Abstract

This document specifies the algorithms, algorithm parameters, asymmetric key formats, asymmetric key sizes, and signature formats used in BGPsec (Border Gateway Protocol Security). This document obsoletes RFC 8208 ("BGPsec Algorithms, Key Formats, and Signature Formats") by adding Documentation Algorithm IDs, Experimentation Algorithm IDs, correcting the range of unassigned algorithms IDs to fill the complete range, and restructured the document for better reading.

This document also includes example BGPsec UPDATE messages as well as the private keys used to generate the messages and the certificates necessary to validate those signatures.

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 August 2, 2018
Internet Draft    BGPsec Algs, Key & Signature Formats    April 15, 2019

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction .................................................. 3
   1.1. Terminology .............................................. 3
   1.2. Changes from RFC 8208 ................................. 4
2. Algorithms .................................................... 4
   2.1. Algorithm ID Types .................................... 4
   2.2. Signature Algorithms ................................. 6
      2.2.1. Algorithm ID 0x01 (1) - (ECDSA-P256) .......... 6
3. Asymmetric Key Pair Formats ................................ 6
   3.1. Asymmetric Key Pair for Algorithm ID 0x01 (1) -
       (ECDSA-P256) ........................................... 6
      3.1.1. Public Key Format ............................... 6
      3.1.2. Private Key Format ............................. 7
4. Signature Formats ............................................ 7
5. Additional Requirements ..................................... 7
6. Security Considerations .................................... 7
7. IANA Considerations ....................................... 8
8. References ................................................ 10
   8.1. Normative References ................................. 10
   8.2. Informative References ............................. 11
Appendix A. Examples ........................................ 13
   A.1. Topology and Experiment Description ................. 13
   A.2. Keys ................................................ 13
   A.3. BGPsec IPv4 .......................................... 17
   A.4. BGPsec IPv6 .......................................... 20
Acknowledgements ............................................. 23
Authors’ Addresses ......................................... 23
1. Introduction

This document specifies the following:

- the digital signature algorithm and parameters,
- the hash algorithm and parameters,
- the algorithm identifier assignment and classification,
- the public and private key formats, and
- the signature formats

used by Resource Public Key Infrastructure (RPKI) Certification Authorities (CAs) and BGPsec (Border Gateway Protocol Security) speakers (i.e., routers). CAs use these algorithms when processing requests for BGPsec Router Certificates [RFC8209]. Examples of when BGPsec routers use these algorithms include requesting BGPsec certificates [RFC8209], signing BGPsec UPDATE messages [RFC8205], and verifying signatures on BGPsec UPDATE messages [RFC8205].

This document updates [RFC7935] to add support for a) a different algorithm for BGPsec certificate requests, which are issued only by BGPsec speakers; b) a different Subject Public Key Info format for BGPsec certificates, which is needed for the specified BGPsec signature algorithm; and c) different signature formats for BGPsec signatures, which are needed for the specified BGPsec signature algorithm. The BGPsec certificates are differentiated from other RPKI certificates by the use of the BGPsec Extended Key Usage as defined in [RFC8209]. BGPsec uses a different algorithm [RFC6090] [DSS] as compared to the rest of the RPKI that provides similar security with smaller keys making the certificates smaller; these algorithms also result in smaller signatures, which makes the PDUs smaller.

Appendix A (non-normative) contains example BGPsec UPDATE messages as well as the private keys used to generate the messages and the certificates necessary to validate the signatures.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.
1.2. Changes from RFC 8208

This section describes the significant changes between [RFC8208] and this document.

- Added Section 2.1 of algorithm ID types. Also, the interpretation of these IDs is described.
- Restructured Sections 2 and 3 to align with the corresponding algorithm suite identifier value.
- Correction of range for unassigned algorithm suite identifier values.
- Adding of Documentation algorithm suite identifier values.
- Adding of Experimentation algorithm suite identifier values.
- Changed Next-HOP IP in Appendix A’s IPv6 Example to use private usage IPv6 address.

