Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2013

M. Lepinski, Ed.
BBN

September 7, 2012

BGPSEC Protocol Specification
draft-ietf-sidr-bgpsec-protocol-05

Abstract

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

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [8].

Status of this Memo

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

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

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

This Internet-Draft will expire on March 11, 2013.

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

BGPSEC relies on the Resource Public Key Infrastructure (RPKI) certificates that attest to the allocation of AS number and IP address resources. (For more information on the RPKI, see [6] and the documents referenced therein.) Any BGPSEC speaker who wishes to send BGP update messages to external peers (eBGP) containing the BGPSEC_Path_Signatures must have an RPKI end-entity certificate (as well as the associated private signing key) corresponding to the BGPSEC speaker's AS number. Note, however, that a BGPSEC speaker does not require such a certificate in order to validate update messages containing the BGPSEC_Path_Signatures attribute.

2. BGPSEC Negotiation

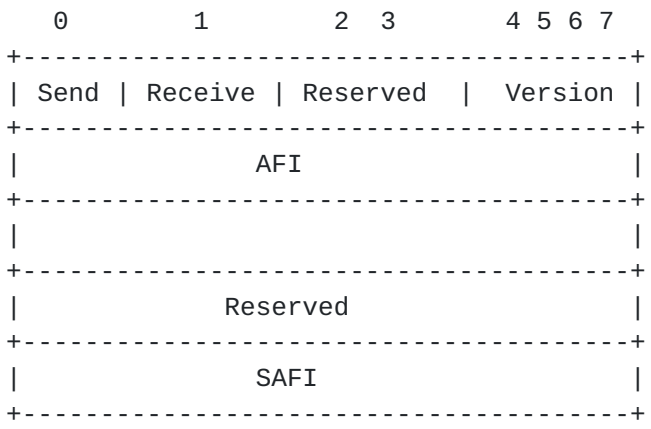
This document defines a new BGP capability [4]that allows a BGP speaker to advertise to its neighbors the ability to send and/or receive BGPSEC update messages (i.e., update messages containing the BGPSEC_Path_Signatures attribute).

This capability has capability code : TBD

The capability length for this capability MUST be set to 5.

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

Capability Value:



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

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

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

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

The second and third octets contain the 16-bit Address Family Identifier (AFI) which indicates the address family for which the BGPSEC speaker is advertising support for BGPSEC. This document only

specifies BGPSEC for use with two address families, IPv4 and IPv6, AFI values 1 and 2 respectively. BGPSEC for use with other address families may be specified in future documents.

The fourth octet in the capability is reserved. It is anticipated that this octet will not be used until such a time as the reserved octet in the Multi-protocol extensions capability advertisement [2] is specified for use. The reserved octet should be set to zero by the sender and ignored by the receiver.

The fifth octet in the capability contains the 8-bit Subsequent Address Family Identifier (SAFI). This value is encoded as in the BGP multiprotocol extensions [2].

Note that if the BGPSEC speaker wishes to use BGPSEC with two different address families (i.e., IPv4 and IPv6) over the same BGP session, then the speaker must include two instances of this capability (one for each address family) in the BGP OPEN message. A BGPSEC speaker SHOULD NOT advertise the capability of BGPSEC support for any <AFI, SAFI> combination unless it has also advertises the multiprotocol extension capability for the same <AFI, SAFI> combination [2].

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

- The BGP speaker understands the BGPSEC_Path_Signatures attribute (see Section 3).

- The BGP speaker supports 4-byte AS numbers (see RFC 4893).

Note that BGPSEC update messages can be quite large, therefore any BGPSEC speaker announcing the capability to receive BGPSEC messages SHOULD also announce support for the capability to receive BGP extended messages [9].

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

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

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

3. The BGPSEC_Path_Signatures Attribute

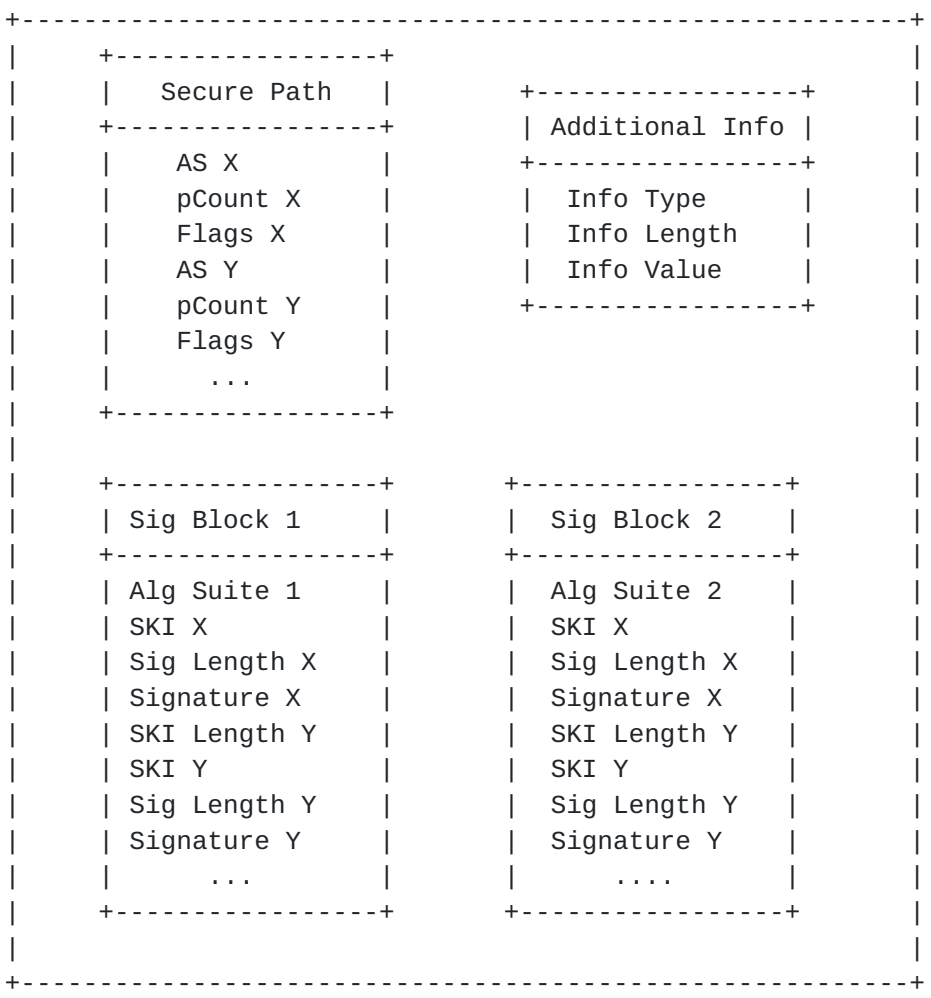
The BGPSEC_Path_Signatures attribute is a new optional (non-transitive) BGP path attribute.

