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 updates RFC 7935 ("The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure").

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 is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8208.
1. Introduction

This document specifies the following:

- the digital signature algorithm and parameters,
- the hash algorithm and parameters,
- 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 to minimize the size of the protocol exchanged between routers.

Appendix A 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.
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; for example, these algorithms are used by BGPsec routers to request BGPsec certificates, by RPKI CAs to verify BGPsec certification requests, by BGPsec routers to generate BGPsec UPDATE messages, and by BGPsec routers to verify BGPsec UPDATE messages:

- 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 0x1 (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 Certificate requests and in BGPsec Router Certificates.

The ECDSA private keys used to compute signatures for certificate requests and BGPsec UPDATE messages MUST come from the P-256 curve [RFC5480]. The public key pair MUST use the uncompressed form.
3.1. 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].

3.2. 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.

IANA has registered 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].

**BGPsec Algorithm Suite Registry**

<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>0x0</td>
<td>Reserved</td>
<td>Reserved</td>
<td>This document</td>
</tr>
<tr>
<td>0x1</td>
<td>SHA-256</td>
<td>ECDSA P-256</td>
<td>[SHS] [DSS] [RFC6090]</td>
</tr>
<tr>
<td>0x2-0xEF</td>
<td>Unassigned</td>
<td>Unassigned</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

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’".

\[ k = A6E3C57DD01ABE90086538398355DD4C3B17AA873382B0F24D6129493D8AAD60 \]

Keys of AS64496:
================
ski: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154

private key:
\[ x = D8AA4DFBE2478F86E88A7451BF075565709C575AC1C136D081C540254CA440B9 \]

public key:
\[ Ux = 7391BABB92A0CB3BE10E59B19EBFFB214E04A91E0CBA1B139A7D38D90F77E55A \]
\[ Uy = A05B8E695678E0FA16904B55D9D4F5C0DFC58895EE50BC4F75D205A25BD36FF5 \]
Router Key Certificate example using OpenSSL 1.0.1e-fips 11 Feb 2013
--------------------------------------------------------------------
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: CN=ROUTER-0000FBF0
Subject Public Key Info:
  Public Key Algorithm: id-ecPublicKey
  Public-Key: (256 bit)
  pub:
    a2:5b:db: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
-----BEGIN CERTIFICATE-----
MIIBiDCCAS+gAwIBAgIEAk3WfDAKBggqhkjOPQQDAjAaMRgwFgYDVQQDDA9ST1V
RUItMDAwMEZCRjAwHhcNMjcxMDUwMDAwMDcQaPllbGdpZ2UwMDAwMDAwWjAaMRgw
FgYDVQQDA9ST1VURVItMDAwMEZCRjAwWTATBgcqhkjOPQIBbghqkjoPQMBBwNC
AAPzkbq7kqDL0+EOwGbGev/shTgSpHgy6GxOaftjZj3fWqBbmlWeOD6fpBLVdnU
9cDfxfYiV7lC873XSbAjb0/1s2MwYyALbGNVHQ8EBAMCB4AwHQYDVR0BBYE
kQ9VucaIV7zyv46zEw17sFUMBMAAIJQCMaOGCCsGAQUBwMeMB4GCCsGAQUF
BwEIAQH/BA8wDaAHMAUCAwD78KCEBOwCgYIKoZIzj0EAwIDRwAwRAIgB7e0a1+k
8czoNjknk5sPsfIC0vYInUay7Cp75pKzb7ECIACRBUqh9bAYnSck6LQi/dEc8D2x
OCRdZCK1KI3uDDgp
-----END CERTIFICATE-----

Keys of AS(65536):
==================
ski: 47F23BF1AB2F8A9D26864EBBD8DF2711C74406EC

private key:
  x = 6CB2E931B112F24554BCDCAAFD9553A9
  519A9AF33C023B60846A21FC95583172

public key:
  Ux = 28FC5FE9AFCF54CBAB3F5F85CB212FC1
  E9D000E08EBAEE425BD2F0D3175AA0E989
  Uy = EA9B603E38F35FB329DF49564F2BA04
       0F1C3AC6138307F257CBA6B8B588F41F
Router Key Certificate example using OpenSSL 1.0.1e-fips 11 Feb 2013
--------------------------------------------------------------------
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: CN=ROUTER-00010000
  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:ff: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
-----BEGIN CERTIFICATE-----
MIIBijCCATCcAwIBAgI+1LEQwCgYIKoZIzj0EAwIwGjEYMBYGA1UEAwwPUk9V
VEVSLTAwMD4wMDAwMB4XDTE3MDAwMTE1MDAwMFowGjEY
MBYGA1UEAwwPUk9VVEVSLTAwMD4wMDAwMFOwGjEY
QgAEKPxm6a/PX0yrP1+FyyEywenQ4Nvq7kJb0vDFTlqg6Ynm2A+OPNfsyhfSVZB
8rOE6xw6x0DB/JXy6a4tYj0H6NjMGEwCwYDVQR0BAQDJageAMB0GA1uDBQWBBRH
8jvxqy+KnSaGTrvY3ycRnxQ7DGATbqNHVSEdADKBggrBgEFQh0BAQdGAIaBgEF
BQcCAEB/wQFM2gBzAFAgMBAAChAgUAMoGCCqGSM49BAMCA0qAMEUCIQCM2fS
loiCdA0hgoIYxTEA7jU46Fqucgn+mGcBeGj3jAIgX+46vxBmviTsxaha9tmIZnt
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 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 82 AC B6 A8 FD 11 40 DD 3C 45 5E
81 D6 9D 2C 87 7B 5E 1A F9 91 C3 4D 0E A8 4E AF
37 16 02 01 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 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 3C 45 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 A5
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 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 A5 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 36 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: FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF
+--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 2AACB6A8FD1140DD
                    9CD45E81D69D2C87 7B56AAF991C34D0E
                    A84EAF3716022100 90F2C129ABB2F39B
                    6A07963BD555A87A  B2B7333B7B91F166
                    8FD8618C83FAC3F1
    ---Signature Segment: (94 bytes)
      ---SKI: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154
      ---Length: 72 bytes
      ---Signature: 3046022100EFD48B 2AACB6A8FD1140DD
                    9CD45E81D69D2C87 7B56AAF991C34D0E
                    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 20 01 00 10 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 2E 8E 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 28 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 9E 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 OC 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 9E
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 28 E6 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

```plaintext
+--marker: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
+--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)
    | +--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 7B56AAAF991C34D0E

##### A84EAF3716022100 D1B94F6251046D21

##### 36A105B0F4727CC5 BCD674D97D28E61B

##### 8F43BDDE91C30626

#### Signature Segment: (94 bytes)

##### SKI: AB4D910F55CAE71A215EF3CAFE3ACC45B5EEC154

##### Length: 72 bytes

##### Signature: 3046022100EFD48B 2AACB6A8FD1140DD

##### 9CD45E81D69D2C87 7B56AAAF991C34D0E

##### A84EAF3716022100 E2A02C68FE53CB96

##### 934C781F5A14A297 1979200C9156EDF8

##### 55058E8053F4ACD3
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.

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