2. Algorithms

The algorithms used to compute signatures on CA certificates, BGPsec Router Certificates, and Certificate Revocation Lists (CRLs) are as specified in Section 2 of [RFC7935]. This section addresses BGPsec algorithms used by BGPsec [RFC8205] [DSS]. For example, these algorithms are used by BGPsec routers to sign and verify BGPsec UPDATE messages. To identify which algorithm is used, the BGPsec UPDATE message contains the corresponding algorithm ID in each Signature_Block of the BGPsec UPDATE message.

2.1. Algorithm ID Types

Algorithms in BGPsec UPDATE messages are identified by the Algorithm Suite Identifier field (Algorithm ID) within the Signature_Block (see Section 3.2 of [RFC8205]).

This document specifies five types of algorithm IDs:

- Reserved Algorithm ID

Reserved algorithm IDs are the values 0x00 (0) and 0xFF (255). These IDs MUST NOT be used in a Signature_Block and if encountered, the router MUST treat BGPsec UPDATE messages as Malformed [RFC4271].
o Signature Algorithm ID

Signature algorithms are defined in Section 2.2 of this document. Processing of BGPsec UPDATE signing and validation using signature algorithms is described in length in Section 4.2 and Section 5.2 of [RFC8205].

o Unassigned Algorithm ID

This type of algorithm ID is free for future assignments and MUST NOT be used until an algorithm is officially assigned (see Section 7). In case a router encounters an unassigned algorithm ID in one of the Signature_Blocks of a BGPsec UPDATE message, the router SHOULD process the Signature_Block as "unsupported algorithm" as specified in Section 5.2 of [RFC8205].

o Experimentation Algorithm ID

Experimentation algorithm IDs span from 0xF7 (247) to 0xFA (250). To allow experimentation to accurately describe deployment examples, the use of publicly assigned algorithm IDs is inappropriate, and a reserved block of Experimentation algorithm IDs is required. This ensures that experimentation does not clash with assigned algorithm IDs in deployed networks, and mitigates the risks to operational integrity of the network through inappropriate use of experimentation to perform literal configuration of routing elements on production systems. A router that encounters an algorithm ID of this type outside of an experimental network, SHOULD treat it the same as "unsupported algorithm" as specified in Section 5.2 of [RFC8205].

o Documentation Algorithm ID

Documentation algorithm IDs span from 0xFB (251) to 0xFE (254). To allow documentation to accurately describe deployment examples, the use of publicly assigned algorithm IDs is inappropriate, and a reserved block of Documentation algorithm IDs is required. This ensures that documentation does not clash with assigned algorithm IDs in deployed networks, and mitigates the risks to operational integrity of the network through inappropriate use of documentation to perform literal configuration of routing elements on production systems. A router that encounters an algorithm ID of this type SHOULD treat it the same as "unsupported algorithm" as specified in Section 5.2 of [RFC8205].
2.2. Signature Algorithms

2.2.1. Algorithm ID 0x01 (1) - (ECDSA-P256)

- The signature algorithm used MUST be the Elliptic Curve Digital Signature Algorithm (ECDSA) with curve P-256 [RFC6090] [DSS].
- The hash algorithm used MUST be SHA-256 [SHS].

Hash algorithms are not identified by themselves in certificates or BGPsec UPDATE messages. They are represented by an OID that combines the hash algorithm with the digital signature algorithm as follows:

- The ecdsa-with-SHA256 OID [RFC5480] MUST appear in the Public-Key Cryptography Standards #10 (PKCS #10) signatureAlgorithm field [RFC2986] or in the Certificate Request Message Format (CRMF) POPOSigningKey algorithm field [RFC4211]; where the OID is placed depends on the certificate request format generated.

- In BGPsec UPDATE messages, the ECDSA with SHA-256 algorithm suite identifier value 0x01 (1) (see Section 7) is included in the Signature_Block List’s Algorithm Suite Identifier field.

3. Asymmetric Key Pair Formats

The key formats used to compute signatures on CA certificates, BGPsec Router Certificates, and CRLs are as specified in Section 3 of [RFC7935]. This section addresses key formats found in the BGPsec Router requests and in BGPsec Router Certificates.

3.1. Asymmetric Key Pair for Algorithm ID 0x01 (1) - (ECDSA-P256)

The ECDSA private keys used to compute signatures for certificate requests and BGPsec UPDATE messages MUST be associated with the P-256 curve domain parameters [RFC5480]. The public key pair MUST use the uncompressed form.