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

The BGPSEC_Path_Signatures algorithm carries the secured AS Path information, including the digital signatures that protect this AS Path information. We refer to those update messages that contain the BGPSEC_Path_Signatures attribute as "BGPSEC Update messages". The BGPSEC_Path_Signatures attribute replaces the AS_PATH attribute, in a BGPSEC update message. That is, update messages that contain the BGPSEC_Path_Signatures attribute MUST NOT contain the AS_PATH attribute.

The BGPSEC_Path_Signatures attribute is made up of several parts. The following high-level diagram provides an overview of the structure of the BGPSEC_Path_Signatures attribute:

High-Level Diagram of the BGPSEC_Path_Signatures Attribute



The following is a more detailed explanation of the format of the BGPSEC_Path_Signatures attribute.

BGPSEC_Path_Signatures Attribute

Secure_Path	(variable)
Additional_Info	(variable)
Sequence of one or two Signature_Blocks	(variable)

The Secure_Path contains AS Path information for the BGPSEC update message. This is logically equivalent to the information that would

be contained in the AS_PATH attribute. A BGPSEC update message containing the BGPSEC_PATH_SIGNATURES attribute MUST NOT contain the AS_PATH attribute. The path information is used by BGPSEC speakers in the same way that information from the AS_PATH is used by non-BGPSEC speakers. The format of the Secure_Path is described below in Section 3.1.

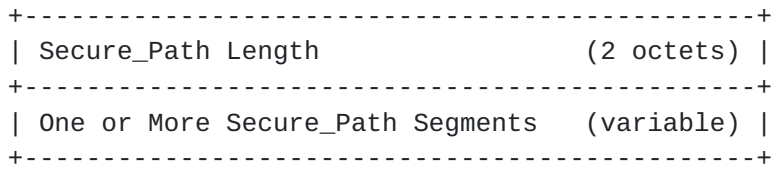
The Additional_Info contains additional signed information about the update message. Additional_Info is specified as a type-length-value field for future extensibility. However, this specification defines only a single (null) type of Additional Info which has zero length. It is anticipated that future specifications may specify semantics for Info Types other than zero. See Section 3.2 below for more detail.

The BGPSEC_Path_Signatures attribute will contain one or two Signature_Blocks, each of which corresponds to a different algorithm suite. Each of the Signature_Blocks will contain a signature segment for each AS number (i.e, secure path segment) in the Secure_Path. In the most common case, the BGPSEC_Path_Signatures attribute will contain only a single Signature_Block. However, in order to enable a transition from an old algorithm suite to a new algorithm suite, it will be necessary to include two Signature_Blocks (one for the old algorithm suite and one for the new algorithm suite) during the transition period. (See Section 6.1 for more discussion of algorithm transitions.) The format of the Signature_Blocks is described below in Section 3.3.

3.1. Secure_Path

Here we provide a detailed description of the Secure_Path information in the BGPSEC_Path_Signatures attribute.

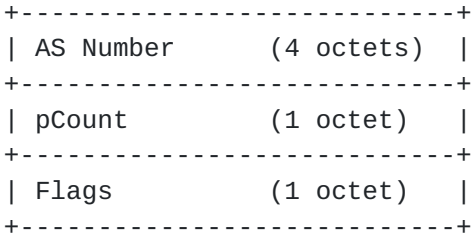
Secure_Path



The Secure_Path Length contains the length (in octets) of the variable-length sequence of Secure_Path Segments. As explained below, each Secure_Path segment is six octets long. Note that this means the Secure_Path Length is six times the number Secure_Path Segments (i.e., the number of AS numbers in the path).

The Secure_Path contains one Secure_Path segment for each (distinct) Autonomous System in the path to the NLRI specified in the update message.

Secure_Path Segment



The AS Number is the AS number of the BGP speaker that added this Secure_Path segment to the BGPSEC_Path_Signatures attribute. (See Section 4 for more information on populating this field.)

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

The first bit of the Flags field is the Confed_Segment flag. The Confed_Segment flag is set to one to indicate that the BGPSEC speaker that constructed this Secure_Path segment is sending the update message to a peer AS within the same Autonomous System confederation [3]. (That is, the Confed_Segment flag is set in a BGPSEC update message whenever in a non-BGPSEC update message the BGP speaker's AS would appear in a AS_PATH segment of type AS_CONFED_SEQUENCE.) In all other cases the Confed_Segment flag is set to zero.

The remaining seven bits of the Flags field are reserved for future use. These bits MUST be set to zero by the sender. The receiver uses the entire Flags octet to verify the digital signature (regardless of what value the reserved bits contain), but otherwise ignores the reserved flags (see Section 4 for sender instructions and Section 5 for receiver validation instructions).

EDITOR'S NOTE: The unused portion of the signed flags field provides the possibility of adding in the future (in a backwards compatible fashion) a new feature that requires some per-AS signed bits. For example, one could use a couple bits from this flag field to mark some other property (besides being in the same confederation) of the connection between two peer ASes.

3.2. Additional_Info

Here we provide a detailed description of the Additional_Info in the BGPSEC_Path_Signatures attribute.

Additional_Info

The Info Type field is a one-octet value that identifies the type of additional information included in the Info Value field. This specification defines a single (null) type of Additional_Info. The Info Type for this null type is zero.

The Info Length field contains the length in octets of the Info Value field. For the (null) Info Type zero specified in this document, the Info Length MUST be zero.

The syntax and semantics contained in the Info Value field depends on the type contained in the Info Type field. For the (null) Info Type zero specified in this document, the Info Value field is empty (since the Info Length field must be zero).

Implementations compliant with this specification MUST set the Info Type to zero in BGPSEC update messages for route advertisements that they originate (see Section 4.1 for more details). When an implementation compliant with this specification receives a BGPSEC update message with an Info Type field that it does not understand (i.e., an Info Type other than zero), the implementation MUST use the Additional_Info when it verifies digital signatures (as per Section 5.2). However, other than signature verification, the implementation MUST ignore the Info Value field when it does not understand the Info Type.

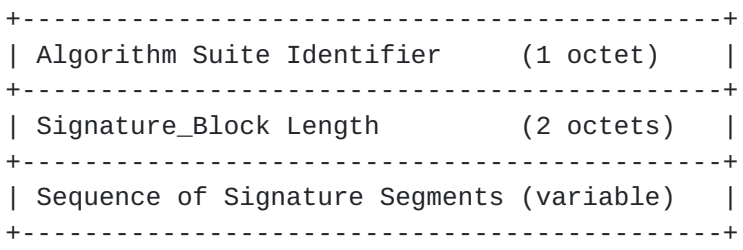
EDITOR'S NOTE: In a previous version of this document there was an Expire Time that was used to provide protection against replay of old (stale) digital signatures or failure to propagate a withdrawal message. This mechanism was removed from the current version of the document. Please see the SIDR mailing list for discussions related to protection against replay attacks. Depending on the result of discussions within the SIDR working group this Additional Info field

could at some future point be used to re-introduce Expire Time, or some other octets used in a future replay protection mechanism. The authors believe that the current instructions whereby the sender uses a null Additional_Info type and the receiver ignores Additional_Info types that it does not understand provides an opportunity to use these octets in the future in a backwards-compatible fashion.

