Internet Engineering Task Force
INTERNET-DRAFT
draft-weis-sobgp-certificates-01.txt
Expires: April, 2004   October, 2003

Brian Weis, Editor
Cisco Systems

Secure Origin BGP (soBGP) Certificates

Status of this Memo

Abstract

This document describes the format of digital certificates that are used by the Secure Origin BGP (soBGP) extensions to BGP, as well as acceptable use of those certificates. Included are certificates providing authentication, authorization, and policy distribution.

Table of Contents

1.0 Introduction

There is a great deal of concern over the security of routing systems within the Internet. This is particularly true in relation to the Border Gateway Protocol [BGP], the protocol used to provide routing information between Autonomous Systems (ASes). Source Origin BGP (soBGP) provides a method that ASes can use to determine the correctness of BGP messages received by their BGP routers. It also provides a method for ASes to detect implausible routes reported in a BGP Update AS_PATH, and acts as an aid in detecting misconfigured routers advertising incorrect routes.

Source Origin BGP does not define changes to BGP Updates. Rather, it provides authorization and path policy "out-of-band" from the BGP Updates. An AS compares the information claimed in BGP Updates to the soBGP policy, and makes judgments to the fitness of the claim.

Source Origin BGP distributes authorization and policy as digitally signed objects, which can be distributed in many ways. To aid interoperability, extensions have been defined in [SOBGP-BGP] that support distribution of the digitally signed soBGP objects within BGP itself..

Source Origin BGP deployment models are discussed in [SOBGP-DEPLOY].

Extensions to RADIUS to support soBGP are defined in [SOBGP-RADIUS].

This document defines the format of the digitally signed objects used by soBGP, as well as the operations to be performed on those objects.

1.1 Key Words

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 [RFC2119].

1.2 Terminology

This document frequently uses the following terms:

AS Policy Certificate (ASPolicycert)
A digital certificate that asserts routing policy for an Autonomous System.

Authorization Certificates (Authcerts)
A digital certificate that asserts that an Autonomous System is authorized to advertise a particular prefix.

Entity
Participants within the routing system. These include Regional Internet Registry (RIR) authorities, Local Internet Registry (LIR) authorities, Internet Service Providers (ISPs), and other

organizations participating in soBGP. An Entity must have an Autonomous System (AS) number assigned to it as a unique identity, even if it does not source routes within the routing system.

Entity Certificate (Entitycert)
An X.509 certificate that asserts a mapping between an Autonomous System identifier and a public key.

Prefix Policy Certificate (Prefixpolicycert)
A digital certificate mapping usage policy to one or more prefixes.

Regional Internet Registry (RIR)
An entity recognized by IANA and tasked with managing IP address space within a wide geographical area. RIRs allocate address space to Local Internet Registries and other entities.

Local Internet Registry (LIR)
An entity that allocates address space to the users of the network services that it provides.

2.0 Overview

Source Origin BGP refers to participants within the routing system as entities. Each entity must have an Autonomous System (AS) number, issued from an authorized entity (e.g., Regional Internet Registry), to participate in soBGP. Entities may have one or more roles within soBGP. They may act as a trusted signer, an authorizer of address blocks, and/or as a route originator.

Source Origin BGP provides a method of verifying that an AS is authorized to advertise certain prefixes. The authorization to advertise prefixes or a given address space is validated through Authorization Certificates (Authcerts). Authcerts are issued by entities (e.g., ISP) that allocate prefixes.

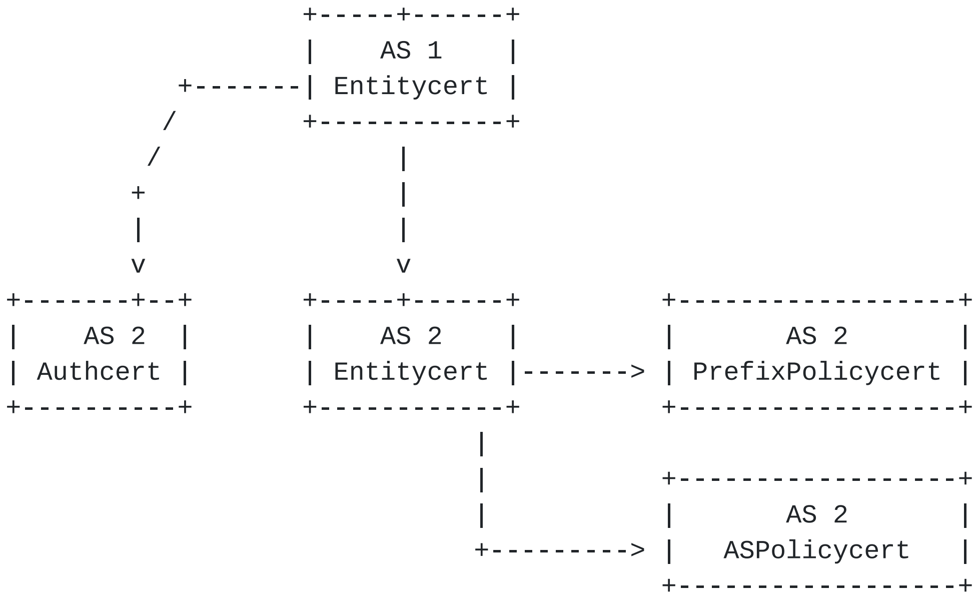
An AS given an Authcert (e.g., ISP customer) may assign local policy to be used with the prefixes listed in the Authcert using a Prefix Policy Certificate (PrefixPolicycert).