3.1.1. Public Key Format

The Subject’s public key is included in subjectPublicKeyInfo [RFC5280]. It has two sub-fields: algorithm and subjectPublicKey. The values for the structures and their sub-structures follow:

- algorithm (an AlgorithmIdentifier type): The id-ecPublicKey OID MUST be used in the algorithm field, as specified in Section 2.1.1 of [RFC5480]. The value for the associated parameters MUST be secp256r1, as specified in Section 2.1.1.1 of [RFC5480].
subjectPublicKey: ECPoint MUST be used to encode the certificate’s subjectPublicKey field, as specified in Section 2.2 of [RFC5480].

3.1.2. Private Key Format

Local policy determines private key format.

4. Signature Formats

The structure for the certificate’s and CRL’s signature field MUST be as specified in Section 4 of [RFC7935]; this is the same format used by other RPKI certificates. The structure for the certification request’s and BGPsec UPDATE message’s signature field MUST be as specified in Section 2.2.3 of [RFC3279].

5. Additional Requirements

It is anticipated that BGPsec will require the adoption of updated key sizes and a different set of signature and hash algorithms over time, in order to maintain an acceptable level of cryptographic security. This profile should be updated to specify such future requirements, when appropriate.

The recommended procedures to implement such a transition of key sizes and algorithms are specified in [RFC6916].

6. Security Considerations

The security considerations of [RFC3279], [RFC5480], [RFC6090], [RFC7935], and [RFC8209] apply to certificates. The security considerations of [RFC3279], [RFC6090], [RFC7935], and [RFC8209] apply to certification requests. The security considerations of [RFC3279], [RFC6090], and [RFC8205] apply to BGPsec UPDATE messages. No new security considerations are introduced as a result of this specification.
7. IANA Considerations

The Internet Assigned Numbers Authority (IANA) has created the "BGPsec Algorithm Suite Registry" in the Resource Public Key Infrastructure (RPKI) group. The one-octet "BGPsec Algorithm Suite Registry" identifiers assigned by IANA identify the digest algorithm and signature algorithm used in the BGPsec Signature_Block List’s Algorithm Suite Identifier field.

[RFC8208] directed IANA to register a single algorithm suite identifier for the digest algorithm SHA-256 [SHS] and for the signature algorithm ECDSA on the P-256 curve [RFC6090] [DSS]. This identifier is still valid, and IANA has updated registration to refer to this document.

IANA is asked to modify the previously registered "Unassigned" address space.

<table>
<thead>
<tr>
<th>Algorithm Suite Identifier</th>
<th>Digest Algorithm</th>
<th>Signature Algorithm</th>
<th>Specification Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x2-0xEF</td>
<td>Unassigned</td>
<td>Unassigned</td>
<td></td>
</tr>
</tbody>
</table>

To be modified to:

<table>
<thead>
<tr>
<th>Algorithm Suite Identifier</th>
<th>Digest Algorithm</th>
<th>Signature Algorithm</th>
<th>Specification Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x02-0xF6</td>
<td>Unassigned</td>
<td>Unassigned</td>
<td></td>
</tr>
</tbody>
</table>

In addition IANA is asked to register the following address space for "Documentation" and "Experimentation":

<table>
<thead>
<tr>
<th>Algorithm Suite Identifier</th>
<th>Digest Algorithm</th>
<th>Signature Algorithm</th>
<th>Specification Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xF7-0xFA</td>
<td>Experimentation</td>
<td>Experimentation</td>
<td>This Document</td>
</tr>
<tr>
<td>0xFB-0xFE</td>
<td>Documentation</td>
<td>Documentation</td>
<td>This Document</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm Suite Identifier</th>
<th>Digest Algorithm</th>
<th>Signature Algorithm</th>
<th>Specification Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After the requested modification, the "BGPsec Algorithm Suite Registry" in the RPKI group should contain the following values:

<table>
<thead>
<tr>
<th>Algorithm Suite Identifier</th>
<th>Digest Algorithm</th>
<th>Signature Algorithm</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Reserved</td>
<td>Reserved</td>
<td>This document</td>
</tr>
<tr>
<td>0x01</td>
<td>SHA-256</td>
<td>ECDSA P-256</td>
<td>[SHS] [DSS] [RFC6090]</td>
</tr>
<tr>
<td>0x02-0xF6</td>
<td>Unassigned</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>0xF7-0xFA</td>
<td>Experimentation</td>
<td>Experimentation</td>
<td>This Document</td>
</tr>
<tr>
<td>0xFB-0xFE</td>
<td>Documentation</td>
<td>Documentation</td>
<td>This Document</td>
</tr>
<tr>
<td>0xFF</td>
<td>Reserved</td>
<td>Reserved</td>
<td>This document</td>
</tr>
</tbody>
</table>

Future assignments are to be made using the Standards Action process defined in [RFC8126]. Assignments consist of the one-octet algorithm suite identifier value and the associated digest algorithm name and signature algorithm name.
8. References

8.1. Normative References


8.2. Informative References


Appendix A. Examples

A.1. Topology and Experiment Description

Topology:

AS(64496)----AS(65536)----AS(65537)

Prefix Announcement: AS(64496), 192.0.2.0/24, 2001:db8::/32

The signature algorithm used in this example is ECDSA P-256 using the algorithm suite identifier ID 0x01 (1) as specified in Section 7 of this document.

A.2. Keys

For this example, the ECDSA algorithm was provided with a static k to make the result deterministic.

The k used for all signature operations was taken from [RFC6979], Appendix A.2.5, "Signatures With SHA-256, message = ‘sample’".

Note: Even though the certificates below are expired, the are still useful within the constraint of this document.

\[ k = \text{A6E3C57DD01ABE90086538398355DD4C3B17AA873382B0F24D6129493D8AAD60} \]

Keys of AS64496:

ski: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154

private key:
\[ x = \text{D8AA4DFBE2478F86E88A7451BF075565709C575AC1C136D081C540254CA440B9} \]

public key:
\[ \text{Ux = 7391BABB92A0CB3BE10E59B19EBFB21} \]
\[ \text{Uy = A05B8E695678E0FA16904B55D9D4F5C0} \]
\[ \text{DFC5895EE50BC4F75D205A25BD36FF5} \]
Certificate:

Data:

Version: 3 (0x2)
Serial Number: 38655612 (0x24dd67c)
Signature Algorithm: ecdsa-with-SHA256
Issuer: CN=ROUTER-0000FBF0
Validity
Not Before: Jan 1 05:00:00 2017 GMT
Not After : Jul 1 05:00:00 2018 GMT
Subject Public Key Info:
Public Key Algorithm: id-ecPublicKey
Public-Key: (256 bit)
pub:
a2:5b:d3:6f:f5
ASN1 OID: prime256v1
X509v3 extensions:
X509v3 Key Usage:
Digital Signature
X509v3 Subject Key Identifier:
AB:4D:91:0F:55:CA:E7:1A:21:5E:
F3:CA:FE:3A:CC:45:B5:EE:C1:54
X509v3 Extended Key Usage:
1.3.6.1.5.5.7.3.30
sbgp-autonomousSysNum: critical
Autonomous System Numbers:
64496
Routing Domain Identifiers:
inherit
Signature Algorithm: ecdsa-with-SHA256

Turner & Borchert Expires October 17, 2019 [Page 14]
-----BEGIN CERTIFICATE-----
MIIBiDCCAS+gAwIBAgIEAk3WfDAKBggqhkjOPQQDAjAaMRgwFgYDVQQDDDA9ST1VURVItMDAwMEZCRjAwHhcNMTcwMTAxMDUwMDAwWhcNMTgwNzAxMDUwMDAwWjAaMRgwFgYDVQQDDDA9ST1VURVItMDAwMEZCRjAwWTATBgcqhkjOPQIBbghhkjOPQMBBwNC
AAPk9a7b7kqDL0+EOWbGev/shTgSpHgy6Gx0afTjZD3f1WqBbmlWeOD6FpBLVdnU9cDfxY1V71C8T3XSbAjb0/1o2MWYTALBqN6NHQ8EBAMCB4AwHQYDVVR0BBYEFDNkQ9YucaIV7zyv46zEW17sFUMMGAA1udJQMMMAoGCCsGAQUFBwMEb4GCCGOAQUFBwEIAQH/BA8wDaAHMAUCAwD78KECBQAwCqYIko2IZj0EAwIDRwAwRAIgB7e0al+k8cxoNjkDpIPsfICv0YInUay7Cp75Kzb7EC1ACBUq9bAYnSck6LQi/dEc8D2xOCRdZCIK13uDDgp
-----END CERTIFICATE-----