3.3. Signature_Block

Here we provide a detailed description of the Signature_Blocks in the BGPSEC_Path_Signatures attribute.

Signature_Block

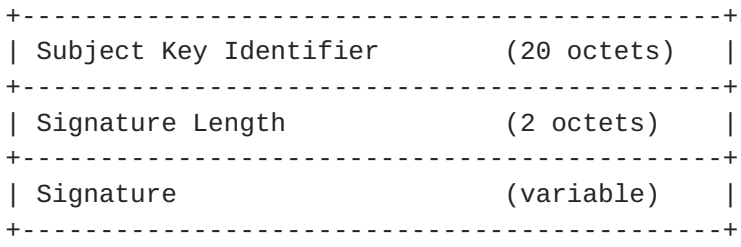


The Algorithm Suite Identifier is a one-octet identifier specifying the digest algorithm and digital signature algorithm used to produce the digital signature in each Signature Segment. An IANA registry of algorithm identifiers for use in BGPSEC is created in the BGPSEC algorithms document[12].

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

A Signature_Block has exactly one Signature Segment for each Secure_Path Segment in the Secure_Path portion of the BGPSEC_Path_Signatures Attribute. (That is, one Signature Segment for each distinct AS on the path for the NLRI in the Update message.)

Signature Segments



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

The Signature Length field contains the size (in octets) of the value in the Signature field of the Signature Segment.

The Signature contains a digital signature that protects the NLRI and the BGPSEC_Path_Signatures attribute (see Sections 4 and 5 for details on generating and verifying this signature, respectively).

4. Generating a BGPSEC Update

Sections 4.1 and 4.2 cover two cases in which a BGPSEC speaker may generate an update message containing the BGPSEC_Path_Signatures attribute. The first case is that in which the BGPSEC speaker originates a new route advertisement (Section 4.1). That is, the BGPSEC speaker is constructing an update message in which the only AS to appear in the BGPSEC_Path_Signatures is the speaker's own AS. The second case is that in which the BGPSEC speaker receives a route advertisement from a peer and then decides to propagate the route advertisement to an external (eBGP) peer (Section 4.2). That is, the BGPSEC speaker has received a BGPSEC update message and is constructing a new update message for the same NLRI in which the BGPSEC_Path_Signatures attribute will contain AS number(s) other than the speaker's own AS.

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

of removing BGPSEC signatures.)

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

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

Note that in order to create or add a new signature to a BGPSEC update message with a given algorithm suite, the BGPSEC speaker must possess a private key suitable for generating signatures for this algorithm suite. Additionally, this private key must correspond to the public key in a valid Resource PKI end-entity certificate whose AS number resource extension includes the BGPSEC speaker's AS number [11]. Note also that new signatures are only added to a BGPSEC update message when a BGPSEC speaker is generating an update message to send to an external peer (i.e., when the AS number of the peer is not equal to the BGPSEC speaker's own AS number). Therefore, a BGPSEC speaker who only sends BGPSEC update messages to peers within its own AS, it does not need to possess any private signature keys.

4.1. Originating a New BGPSEC Update

In an update message that originates a new route advertisement (i.e., an update whose path will contain only a single AS number), when sending the route advertisement to an external, BGPSEC-speaking peer, the BGPSEC speaker creates a new BGPSEC_Path_Signatures attribute as follows.

First, the BGPSEC speaker constructs the Secure_Path with a single Secure_Path Segment. The AS in this path is the BGPSEC speaker's own AS number. In particular, this AS number MUST match the AS number in the AS number resource extension field of the Resource PKI end-entity certificate(s) that will be used to verify the digital signature(s) constructed by this BGPSEC speaker.

Note that the BGPSEC_Path_Signatures attribute and the AS4_Path attribute are mutually exclusive. That is, any update message containing the BGPSEC_Path_Signatures attribute MUST NOT contain the

AS4_Path attribute nor the AS_Path attribute. The information that would be contained in the AS4_Path (or AS_Path) attribute is instead conveyed in the Secure_Path portion of the BGPSEC_Path_Signatures attribute.

Note that the Resource PKI enables the legitimate holder of IP address prefix(es) to issue a signed object, called a Route Origination Authorization (ROA), that authorizes a given AS to originate routes to a given set of prefixes (see [7]). Note that validation of a BGPSEC update message will fail (i.e., the validation algorithm, specified in Section 5.2, returns 'Not Good') unless there exists a valid ROA authorizing the first AS in the Secure_Path portion of the BGPSEC_Path_Signatures attribute to originate routes to the prefix being advertised. Therefore, a BGPSEC speaker SHOULD NOT originate a BGPSEC update advertising a route for a given prefix unless there exists a valid ROA authorizing the BGPSEC speaker's AS to originate routes to this prefix.

The pCount field of the Secure_Path Segment is typically set to the value 1. However, a BGPSEC speaker may set the pCount field to a value greater than 1. Setting the pCount field to a value greater than one has the same semantics as repeating an AS number multiple times in the AS_PATH of a non-BGPSEC update message (e.g., for traffic engineering purposes). Setting the pCount field to a value greater than one permits this repetition without requiring a separate digital signature for each repetition.

If the BGPSEC speaker is not a member of an autonomous system confederation [3], then the Flags field of the Secure_Path Segment MUST be set to zero. (Members of a confederation should follow the special processing instructions for confederation members in Section 4.4.)

The BGPSEC speaker next constructs the Additional_Info portion of the BGPSEC_Path_Signatures attribute. The Info Type MUST be set to zero and the Info Length MUST also be set to zero. The Info Value field is empty (has length zero). It is anticipated that future specifications may specify values of Info Type other than zero. Therefore, BGPSEC receivers compliant with this specification must be able to accept Additional_Info fields with non-zero Info Type. Such receivers will use the Additional_Field to verify digital signatures (see Section 5) but will otherwise ignore Additional_Field non-zero Info Fields.