Policies specific to an Autonomous System are provided through AS Policy Certificates (ASPolicycerts). This policy enables another entity to develop a database of plausible paths through the routing system, and aids in detecting impossible and fraudulent paths.

Authcerts, PrefixPolicycerts, and ASPolicycerts are verified using public keys embedded in Entity Certificates (Entitycerts). Entitycerts are X.509 certificates as specified by [RFC3280].

Figure 1 illustrates the relationship between soBGP certificates for a single AS. AS 1 allocates a prefix to AS 2. AS 1 also issues an Authcert to AS 2 proving that AS 2 may legitimately use that prefix. In this example, AS 1 also acts as an Entitycert issuer for AS 2. AS

2 then creates two policy certificates: one specifying particular policy for the authorized prefix, and one specifying particular policy for the AS.



```
                      +-----+------+
                      |    AS 1    |
              +-------| Entitycert |
             /        +------------+
            /               |
           +                |
           |                |
           v                v
   +-------+--+       +-----+------+         +------------------+
   |    AS 2  |       |    AS 2    |         |       AS 2       |
   | Authcert |       | Entitycert |-------> | PrefixPolicycert |
   +----------+       +------------+         +------------------+
                                 |
                                 |           +------------------+
                                 |           |       AS 2       |
                                 +---------> |   ASPolicycert   |
                                             +------------------+
```

Figure 1. Relationship between soBGP certificates

Each of the soBGP certificates is discussed in detail in subsequent sections of this document.

2.1 Digital Signature Algorithms

The RSA Public Key Algorithm [RSA] is a widely deployed public key algorithm commonly used for digital signatures. Compared to other public key algorithms, signature verification is relatively quick. This property is useful considering the large number of signature verifications that will be done on soBGP certificates. The RSA Algorithm is commonly supported in hardware, and is no longer encumbered by intellectual property claims.

All soBGP implementations MUST support a digital signature of a SHA1 digest encrypted with the RSA algorithm. An implementation MAY support other signature methods, but any AS using alternate signature methods run the risk of their signatures not being universally verifiable.

3.0 Entity Certificate (Entitycert)

Entitycerts provide authentication, providing a binding of an identity (i.e., autonomous system number) to a public key. The authenticity of the binding is verified with a digital signature, where the public key of the certificate issuer has been previously accepted by an receiver as valid. Issuer public keys can either be

manually configured, or are verified through the use of another issuer's trusted public key in a "web of trust" built by the receiver.

Entitycerts are used to verify, through a trust model, the existence of an entity within the routing system, and the value of that entity's public key for use in the routing system. Each entity within the routing system participating in soBGP MUST generate a public/private key pair. The public key portion of this pair is then signed, verifying that anyone using this public key is actually the entity in question. This signature may be provided by various other trusted parties within the routing system, including (but not limited to):

- The authority that issued the autonomous system number.

- An external commercial authority that provides authentication certificates for other commercial transactions.

- Any other trusted party within the domain of Internet routing, such as a well known Service Provider.

- Self-signed if the entity is well known within the routing system.

A public key is used to verify the validity of other messages transmitted by this entity within the routing system.The public key, along with other verifying information, is formatted into an Entitycert, as described in the next section.

3.1 Format

An Entitycert MUST be formatted as an X.509 authentication certificate, as defined in [RFC3280]. The Entitycert MUST be generated with a signature of type sha1withRSAEncryption [RFC3279].