Keys of AS(65536):
==================
ski: 47F23BF1AB2F8A9D26864EBBD08DF2711C74406EC

private key:
  x = 6CB2E931B112F24554BCDCAAFD9553A9
  519A9AF33C023B60846A21FC95583172

public key:
  Ux = 28FC5FE9AFCF5F4CAB3F5F85CB212FC1
  E9D0EE0DBEEAE425B2F0D3175AA0E989
  Uy = EA9B603E38F35FB32DF495641F2BA04
  0F1C3AC6138307F257CBA63B8B588F41F
Certificate:

Data:

Version: 3 (0x2)
Serial Number: 3752143940 (0xdfa52c44)
Signature Algorithm: ecdsa-with-SHA256
Issuer: CN=ROUTER-00010000
Validity
Not Before: Jan 1 05:00:00 2017 GMT
Not After: Jul 1 05:00:00 2018 GMT
Subject Public Key Info:
Public Key Algorithm: id-ecPublicKey
Public-Key: (256 bit)
pub:
2f:c1:e9:d0:e0:db:ea:ee:42:5b:d2:f0:d3:17:5a:
b8:b5:88:f4:1f
ASN1 OID: prime256v1
X509v3 extensions:
X509v3 Key Usage:
Digital Signature
X509v3 Subject Key Identifier:
47:F2:3B:F1:AB:2F:8A:9D:26:86:
X509v3 Extended Key Usage:
1.3.6.1.5.5.7.3.30
sbgp-autonomousSysNum: critical
Autonomous System Numbers:
65536
Routing Domain Identifiers:
inherit

Signature Algorithm: ecdsa-with-SHA256
Internet Draft    BGPsec Algs, Key & Signature Formats    April 15, 2019

-----BEGIN CERTIFICATE-----
MIIBijCCATCgAwIBAgIFAN+lLEQwCgYIKoZIzj0EAwIwGjEYMBYGA1UEAwwPUk9V
VEVSLTAwMDAwMDAwMDB4XzExMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MjEwMAYGAEwEwYH/0QXx0dAQoGAEwMAYGAEgEwYH/x2gwCgYIKoZIzj0CAQYIKoZIzj0
DAQcD
QgAEPKpx6a/PX0yrP1+FyyEvwenQ4Nvq7kJb0vDFTlqg9Ynmq2A+OPNfsynfSVZB
8roEDxw6xh0DB/JXy6a4tYj0H/H6n/jMGEwCwYDVR0PBAQDAQAgeAMB0GA1uDgQWBRRH
8jvxq+yKnSaGTrv3ycRxcQ7DAQ7bqNVHSUEDDAKbBgrBgEBQcDHJaeBgrBgEF
BQcCAEEB/wQFMMA2gBzaAFqMBAAChAgUAMAoGCCqGSM49BAMCA0qAMEUCIQCM2qS
loiqAhgoIYXTEA7jU46Fqucgn+mGcBeGl3jAigX46uxBmviTxsaha9tmIZnt
puStZDy6v0T7y7dQkXQ=
-----END CERTIFICATE-----

A.3.  BGPsec IPv4

BGPsec IPv4 UPDATE from AS(65536) to AS(65537):

===============================================

Binary Form of BGPsec UPDATE (TCP-DUMP):

FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
01 03 02 00 00 00 EC 40 01 01 02 80 04 04 00 00
00 00 80 0E 0D 00 01 01 04 C6 33 64 64 00 18 C0
00 02 0A 1E 00 CD 00 0E 01 00 00 01 00 00 01 00
00 00 FB F0 00 BF 00 17 F2 3B F1 AB 2F 8A 9D 26
86 4E BB D8 DF 27 11 C7 44 06 EC 00 48 30 46 02
21 00 EF D4 8B 2A AC B6 A8 FD 11 40 DD 9C D4 5E
81 D6 9D 2C 87 7B 56 AA F9 91 C3 4D 0E 8A 4E AF
37 16 02 18 00 9D F0 C1 29 AB B2 F3 9B 6A 07 96
3B D5 55 A8 7A B2 B7 33 3B 7B 91 F1 66 8F D8 61
8C B3 FA C3 F1 AB 4D 91 0F 55 CA E7 1A 21 5E F3
CA FE 3A CC 45 B5 EE C1 54 00 48 30 46 02 21 00
EF D4 8B 2A AC B6 A8 FD 11 40 DD 9C D4 5E 81 D6
9D 2C 87 7B 56 AA F9 91 C3 4D 0E 8A 4E AF 37 16
02 21 00 8E 21 F6 0E 44 C6 06 6C 8B 8A 95 A3 C0
9D 3A D4 37 95 85 A2 D7 28 EE AD 07 A1 7E D7 AA
05 5E CA

Signature from AS(64496) to AS(65536):

===============================================

Digest: 21 33 E5 CA A0 26 BE 07 3D 9C 1B 4E FE B9 B9 77
9F 20 F8 F5 F5 DE 29 FA 98 40 00 9F 60 47 D0 81 54

Signature: 30 46 02 21 00 EF D4 8B 2A AC B6 A8 FD 11 40 DD
9C D4 5E 81 D6 9D 2C 87 7B 56 AA F9 91 C3 4D 0E
A8 4E AF 37 16 02 21 00 8E 21 F6 0E 44 C6 06 6C
8B 8A 95 A3 C0 9D 3A D4 37 95 85 A2 D7 28 EE AD
07 A1 7E D7 AA 05 5E CA
Signature from AS(65536) to AS(65537):
--------------------------------------
Digest: 01 4F 24 DA E2 A5 21 90 B0 80 5C 60 5D B0 63 54
       22 3E 93 BA 41 1D 3D 82 A3 EC 26 52 0C 5F 84
Signature: 30 46 02 21 00 EF D4 8B 2A AC B6 A8 FD 11 40 DD
    9C D4 5E 81 D6 9D 2C 87 7B 56 AA F9 91 C3 4D 0E
   A8 4E AF 37 16 02 21 00 90 F2 C1 29 AB B2 F3 9B
  6A 07 96 3B D5 55 A8 7A B2 B7 33 3B 7B 91 F1 66
  8F D8 61 8C 83 FA C3 F1

The human-readable output is produced using bgpsec-io, a BGPsec traffic generator that uses a Wireshark-like printout.

Send UPDATE Message
  +--marker: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
  +--length: 259
  +--type: 2 (UPDATE)
  +--withdrawn_routes_length: 0
  +--total_path_attr_length: 236
    +--ORIGIN: INCOMPLETE (4 bytes)
      |   +--Flags: 0x40 (Well-Known, Transitive, Complete)
      |   +--Type Code: ORIGIN (1)
      |   +--Length: 1 byte
      |   +--Origin: INCOMPLETE (1)
    +--MULTI_EXIT_DISC (7 bytes)
      |   +--Flags: 0x80 (Optional, Non-transitive, Complete)
      |   +--Type Code: MULTI_EXIT_DISC (4)
      |   +--Length: 4 bytes
      |   +--data: 00 00 00 00
    +--MP_REACH_NLRI (16 bytes)
      |   +--Flags: 0x80 (Optional, Non-transitive, Complete)
      |   +--Type Code: MP_REACH_NLRI (14)
      |   +--Length: 13 bytes
      |   +--Address family: IPv4 (1)
      |   +--Subsequent address family identifier: Unicast (1)
      |   +--Next hop network address: (4 bytes)
      |      |   +--Next hop: 198.51.100.100
      |   +--Subnetwork points of attachment: 0
      |   +--Network layer reachability information: (4 bytes)
      |      |   +--192.0.2.0/24
      |      |   +--MP Reach NLRI prefix length: 24
      |      |   +--MP Reach NLRI IPv4 prefix: 192.0.2.0
++--BGPSEC Path Attribute (209 bytes)
  +--Flags: 0x90 (Optional, Complete, Extended Length)
  +--Type Code: BGPSEC Path Attribute (30)
  +--Length: 205 bytes
  ++--Secure Path (14 bytes)
        ++--Length: 14 bytes
              ++--Secure Path Segment: (6 bytes)
                    ++--pCount: 1
                    ++--Flags: 0
                    ++--AS number: 65536 (1.0)
              ++--Secure Path Segment: (6 bytes)
                    ++--pCount: 1
                    ++--Flags: 0
                    ++--AS number: 64496 (0.64496)
  ++--Signature Block (191 bytes)
        ++--Length: 191 bytes
        ++--Algo ID: 1
        ++--Signature Segment: (94 bytes)
                      ++--SKI: 47F23BF1AB2F8A9D26864EBBD8DF2711C74406EC
                      ++--Length: 72 bytes
                      ++--Signature: 3046022100EFD48B 2AACC6A8FD1140DD
                           9CD45E81D69D2C87 7B56AA9F991C34D0E
                           AB4EAF3716022100 90F2C129ABB2F39B
                           6A07963BD555A87A B2B7333B7B91F166
                           8FD8618C83FAC3F1
                      ++--Signature Segment: (94 bytes)
                      ++--SKI: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154
                      ++--Length: 72 bytes
                      ++--Signature: 3046022100EFD48B 2AACC6A8FD1140DD
                           9CD45E81D69D2C87 7B56AA9F991C34D0E
                           A84EAF3716022100 8E21F60E44C6066C
                           8B8A95A3C09D3AD4 379585A2D728EEAD
                           07A17ED7AA055ECA
A.4. BGPsec IPv6