Typically, a BGPSEC speaker will use only a single algorithm suite, and thus create only a single Signature_Block in the BGPSEC_Path_Signatures attribute. However, to ensure backwards compatibility during a period of transition from a 'current'

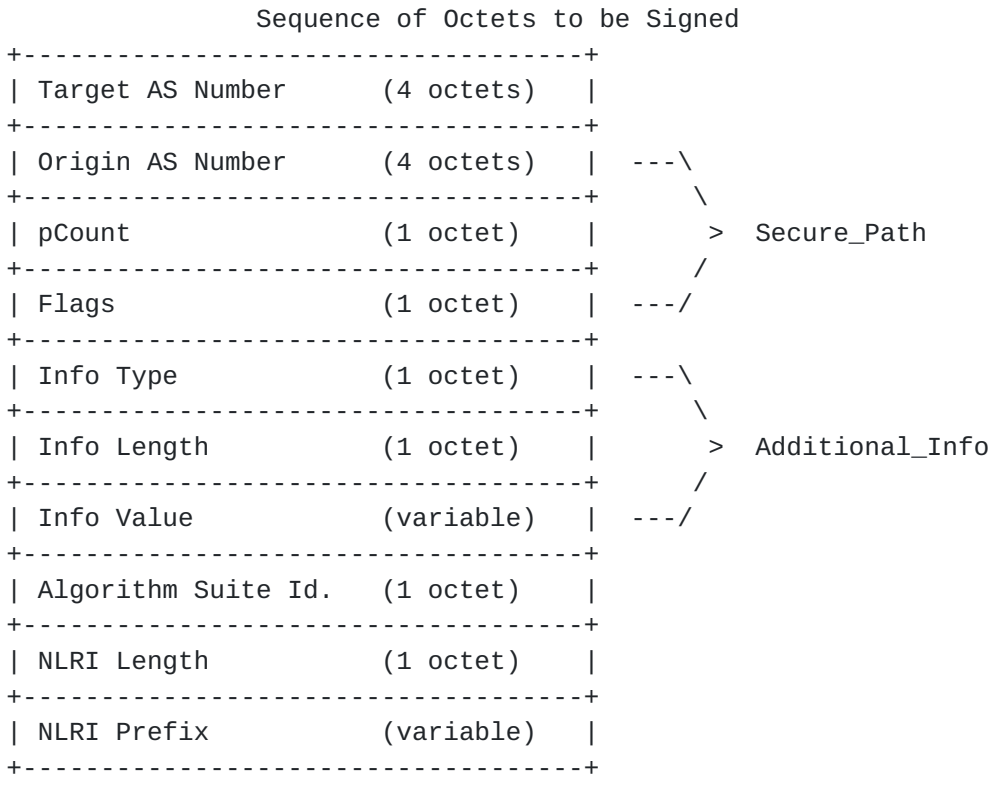
algorithm suite to a 'new' algorithm suite, it will be necessary to originate update messages that contain a Signature_Block for both the 'current' and the 'new' algorithm suites (see Section 6.1).

When originating a new route advertisement, each Signature_Block MUST consist of a single Signature Segment. The following describes how the BGPSEC speaker populates the fields of the Signature_Block.

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

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

- Construct a sequence of octets by concatenating the Target AS Number, the Secure_Path (Origin AS, pCount, and Flags), the Additional_Info (Info Type, Info Length, and Info Value), Algorithm Suite Identifier, and NLRI. The Target AS Number is the AS to whom the BGPSEC speaker intends to send the update message. (Note that the Target AS number is the AS number announced by the peer in the OPEN message of the BGP session within which the update is sent.)



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

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

The Signature Length field is populated with the length (in octets) of the Signature field.

4.2. Propagating a Route Advertisement

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

If a BGPSEC router has received only non-BGPSEC update messages (without the BGPSEC_Path_Signatures attribute), containing the AS_Path attribute, from a peer for a given prefix and if it chooses to propagate that peer's route for the prefix, then it MUST NOT attach any BGPSEC_Path_Signatures attribute to the corresponding update being propagated. (Note that a BGPSEC router may also receive

a non-BGPSEC update message from an internal peer without the AS_Path attribute, i.e., with just the NLRI in it. In that case, the prefix is originating from that AS and hence the BGPSEC speaker SHOULD sign and forward the update to its external peers, as specified in Section 4.1.)

Conversely, if a BGPSEC router has received a BGPSEC update message (with the BGPSEC_Path_Signatures attribute) from a peer for a given prefix and it chooses to propagate that peer's route for the prefix, then it SHOULD propagate the route as a BGPSEC update message containing the BGPSEC_Path_Signatures attribute. However, the BGPSEC speaker MAY propagate the route as a (unsigned) BGP update message without the BGPSEC_Path_Signatures attribute.

Note that removing BGPSEC signatures (i.e., propagating a route advertisement without the BGPSEC_Path_Signatures attribute) has significant security ramifications. (See Section 7 for discussion of the security ramifications of removing BGPSEC signatures.) Therefore, when a route advertisement is received via a BGPSEC update message, propagating the route advertisement without the BGPSEC_Path_Signatures attribute is NOT RECOMMENDED. Furthermore, note that when a BGPSEC speaker propagates a route advertisement with the BGPSEC_Path_Signatures attribute it is not attesting to the validation state of the update message it received. (See Section 7 for more discussion of the security semantics of BGPSEC signatures.)

If the BGPSEC speaker is producing an update message which would, in the absence of BGPSEC, contain an AS_SET (e.g., the BGPSEC speaker is performing proxy aggregation), then the BGPSEC speaker MUST NOT include the BGPSEC_Path_Signatures attribute. In such a case, the BGPSEC speaker must remove any existing BGPSEC_Path_Signatures in the received advertisement(s) for this prefix and produce a standard (non-BGPSEC) update message. It should be noted that BCP 172 [5] recommends against the use of AS_SET and AS_CONFED_SET in AS_PATH in BGP updates.

To generate the BGPSEC_Path_Signatures attribute on the outgoing update message, the BGPSEC speaker first prepends a new Secure_Path Segment (places in first position) to the Secure_Path. The AS number in this Secure_Path segment MUST match the AS number in the AS number resource extension field of the Resource PKI end-entity certificate(s) that will be used to verify the digital signature(s) constructed by this BGPSEC speaker.

The pCount is typically set to the value 1. A BGPSEC speaker may set the pCount field to a value greater than 1. (See Section 4.1 for a discussion of setting pCount to a value greater than 1.) A route server that participates in the BGP control path, but does not act as

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

If the BGPSEC speaker is not a member of an autonomous system confederation [3], then the Flags field of the Secure_Path Segment MUST be set to zero. (Members of a confederation should follow the special processing instructions for confederation members in Section 4.4.)

The BGPSEC speaker next copies the Additional_Info portion of the BGPSEC_Path_Signatures directly from the received update message to the new update message (that it is constructing). Note that the BGPSEC speaker MUST NOT change the Additional_Info as any change to Additional_Info will cause the new BGPSEC update message to fail validation (see Section 5).

If the received BGPSEC update message contains two Signature_ Blocks and the BGPSEC speaker supports both of the corresponding algorithms suites, then the new update message generated by the BGPSEC speaker SHOULD include both of the Signature_Blocks. If the received BGPSEC update message contains two Signature_Blocks and the BGPSEC speaker only supports one of the two corresponding algorithm suites, then the BGPSEC speaker MUST remove the Signature_Block corresponding to the algorithm suite that it does not understand. If the BGPSEC speaker does not support the algorithm suites in any of the Signature_Blocks contained in the received update message, then the BGPSEC speaker MUST NOT propagate the route advertisement with the BGPSEC_Path_Signatures attribute (i.e., propagate it as an unsigned BGP update message).