The primary identity in soBGP is the autonomous system number. Because of this, each entity that issues Entitycerts MUST be assigned an AS number, even if they do not originate routes into the internetwork. In accordance with Section 4.2.1.7 of [RFC3280], issuers MUST verify all parts of the subject alternative name, including the AS number, before issuing the certificate.

An Entitycert MUST have a subjectAltname critical extension, which MUST contain the AS number of the subject as an otherName choice. The AS number is encoded with the OID defined in Section 3.2.1 of [ADDR-EXT].

An Entitycert MUST have an issuerAltname critical extension, which MUST contain the AS number of the subject as an otherName choice. The AS number is encoded with the OID defined in Section 3.2.1 of [ADDR-EXT].

The X.509 Issuer and Subject distinguished names are not used by soBGP. In accordance with Section 4.2.1.7 of [RFC3280], when subjectAltName is required, the Subject field MAY be empty.

3.2 Creation

An Entitycert is usually created with the following steps:

- The entity requesting an Entitycert generates a signature key pair
- The entity forwards its identity (including its AS number) and the public key to an Entitycert issuer using the certificate registration mechanism supported by the issuer.
- The issuing autonomous system verifies that the identity of the receiving autonomous system, generates an Entitycert including that identity, and signs it with its own private key.
- The issuing autonomous system returns the Entitycert to the receiving autonomous system.

3.2.1 Certificate Uniqueness

Digital certificates are created as uniquely named objects, which allows them to be uniquely identified. For the purposes of soBGP, the pair of CertificateSerialNumber and IssuerAltName values uniquely identifies entity Certificates. Note that although RFC 3280 contains an X.509v3 IssuerName, it is not used elsewhere within soBGP.

3.2.2 Certificate Encoding

Entitycerts distributed in [SOBGP-BGP] use their native DER [X.690] form. If Entitycerts are manually distributed (e.g., through electronic mail) they may need to be base64 encoded into ASCII as described in Section 4.3 of [RFC1421].

3.2.3 Multiplicity of Entitycerts

An autonomous system MAY enroll with more than one issuer, which results in multiple Entitycerts. An AS holding certificates from different well-known issuing entities within the routing system may result in a greater number of other autonomous systems accepting their public key. Or, it may simply result in other autonomous systems accepting their public key faster, which increases BGP convergence times.

If an entity detects that an autonomous system has valid Entitycerts from different issuers, the entity SHOULD treat the various Entitycerts as independent. Revocation from one issuer does not necessarily imply that Entitycerts from other issuers are invalid. An issuer may revoke a certificate for reasons other than private key compromise or loss.

However, even if an issuer states key compromise as the reason for revocation, a receiving entity SHOULD treat this state as specific to the issuer. Note that if the state of one issuer were instead considered transitive, the erroneous revocation of a single issuer would result in a denial of service attack on the victim autonomous system.

In the face of inconsistent state from different issuers, a receiver MAY choose to trust one issuer over another. For example, a receiver may choose to prefer the result of an issuer they directly trust to an issuer that was verified further away in the "web of trust".

3.3 Distribution

Entitycerts may be distributed using any number of methods, for example:

- maintained in a directory maintained by the issuing autonomous system,
- distributed via some out of band mechanism, and/or
- distributed within BGP using extensions defined in [SOBGP-BGP].

To ensure interoperability, the receiving autonomous system SHOULD distribute its Entitycert within BGP.

3.4 Validation

Any device receiving an Entitycert can verify it by validating the signature on the certificate, along with the verifying information. If a Certificate Revocation List (CRL) is available for that issuer, it MUST be consulted to verify that this certificate has not been revoked. Once validation is complete, the public key contained in this certificate may be used to verify messages purportedly sent by this entity.

3.4.1 Web of Trust

An soBGP entity uses the "web of trust" paradigm for purposes of Entitycert validation, where the entity learns the validity of public keys over time. An entity follows the following procedure for validating Entitycerts in the web of trust.

- A small number of Entitycerts are manually configured and copied to a device's local configuration. These are implicitly trusted as being previously verified and authenticated.
- When the entity receives a new Entitycert, it checks to see if it has the public key of the issuing autonomous system in its configuration. If so, it attempts to validate the Entitycert, using the previously known public key, and any revocation material that is available from the issuer.

- If the new Entitycert proves valid, it is added to the device's local configuration and may be used to validate subsequently received Entitycerts.
- If the new Entitycert cannot be validated because the issuer?s public key is not yet available, local policy dictates as to whether or not the certificate is held awaiting the issuer?s certificate.