BGPsec IPv6 UPDATE from AS(65536) to AS(65537):

Binary Form of BGP/BGPsec UPDATE (TCP-DUMP):

FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
01 10 02 00 00 00 F9 40 01 01 02 80 04 04 00 00
00 00 80 0E 1A 00 02 01 10 FD 00 00 00 00 00 00
00 00 00 00 00 C6 33 64 64 00 20 20 01 0D B8 90
1E 00 CD 00 0E 01 00 00 01 00 00 01 00 00 00 FB
F0 00 BF 01 47 F2 3B F1 AB 2F 8A 9D 26 86 4E BB
D8 DF 27 11 C7 44 06 EC 00 48 30 46 02 21 00 EF
D4 8B 2A AC B6 A8 FD 11 40 DD 9C D4 5E 81 D6 9D
2C 87 7B 56 AA F9 91 C3 4D 0E A8 4E AF 37 16 02
21 00 D1 B9 4F 62 51 04 6D 21 36 A1 05 B0 F4 72
7C C5 BC D6 74 D9 7D 2B E6 1B 8F 43 BD DE 91 C3
06 26 AB 4D 91 0F 55 CA E7 1A 21 5E F3 CA FE 3A
CC 45 B5 EE C1 54 00 48 30 46 02 21 00 EF D4 8B
2A AC B6 A8 FD 11 40 DD 9C D4 5E 81 D6 9D 2C 87
7B 56 AA F9 91 C3 4D 0E A8 4E AF 37 16 02 21 00
E2 A0 2C 68 FE 53 CB 96 93 4C 78 1F 5A 14 A2 97
19 79 20 0C 91 56 ED F8 55 05 8E 80 53 F4 AC D3

Signature from AS(64496) to AS(65536):

Digest: 8A 0C D3 E9 8E 55 10 45 82 1D 80 46 01 D6 55 FC
52 11 89 DF 4D B0 28 7D 84 AC FC 77 55 6D 06 C7
Signature: 30 46 02 21 00 EF D4 8B 2A AC B6 A8 FD 11 40 DD
9C D4 5E 81 D6 9D 2C 87 7B 56 AA F9 91 C3 4D 0E
A8 4E AF 37 16 02 21 00 E2 A0 2C 68 FE 53 CB 96
93 4C 78 1F 5A 14 A2 97 19 79 20 0C 91 56 ED F8
55 05 8E 80 53 F4 AC D3

Signature from AS(65536) to AS(65537):

Digest: 44 49 EC 70 8D EC 5C 85 00 C2 17 8C 72 FE 4C 79
FF A9 3C 95 31 61 01 2D EE 7E EE 05 46 AF 5F D0
Signature: 30 46 02 21 00 EF D4 8B 2A AC B6 A8 FD 11 40 DD
9C D4 5E 81 D6 9D 2C 87 7B 56 AA F9 91 C3 4D 0E
A8 4E AF 37 16 02 21 00 D1 B9 4F 62 51 04 6D 21
36 A1 05 B0 F4 72 7C C5 BC D6 74 D9 7D 2B 8E 1B
8F 43 BD DE 91 C3 06 26
The human-readable output is produced using bgpsec-io, a BGPsec traffic generator that uses a Wireshark-like printout.