Note that in the case where there are two Signature_Blocks (corresponding to different algorithm suites) that the validation algorithm (see Section 5.2) deems a BGPSEC update message to be 'Good' if there is at least one supported algorithm suite (and

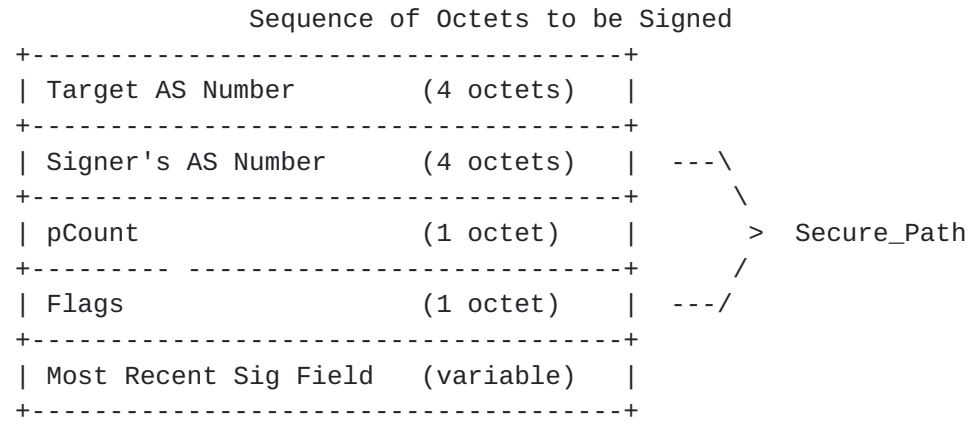
corresponding Signature_Block) that is deemed 'Good'. This means that a 'Good' BGPSEC update message may contain a Signature_Block which is not deemed 'Good' (e.g., contains signatures that the BGPSEC does not successfully verify). Nonetheless, such Signature_Blocks MUST NOT be removed. (See Section 7 for a discussion of the security ramifications of this design choice.)

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

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

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

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



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

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

The Signature Length field is populated with the length (in octets) of the Signature field.

4.3. Processing Instructions for Confederation Members

Members of autonomous system confederations [3] must additionally follow the instructions in this section for processing BGPSEC update messages.

When a confederation member sends a BGPSEC update message to a peer that is a member of the same confederation, the confederation member puts its (private) Member-AS Number (as opposed to the public AS Confederation Identifier) in the AS Number field of the Secure_Path Segment that it adds to the BGPSEC update message. Furthermore, when a confederation member sends a BGPSEC update message to a peer that is a member of the same confederation, the BGPSEC speaker that generates the Secure_Path Segment sets the Confed_Segment flag to one. Note that this means that in a BGPSEC update message, an AS number appears in a Secure_Path Segment with the Confed_Segment flag set to one, in precisely those circumstances where the AS number would appear in a segment of type AS_CONFED_SEQUENCE in a non-BGPSEC update message.

Within a confederation, the verification of BGPSEC signatures added by other members of the confederation is optional. If a confederation chooses to have its members not verify signatures added by other confederation members, then when sending a BGPSEC update

message to a peer that is a member of the same confederation, the confederation MAY set the Signature field within the Signature_Segment that it generates to be zero (in lieu of calculating the correct digital signature as described in Sections 4.1 and 4.2). Note that if a confederation chooses not to verify digital signatures within the confederation, then BGPSEC is able to provide no assurances about the integrity of the (private) Member-AS Numbers placed in Secure_Path segments where the Confed_Segment flag is set to one.

When a confederation member receives a BGPSEC update message from a peer within the confederation and propagates it to a peer outside the confederation, it must remove all of the Secure_Path Segments added by confederation members as well as the corresponding Signature Segments. To do this, the confederation member propagating the route outside the confederation does the following:

- First, starting with the least recently added Secure_Path segments, remove all of the consecutive Secure_Path segments that have the Confed_Segment flag set to one. Stop this process once a Scure_Path segment is reached which has its Confed_Segment flag set to zero. Keep a count of the number of segments removed in this fashion.

- Second, starting with the most recently added Signature Segment, remove a number of Signature Segments equal to the number of Secure_Path Segments removed in the previous step. (That is, remove the K most recently added signature segments, where K is the number of Secure_Path Segments removed in the previous step.)

- Finally, add a Secure_Path Segment containing, in the AS field, the AS Confederation Identifier (the public AS number of the confederation) as well as a corresponding Signature Segment. Note that all fields other that the AS field are populated as per Sections 4.1 and 4.2.

When validating a received BGPSEC update message, confederation members must make the following adjustment to the algorithm presented in Section 5.2. When a confederation member processes (validates) a Signature Segment and its corresponding Secure_Path Segment, the confederation member must note that for a signature produced by a BGPSEC speaker outside of a confederation, the Target AS will always be the AS Confederation Identifier (the public AS number of the confederation) as opposed to the Member-AS Number.

To handle this case, when a BGPSEC speaker (that is a confederation member) processes a current Secure_Path Segment that has the Confed_Segment flag set to zero, if the next most recently added

Secure_Path segment has the Confed_Segment flag set to one then, when computing the digest for the current Secure_Path segment, the BGPSEC speaker takes the Target AS Number to be the AS Confederation Identifier of the validating BGPSEC speaker's own confederation. (Note that the algorithm in Section 5.2 processes Secure_Path Segments in order from most recently added to least recently added, therefore this special case will apply to the first Secure_Path segment that the algorithm encounters that has the Confed_Segment flag set to one.)

Finally, as discussed above, an AS confederation may optionally decide that its members will not verify digital signatures added by members. In such a federation, when a confederation member runs the algorithm in Section 5.2, when processing a Signature_Segment, the confederation member first checks whether the Confed_Sequence flag in the corresponding Secure_Path segment is set to one. If the Confed_Sequence flag is set to one in the corresponding Secure_Path segment, the confederation member does not perform any further checks on the Signature_Segment and immediately moves on to the next Signature_Segment (and checks its corresponding Secure_Path segment). Note that as specified in Section 5.2, it is an error for a BGPSEC speaker to receive a BGPSEC update messages containing a Secure_Path segment with the Confed_Sequence flag set to one from a peer who is not a member of the same AS confederation. (Such an error is treated in exactly the same way as receipt of a non-BGPSEC update message containing an AS_CONFED_SEQUENCE from a peer that is not a member of the same AS confederation.)

4.4. Reconstructing the AS_PATH Attribute

BGPSEC update messages do not contain the AS_PATH attribute. Note, however, that the AS_PATH attribute can be reconstructed from the BGPSEC_Path_Signatures attribute. This is necessary in the case where a route advertisement is received via a BGPSEC update message and then propagated to a peer via a non-BGPSEC update message. There may be additional cases where an implementation finds it useful to perform this reconstruction.

The AS_PATH attribute can be constructed from the BGPSEC_Path_Signatures attribute as follows. Starting with an empty AS_PATH attribute, process the Secure_Path segments in order from least-recently added (corresponding to the origin) to most-recently added. For each Secure_Path segment perform the following steps:

1. If the Confed_Segment flag in the Secure_Path segment is set to one, then look at the most-recently added segment in the AS_PATH.