Figure 2 shows an example web of trust. In this example, Entitycerts for AS 1 and AS 5 would be manually copied to the local configuration on the box. Other Entitycerts would be validated using the usual PKI path validation techniques.

```
        +------------+                 +------------+
        |    AS 1    |                 |    AS 5    |
        | Entitycert |                 | Entitycert |
        +-----+------+                 +---+----+---+
              |                           /      \
              |                          /        \
              |                         +          +
              |                         |          |
              V                         V          V
        +------------+         +------------+ +------------+
        |    AS 2    |         |    AS 6    | |    AS 7    |
        | Entitycert |         | Entitycert | | Entitycert |
        +---+----+---+         +------------+ +-----+------+
           /      \                                 |
          /        \                                V
         +          +                         +------------+
         |          |                         |    AS 8    |
         V          V                         | Entitycert |
+------------+ +------------+                 +-----+------+
|    AS 3    | |    AS 4    |                       |
| Entitycert | | Entitycert |                       V
+------------+ +------------+                 +------------+
                                              |    AS 9    |
                                              | Entitycert |
                                              +------------+
```

Figure 2. Example Web of Trust

An autonomous system may define local policy to restrict the scope of the web of trust. However it should be noted that any local policy restricting the web of trust reduces the value of soBGP authorization and path validation.

One type of local policy would be to accept only a certain "depth" of Entitycert issuers. For example, consider if AS 6 in Figure 2 only accepted two levels of issuers. AS 6 would only trust ASes 1,2,5,6 and 7 to issue Entitycerts. It would never validate the Entitycert from ASes 3, 4, 8, and 9.

3.4.2 Self-signed Entitycerts

Entitycerts MAY be self-signed, but SHOULD only be accepted from autonomous systems when an alternative method exists of validating that the self-signed certificate is genuine. For example, distribution out-of-band using a trusted delivery procedure would be acceptable.

Typical users of a self-signed Entitycert would be:

- A commercial authority in the business of providing authentication certificates for many types of commercial transactions
- An Entitycert issuer that is at the top of a hierarchy of issuers
- A well-known trusted party within the domain of Internet routing

3.5 Revocation and Expiration

Any entity issuing an Entitycert may have need to revoke it. The entity MAY use any form for propagating that revocation list, but SHOULD also send it as part of an AS Policy Certificate (distributed using [SOBGP-BGP]). This allows autonomous systems that cannot route to the issuing autonomous system to verify that the Entitycert has not been revoked.

If an Entitycert is discarded due to revocation, the Authcert and Policy databases should be examined. Any Authcerts and Policy certificates that were validated using the discarded certificate should be removed from the database.

X.509 certificates contain expiration dates. Any device validating Entitycerts MUST have a time of day clock that is close to real time in order to properly deal with expired certificates

If an Entitycert is discarded due to expiration, an Authcerts or Policy certificates validated using the discarded certificate remain valid if another valid Entitycert for the AS can be found containing the same public key.

4.0 Authorization Certificates (Authcert)

Authcerts prove the right of an entity to advertise particular address spaces. They are generated in a hierarchical manner following the order of address space allocation (i.e., from RIR, to LIR or ISP, to customer), and are distributed along with the address space allocation. Receivers use the Authcert to validate announcements received in BGP UPDATE messages.

The authorization certificate binds one or more prefix blocks to a particular autonomous system. It is typically provided by an entity issuing a prefix block to an autonomous system, and is digitally signed by the issuing autonomous system. The Authcert can be thought of as an "Attribute Certificate" in the spirit of RFC 3281, although it does not follow the syntax of that document.

The authenticity of Authcerts is verified with a digital signature provided by the issuing autonomous system. Authcerts do not contain public keys. Rather, they bind an address space to a particular identity (i.e., autonomous system).

4.1 Format

The Authcert is defined as a header block followed by a set of Type/Length/Value attributes, as identified in the following sections. Each Authcert TLV includes a type, which is treated as a 16 bit (two octet) unsigned integer. The TLVs described must be placed within the Authcert in type order; every Authcert should begin with a TLV type 1 (Autonomous System and Options). All TLVs are REQUIRED to be in an Authcert unless otherwise noted.