Send UPDATE Message

```
+--marker: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
+--length: 272
+--type: 2 (UPDATE)
+--withdrawn_routes_length: 0
+--total_path_attr_length: 249
  +--ORIGIN: INCOMPLETE (4 bytes)
    +--Flags: 0x40 (Well-Known, Transitive, Complete)
    +--Type Code: ORIGIN (1)
    +--Length: 1 byte
    +--Origin: INCOMPLETE (1)
  +--MULTI_EXIT_DISC (7 bytes)
    +--Flags: 0x80 (Optional, Non-transitive, Complete)
    +--Type Code: MULTI_EXIT_DISC (4)
    +--Length: 4 bytes
    +--data: 00 00 00 00
  +--MP_REACH_NLRI (29 bytes)
    +--Flags: 0x80 (Optional, Non-transitive, Complete)
    +--Type Code: MP_REACH_NLRI (14)
    +--Length: 26 bytes
    +--Address family: IPv6 (2)
    +--Subsequent address family identifier: Unicast (1)
    +--Next hop network address: (16 bytes)
      +--Next hop: fd00:0000:0000:0000:0000:0000:c633:6464
    +--Subnetwork points of attachment: 0
    +--Network layer reachability information: (5 bytes)
      +--2001:db8::/32
      +--MP Reach NLRI prefix length: 32
      +--MP Reach NLRI IPv6 prefix: 2001:db8::
```
---BGPSEC Path Attribute (209 bytes)
  +--Flags: 0x90 (Optional, Complete, Extended Length)
  +--Type Code: BGPSEC Path Attribute (30)
  +--Length: 205 bytes
  +--Secure Path (14 bytes)
    +--Length: 14 bytes
    |   +--Secure Path Segment: (6 bytes)
    |     +--pCount: 1
    |     +--Flags: 0
    |     +--AS number: 65536 (1.0)
    +--Secure Path Segment: (6 bytes)
     +--pCount: 1
     +--Flags: 0
     +--AS number: 64496 (0.64496)
  +--Signature Block (191 bytes)
    +--Length: 191 bytes
    +--Algo ID: 1
    +--Signature Segment: (94 bytes)
      +--SKI: 47F23BF1AB2F8A9D26864EBBD8DF2711C74406EC
      +--Length: 72 bytes
      |   +--Signature: 3046022100EFD48B 2AACB6A8FD1140DD
      |   |   |   9CD45E81D69D2C87 7B56AAF991C34D0E
      |   |   |   A84EAF3716022100 D1B94F6251046D21
      |   |   |   36A105B0F4727CC5 BCD674D97D28E61B
      |   |   |   8F43BDDE91C30626
      +--Signature Segment: (94 bytes)
        +--SKI: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154
        +--Length: 72 bytes
        +--Signature: 3046022100EFD48B 2AACB6A8FD1140DD
        |   |   9CD45E81D69D2C87 7B56AAF991C34D0E
        |   |   A84EAF3716022100 E2A02C68FE53CB96
        |   |   934C781F5A14A297 1979200C9156EDF8
        |   |   55058E8053F4ACD3
        +--Signature Segment: (94 bytes)
Acknowledgements

The authors wish to thank Geoff Huston and George Michaelson for producing [RFC7935], which this document is entirely based on. The authors would also like to thank Roque Gagliano, David Mandelberg, Tom Petch, Sam Weiler, and Stephen Kent for their reviews and comments. Mehmet Adalier, Kotikalapudi Sriram, and Doug Montgomery were instrumental in developing the test vectors found in Appendix A. Additionally we want to thank Geoff Huston, author of [RFC5398] from where we borrowed wording for Section 2.1 of this document.

Authors’ Addresses

Sean Turner
sn3rd

Email: sean@sn3rd.com

Oliver Borchert
NIST
100 Bureau Drive
Gaithersburg, MD 20899
United States of America

Email: oliver.borchert@nist.gov