* In the case where the AS_PATH is empty or in the case where the most-recently added segment is of type AS_SEQUENCE then add (prepend to the AS_PATH) a new AS_PATH segment of type AS_CONFED_SEQUENCE. This segment of type AS_CONFED_SEQUENCE shall contain a number of elements equal to the pCount field in the current Secure_Path segment. Each of these elements shall be the AS number contained in the current Secure_Path segment. (That is, if the pCount field is X, then the segment of type AS_CONFED_SEQUENCE contains X copies of the Secure_Path segment's AS Number field.)

* In the case where the most recently added segment in the AS_PATH is of type AS_CONFED_SEQUENCE then add (prepend to the segment) a number of elements equal to the pCount field in the current Secure_Path segment. The value of each of these elements shall be the AS number contained in the current Secure_Path segment. (That is, if the pCount field is X, then add X copies of the Secure_Path segment's AS Number field to the existing AS_CONFED_SEQUENCE.)

2. If the Confed_Segment flag in the Secure_Path segment is set to zero, then look at the most-recently added segment in the AS_PATH.

   * In the case where the AS_PATH is empty then add (prepend to the AS_PATH) a new AS_PATH segment of type AS_SEQUENCE. This segment of type AS_SEQUENCE shall contain a number of elements equal to the pCount field in the current Secure_Path segment. Each of these elements shall be the AS number contained in the current Secure_Path segment. (That is, if the pCount field is X, then the segment of type AS_SEQUENCE contains X copies of the Secure_Path segment's AS Number field.)

   * In the case where the most recently added segment in the AS_PATH is of type AS_SEQUENCE then add (prepend to the segment) a number of elements equal to the pCount field in the current Secure_Path segment. The value of each of these elements shall be the AS number contained in the current Secure_Path segment. (That is, if the pCount field is X, then add X copies of the Secure_Path segment's AS Number field to the existing AS_SEQUENCE.)

5. Processing a Received BGPSEC Update

Upon receiving a BGPSEC update message from an external (eBGP) peer, a BGPSEC speaker SHOULD validate the message to determine the authenticity of the AS PATH information contained in the

BGPSEC_Path_Signatures attribute. Section 5.1 provides an overview of BGPSEC validation and Section 5.2 provides a specific algorithm for performing such validation. (Note that an implementation need not follow the specific algorithm in Section 5.2 as long as the input output behavior of the validation is identical to that of the algorithm in Section 5.2.) During exceptional conditions (e.g., the BGPSEC speaker receives an incredibly large number of update messages at once) a BGPSEC speaker MAY defer validation of incoming BGPSEC update messages. The treatment of such BGPSEC update messages, whose validation has been deferred, is a matter of local policy. Implementations that support such deferment of validation MUST perform validation of these messages as soon as possible (i.e., as soon as resources are available to perform validation) and MUST re-run best path selection once the validation status of such update messages is known.

BGPSEC update messages do not contain an AS_PATH attribute. Therefore, a BGPSEC speaker MUST utilize the AS path information in the BGPSEC_Path_Signatures attribute in all cases where it would otherwise use the AS path information in the AS_PATH attribute. The only exception to this rule is when AS path information must be updated in order to propagate a route to a peer (in which case the BGPSEC speaker follows the instructions in Section 4). Section 4.4 provides an algorithm for constructing an AS_PATH attribute from a BGPSEC_Path_Signatures attribute. Whenever the use of AS path information is called for (e.g., loop detection, or use of AS path length in best path selection) the externally visible behavior of the implementation shall be the same as if the implementation had run the algorithm in Section 4.4 and used the resulting AS_PATH attribute as it would for a non-BGPSEC update message. However, in practice, it is expected that most implementations will not actually run the algorithm from Section 4.4, and will instead transform the BGPSEC_Path_Signatures attribute directly into some internal representation of AS path.

Many signature algorithms are non-deterministic. That is, many signature algorithms will produce different signatures each time they are run (even when they are signing the same data with the same key). Therefore, if an implementation receives a BGPSEC update from a peer and later receives a second BGPSEC update message from the same peer, the implementation SHOULD treat the second message as a duplicate update message if it differs from the first update message only in the Signature fields (within the BGPSEC_Path_Signatures attribute). That is, if all the fields in the second update are identical to the fields in the first update message, except for the Signature fields, then the second update message should be treated as a duplicate of the first update message. Note that if other fields (e.g., the Subject Key Identifier field) within a Signature segment differ

between two update messages then the two updates are not duplicates.

With regards to the processing of duplicate update messages, if the first update message is valid, then an implementation SHOULD NOT run the validation procedure on the second, duplicate update message (even if the bits of the signature field are different). If the first update message is not valid, then an implementation SHOULD run the validation procedure on the second duplicate update message (as the signatures in the second update may be valid even though the first contained a signature that was invalid).

5.1. Overview of BGPSEC Validation

Validation of a BGPSEC update messages makes use of data from RPKI certificates and signed Route Origination Authorizations (ROA). In particular, to validate update messages containing the BGPSEC_Path_Signatures attribute, it is necessary that the recipient have access to the following data obtained from valid RPKI certificates and ROAs:

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

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

Note that the BGPSEC speaker could perform the validation of RPKI certificates and ROAs on its own and extract the required data, or it could receive the same data from a trusted cache that performs RPKI validation on behalf of (some set of) BGPSEC speakers. (The latter case in analogous to the use of the RPKI-RTR protocol [13] for origin validation.)

To validate a BGPSEC update message containing the BGPSEC_Path_Signatures attribute, the recipient performs the validation steps specified in Section 5.2. The validation procedure results in one of two states: 'Good' and 'Not Good'.

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

changes to the BGP decision process and leaves to the operator the selection of an appropriate policy mechanism to achieve the operator's desired results within the BGP decision process.

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

5.2. Validation Algorithm

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

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

1. Check to ensure that the entire BGPSEC_Path_Signatures attribute is syntactically correct (conforms to the specification in this document).

2. Check that each Signature_Block contains one Signature segment for each Secure_Path segment in the Secure_Path portion of the BGPSEC_Path_Signatures attribute. (Note that the entirety of each Signature_Block must be checked to ensure that it is well formed, even though the validation process may terminate before all signatures are cryptographically verified.)

3. Check that the update message does not contain both a BGPSEC_Path_Signatures attribute and an AS_PATH attribute.

4. If the update message was received from a peer that is not a member of the BGPSEC speaker's AS confederation, check to ensure that none of the Secure_Path segments contain a Flags field with the Confed_Sequence flag set to one.

5. If the update message was received from a peer that is not expected to set pCount equal to zero (see Section 4.2) then check to ensure that the pCount field in the most-recently added Secure_Path segment is not equal to zero.