4.1.1 Authcert Header

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +---------------+---------------+-------------------------------+
    | Cert. Marker  |    Type Id    | Length                        |
    +---------------+---------------+-------------------------------+
    | TLVs
    +----------------
```

- o Certificate Marker: "162(0xa2), identifying this as an soBGP certificate.
- o Type ID: "1(0x01), identifying this as an Authcert.
- o Length: Set to the length of the TLVs.
- o TLVs: The Type/Length/Value attributes making up an Authcert.

4.1.2 The Authorizing AS

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-------------------------------+-------------------------------+
    | TLV Type                      | Length                        |
    +-------------------------------+-------------------------------+
    | Autonomous System                                             |
    +---------------------------------------------------------------+
```

- o TLV type: 1 (0x0001)
- o Length: Set to 4.

- o AS: (4 octets), the autonomous system authorizing other entities to advertise prefixes within this block. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

Each authorizing entity MUST have an autonomous system number, used as a unique identifier, even though they may not advertise prefixes into the routing system.

4.1.3 Authorized Originator

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Autonomous System                                             |
   +---------------------------------------------------------------+
```

o   TLV type: 2 (0x0002)

o   Length: Set to 4.

o   AS: (4 octets), the autonomous system of an entity authorized to advertise prefixes within this block. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

Multiple authorized originator TLVs may be included in the Authcert.

4.1.4 The Serial Number TLV

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Serial Number                                                 |
   +---------------------------------------------------------------+
```

o   TLV type: 3 (0x0003)

o   Length: Set to 4.

o   Serial Number: (4 octets), unsigned integer taken from a number space maintained by the Authorizing AS indicating the serial number of this Authorization certificate. The Authorizing AS MUST manage the number space as a monotonically increasing value so that a relative ordering of Authcerts is maintained.

4.1.5 Authorizing AS Entitycert Uniform Resource Locator

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | URL                                                           |
   +----------------
```

- o TLV type: 4 (0x0004)
- o Length: Denotes the length of the URL in octets.
- o URL: A uniform resource locator indicating a location where the Authorizing AS?s Entitycert can be found.

An Authcert may omit this TLV. However, an implementation is REQUIRED to correctly parse them if they are present. A receiving device MAY choose to ignore the URL TLV.

4.1.6 Authorizing AS Validation List Uniform Resource Locator

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | URL                                                           |
   +----------------
```

- o TLV type: 5 (0x0005)
- o Length: Denotes the length of the URL in octets.
- o URL: A uniform resource locator indicating a location where the Authorizing AS?s Validation List can be found.

An Authcert may omit this TLV. However, an implementation is REQUIRED to correctly parse them if they are present. A receiving device MAY choose to ignore the URL TLV.

4.1.7 The Address Prefix TLV

The address prefix TLV shall define blocks of address within which the authorized AS' are allowed to advertise prefixes (or routes).

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-------------------------------+-------------------------------+
      | TLV Type                      | Length                        |
      +-------------------------------+---------------+---------------+
      | Address Family Identifier     |   RESERVED    | Subsequent AFI|
      +-------------------------------+---------------+---------------+
      | NLRI Data                                                     |
      +----------------
```

o   TLV Type: 14 (0x000D)

o   Length (2 octets), set to 4 + the length of the NLRI Data.

o   Address Family Identifier: This field carries the identity of the Network Layer protocol associated with the Network Address that follows. Presently defined values for this field are specified in RFC 1700 (see the Address Family Numbers section).

o   RESERVED: Set to 0.

o   Subsequent AFI: This field provides additional information about the type of the Network Layer Reachability Information carried in the attribute.

o   NLRI Data: An address prefix as described in Section 4 of [RFC2858].

4.1.8 Signature

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-------------------------------+-------------------------------+
      | TLV Type                      | Length                        |
      +-------------------------------+-------------------------------+
      | Signature Type                | Number of Issuers             |
      +-------------------------------+-------------------------------+
      | Entity Certificate Issuer Autonomous System                   |
      +-------------------------------+-------------------------------+
      | Entity Certificate Serial Number                              |
      +-------------------------------+-------------------------------+
      | ...                                                           |
      +---------------------------------------------------------------+
      | Signature                                                     |
      +------------------
```

o   TLV type: 65535 (0xFFFF)

o   Length: (2 octets), unsigned integer denoting the length of the payload bytes which follow.

- Signature Type: (2 octets), unsigned integer denoting the type of signature (the algorithm used to build this signature). Each possible signing algorithm is assigned an integer from this field. Signature type 1 is defined as an RSA encryption of a SHA1 digest.

- Number of Issuers (2 octets): The number of Entitycert references included in the signature payload. If more than one Entitycert reference follows, all Entitycerts MUST contain the same public key for the same authorizing autonomous system.

