document for the Independent Submissions stream. The authors are grateful to Nevil Brownlee for shepherding this document through the Independent Submissions review process. Many thanks are also due to Michael Baer, Oliver Borchert, David Mandelberg, Sean Turner, Alvaro Retana, Matthias Waehlisch, Tim Polk, Russ Mundy, Wes Hardaker, Sharon Goldberg, Ed Kern, Chris Hall, Shane Amante, Luke Berndt, Doug Maughan, Pradosh Mohapatra, Mark Reynolds, Heather Schiller, Jason Schiller, Ruediger Volk, and David Ward for their review, comments, and suggestions during the course of this work.

Contributors

The following people have 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

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

Russ Housley
Vigil Security
Email: housley@vigilsec.com

Stephen Kent
BBN Technologies
Email: kent@alum.mit.edu

Warren Kumari
Google
Email: warren@kumari.net

Matt Lepinski
New College of Florida
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
SPARTA, Inc., a Parsons Company
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 NIST
100 Bureau Drive
Gaithersburg, MD 20899
USA

Email: ksriram@nist.gov