If there are two Signature_Blocks within the BGPSEC_Path_Signatures attribute and one of them is poorly formed (or contains the wrong number of Signature segments) , then the recipient should log that an error occurred, strip off that particular Signature_Block and process the update message as though it arrived with a single Signature_Block. If the BGPSEC_Path_Signatures attribute contains an error that is not local to one of two Signature_Blocks, then the recipient should log that an error occurred and drop the update message containing the error. (In particular, if any of checks 3-5 above fail, the recipient should log that an error occurred and drop the update message containing the error.)

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

Finally, the BGPSEC speaker examines the Signature_Blocks in the BGPSEC_Path_Signatures attribute. A Signature_Block corresponding to an algorithm suite that the BGPSEC speaker does not support is not considered in validation. If there does not exist a Signature_Block corresponding to an algorithm suite that the BGPSEC speaker supports, then the BGPSEC speaker MUST treat the update message in the same manner that the BGPSEC speaker would treat an (unsigned) update message that arrived without a BGPSEC_Path_Signatures attribute.

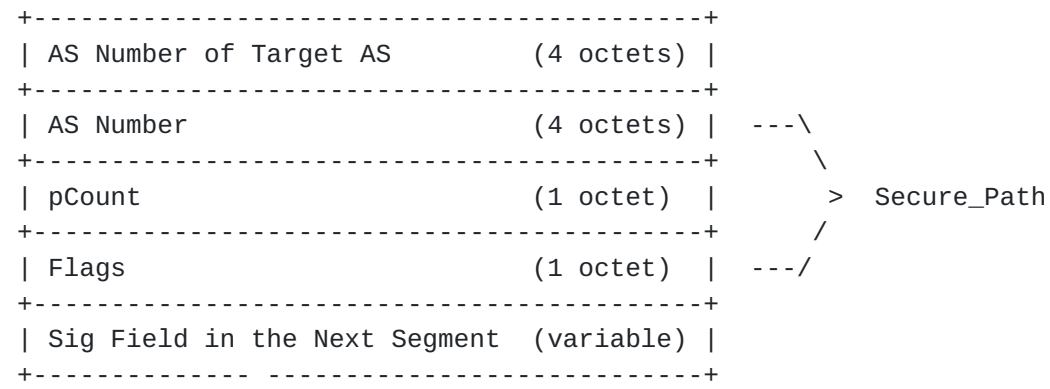
For each remaining Signature_Block (corresponding to an algorithm suite supported by the BGPSEC speaker), the BGPSEC speaker iterates through the Signature segments in the Signature_Block, starting with the most recently added segment (and concluding with the least recently added segment). Note that there is a one-to-one correspondence between Signature segments and Secure_Path segments within the BGPSEC_Path_Signatures attribute. The following steps make use of this correspondence.

- (Step I): Locate the public key needed to verify the signature (in the current Signature segment). To do this, consult the valid RPKI end-entity certificate data and look up all valid (AS, SKI,

Public Key) triples in which the AS matches the AS number in the corresponding Secure_Path segment. Of these triples that match the AS number, check whether there is an SKI that matches the value in the Subject Key Identifier field of the Signature segment. If this check finds no such matching SKI value, then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature-List Block.

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

Sequence of Octets to be Hashed

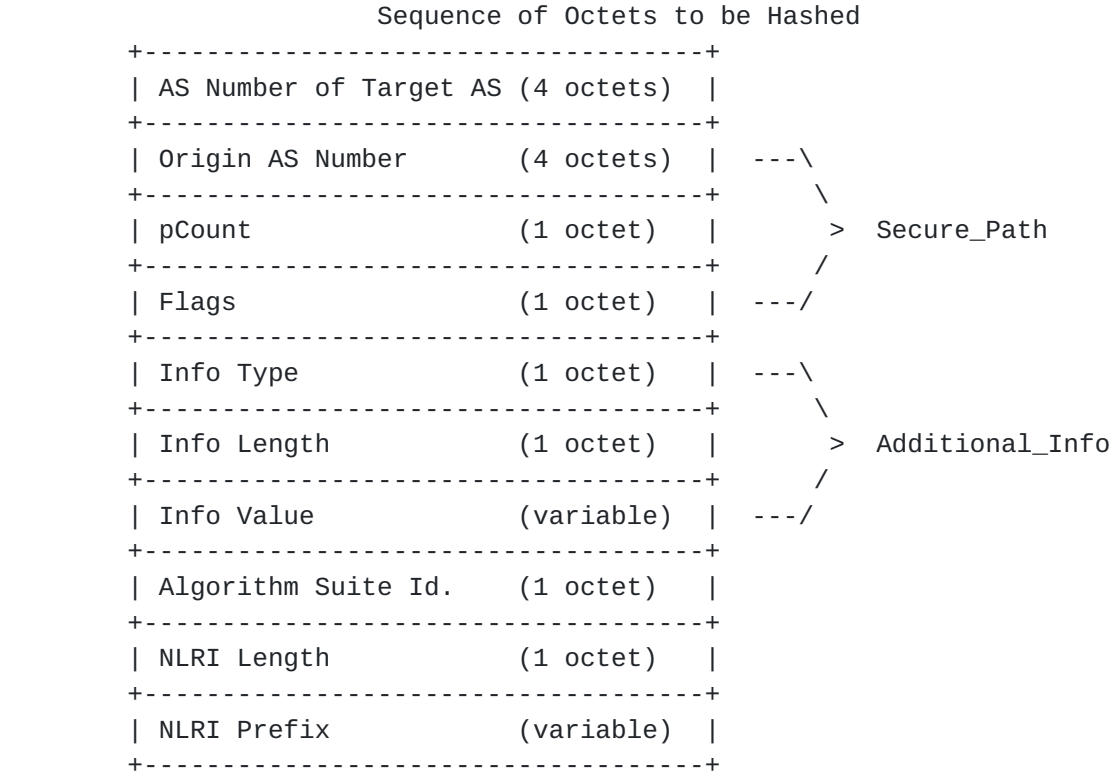


For the first segment to be processed (the most recently added segment), the 'AS Number of Target AS' is the AS number of the BGPSEC speaker validating the update message. Note that if a BGPSEC speaker uses multiple AS Numbers (e.g., the BGPSEC speaker is a member of a confederation), the AS number used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

For each other Signature Segment, the 'AS Number of Target AS' is the AS number in the Secure_Path segment that corresponds to the Signature Segment added immediately after the one being processed. (That is, in the Secure_Path segment that corresponds to the Signature segment that the validator just finished processing.)

The AS Number, pCount and Flags fields are taken from the Secure_Path segment that corresponds to the Signature segment currently being processed. The 'Signature Field in the Next Segment' is the Signature field found in the Signature segment that is next to be processed (that is, the next most recently added Signature Segment).

Alternatively, if the segment being processed corresponds to the origin AS (i.e., if it is the least recently added segment), then the digest function should be computed on the following sequence of octets:



The NLRI Length, NLRI Prefix, Additional_Info, and Algorithm Suite Identifier are all obtained in a straight forward manner from the NLRI of the update message or the BGPSEC_Path_Signatures attribute being validated. The Origin AS Number, pCount, and Flags fields are taken from the Secure_Path segment corresponding to the Signature Segment currently being processed.

The 'AS Number of Target AS' is the AS Number from the Secure_Path segment that was added immediately after the Secure_Path segment containing the Origin AS Number. (That is, the Secure_Path segment corresponding to the Signature segment that the receiver just finished processing prior to the current Signature segment.)

o (Step III): Use the signature validation algorithm (for the given algorithm suite) to verify the signature in the current segment. That is, invoke the signature validation algorithm on the following three inputs: the value of the Signature field in the current segment; the digest value computed in Step II above; and the public key obtained from the valid RPKI data in Step I above. If the signature validation algorithm determines that the

signature is invalid, then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature_Block. If the signature validation algorithm determines that the signature is valid, then continue processing Signature-Segments (within the current Signature-List Block).

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

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

6. Algorithms and Extensibility

6.1. Algorithm Suite Considerations

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

To this end, a mandatory algorithm suites document will be created which specifies a mandatory-to-use 'current' algorithm suite for use by all BGPSEC speakers [12]. Additionally, the document specifies an additional 'new' algorithm suite that is recommended to implement.

It is anticipated that in the future the mandatory algorithm suites document will be updated to specify a transition from the 'current' algorithm suite to the 'new' algorithm suite. During the period of transition (likely a small number of years), all BGPSEC update messages SHOULD simultaneously use both the 'current' algorithm suite and the 'new' algorithm suite. (Note that Sections 3 and 4 specify how the BGPSEC_Path_Signatures attribute can contain signatures, in parallel, for two algorithm suites.) Once the transition is complete, use of the old 'current' algorithm will be deprecated, use of the 'new' algorithm will be mandatory, and a subsequent 'even

newer' algorithm suite may be specified as recommend to implement. Once the transition has successfully been completed in this manner, BGPSEC speakers SHOULD include only a single Signature_Block (corresponding to the 'new' algorithm).

6.2. Extensibility Considerations

This section discusses potential changes to BGPSEC that would require substantial changes to the processing of the BGPSEC_Path_Signatures and thus necessitate a new version of BGPSEC. Examples of such changes include:

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

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

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

7. Security Considerations

For discussion of the BGPSEC threat model and related security considerations, please see [10].

A BGPSEC speaker who receives a valid BGPSEC update message, containing a route advertisement for a given prefix, is provided with

the following security guarantees:

- The origin AS number corresponds to an autonomous system that has been authorized by the IP address space holder to originate route advertisements for the given prefix.

- For each AS number in the AS Path, a BGPSEC speaker authorized by the holder of the AS number intentionally chose (in accordance with local policy) to propagate the route advertisement to the next AS in the Secure_Path.

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

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

Note that there may be cases where a BGPSEC speaker deems 'Good' (as per the validation algorithm in Section 5.2) a BGPSEC update message that contains both a 'Good' and a 'Not Good' Signature_Block. That is, the update message contains two sets of signatures corresponding to two algorithm suites, and one set of signatures verifies correctly and the other set of signatures fails to verify. In this case, the protocol specifies that if the BGPSEC speaker propagates the route advertisement received in such an update message then the BGPSEC speaker SHOULD add its signature to each of the Signature_Blocks using both the corresponding algorithm suite. Thus the BGPSEC speaker creates a signature using both algorithm suites and creates a new update message that contains both the 'Good' and the 'Not Good' set of signatures (from its own vantage point).

To understand the reason for such a design decision consider the case where the BGPSEC speaker receives an update message with both a set of algorithm A signatures which are 'Good' and a set of algorithm B signatures which are 'Not Good'. In such a case it is possible (perhaps even quite likely) that some of the BGPSEC speaker's peers (or other entities further 'downstream' in the BGP topology) do not support algorithm A. Therefore, if the BGPSEC speaker were to remove the 'Not Good' set of signatures corresponding to algorithm B, such entities would treat the message as though it were unsigned. By including the 'Not Good' set of signatures when propagating a route advertisement, the BGPSEC speaker ensures that 'downstream' entities have as much information as possible to make an informed opinion about the validation status of a BGPSEC update.

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

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

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

- The BGPSEC speaker received the given route advertisement with the indicated NLRI and Secure_Path; and

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

The BGPSEC update validation procedure is a potential target for denial of service attacks against a BGPSEC speaker. To mitigate the effectiveness of such denial of service attacks, BGPSEC speakers should implement an update validation algorithm that performs expensive checks (e.g., signature verification) after performing less expensive checks (e.g., syntax checks). The validation algorithm specified in Section 5.2 was chosen so as to perform checks which are likely to be expensive after checks that are likely to be inexpensive. However, the relative cost of performing required validation steps may vary between implementations, and thus the algorithm specified in Section 5.2 may not provide the best denial of service protection for all implementations.

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

Finally, BGPSEC does not provide protection against all attacks at the transport layer. An adversary on the path between a BGPSEC speaker and its peer is able to perform attacks such as modifying valid BGPSEC updates to cause them to fail validation, injecting (unsigned) BGP update messages without BGPSEC_Path_Signature attributes, or injecting BGPSEC update messages with BGPSEC_Path_Signature attributes that fail validation, or causing the peer to tear-down the BGP session. Therefore, BGPSEC implementations MUST support appropriate transport security mechanisms.

EDITOR'S NOTE: Do we want to mandate a specific transport security mechanism (e.g., TCP-AO)?

8. Contributors

8.1. Authors

Rob Austein
Dragon Research Labs
sra@hactrn.net

Steven Bellovin
Columbia University
smb@cs.columbia.edu

Randy Bush
Internet Initiative Japan
randy@psg.com

Russ Housley
Vigil Security
housley@vigilsec.com

Matt Lepinski
BBN Technologies
lepinski@bbn.com

Stephen Kent
BBN Technologies
kent@bbn.com

Warren Kumari
Google
warren@kumari.net

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

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

Samuel Weiler
Cobham
weiler+ietf@watson.org

8.2. Acknowledgements

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

9. References

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

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

[3] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, August 2007.

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

[5] Kumari, W. and K. Sriram, "Recommendation for Not Using AS_SET and AS_CONFED_SET in BGP", RFC 6472, December 2011.

[6] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[7] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, February 2012.

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

[9] Patel, K., Ward, D., and R. Bush, "Extended Message support for BGP", draft-ietf-idr-bgp-extended-messages, July 2012.

[10] Kent, S., and A. Chi, "Threat Model for BGP Path Security", draft-ietf-sidr-bgpsec-threats-02, February 2012.

[11] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests", draft-ietf-sidr-bgpsec-pki-profiles-03, April 2012.

[12] Turner, S., "BGP Algorithms, Key Formats, & Signature Formats", draft-ietf-sidr-bgpsec-algs-02, March 2012.

[13] Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rtr-26, February 2012.

Author's Address

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

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