- Entity Certificate Issuer Autonomous System: (4 octets), the autonomous system of the entity that provided the Entitycert to the Authorizing AS. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

- Entity Certificate Serial Number: (4 octets), the Entitycert serial number containing the public key of the Authorizing AS.

- Signature: The signature itself.

The signature is calculated using the private key of the authorizing entity across all the TLVs within the Authcert. The Signature TLV MUST be appended as the last TLV in the Authcert after the signature has been computed.

4.2 Creation

An Authcert is usually created by the authorizing autonomous system with the following steps:

- Allocate a prefix block to the receiving autonomous system.
- Build an Authcert by adding TLVs containing its own AS number, the receiving (authorized) AS number, the prefix block, a unique sequence number, and any other information (e.g., URL pointing to the Entitycert that signed this Authcert.).
- Sign the Authcert by hashing and encrypting the Authcert TLVs. Place the signature (and other required) information in a Signature TLV, and append it to the Authcert.

4.2.1 Certificate Uniqueness

Digital certificates are created as uniquely named objects, which allows them to be uniquely identified. An Authcert is uniquely identified by the pair of Authorized Originator and Serial Number TLV values.

4.2.2 Certificate Encoding

Authcerts distributed in [SOBGP-BGP] are distributed in TLV form. However if they are manually distributed (e.g., through electronic mail) they may need to be base64 encoded into ASCII as described in Section 4.3 of [RFC1421].

4.3 Distribution

Authcerts are distributed as part of a Prefix Policy Certificate, so that an autonomous system can reliably match distribution policy to the prefix block.

4.4 Validation

The Authcert is validated using the following steps.

- Identify the Entitycert that signed the Authcert. The correct Entitycert is uniquely identified with the Entity Certificate Issuer Autonomous System and Entity Certificate Serial Number contained in the Signature TLV. The Entity Certificate Issuer Autonomous System is compared with the AS number in the Entitycert IssuerAltName field. The Entity Certificate Serial Number is compared with the Entitycert CertificateSerialNumber.
- Obtain the Entitycert that signed the Authcert, and validate it. The Entitycert may be in a local cache (already received via BGP extensions), retrieved using the URL in the Authcert, or through other means. If an entity does not have the validating public key it MUST NOT assume the Authcert is valid.
- Verify that the autonomous system identifier in SubjectAltname matches the Authorized Originator TLV value of the Authcert.
- If an Authorization Certificate Validity List is available, validate that the issuer of the Entitycert has not invalidated the Authcert. Validity lists may be distributed in the signers ASPolicycert, or a pointer to the list may be distributed in the Authcert in an Authorizing AS Validation List URL. If no Authorization Certificate Validity List is available, an entity MAY accept the certificate. However if a validation list is received later, the entity MUST check the validity of all certificates that had been previously accepted.
- Hash the Authcert TLVs.

- Extract the signature from the Authcert.
- Extract the public key from the Entitycert, and use it to decrypt the signature.
- Accept the Authcert as valid if the computed hash matches the decrypted hash. If the hashes do not match, the Authcert MUST be discarded.

4.4.1 Self-generated Authcerts

Self-generated Authcerts are dangerous, because a responsible third party does not assign the authorization. Trusting an autonomous

system to declare its own address space nullifies most of the protections outlined in this document.

However, the autonomous systems at the highest level of allocation (e.g. Regional Internet Registries (RIRs) or Local Internet Registries (LIRs)) may not be able to find a responsible third party to sign their Authcerts. In this case, self-generated Authcerts may be unavoidable.

Authcerts MAY be self-generated, but MUST only be accepted from autonomous systems that have been explicitly authorized and locally configured. For example, a device may be configured to accept Authcerts for the RIR autonomous systems.

4.5 Revocation

An entity issuing an Authcert MUST keep an Authcert revocation list. The entity MAY use any form for propagating that revocation list.

Because BGP routers do not necessarily have synchronized clocks, Authcerts do not carry expiration times, and thus do not expire. Revocation is only method of invalidating an Authcert.

Revocation information may be represented as a "validation list". A validation list includes lists of both valid and invalid (i.e., revoked) certificates. Any number not appearing in the list MUST be considered invalid. Validation list may be more efficient than a pure revocation list for Authcerts in the case where a large number of serial numbers have been revoked by an issuer.

An autonomous system SHOULD include an Authcert validation list in their AS Policy Certificate (distributed using [SOBGP-BGP]). This allows autonomous systems that cannot route to the issuing autonomous system to verify that the Entitycert has not been revoked.

5.0 Prefix Policy Certificates (PrefixPolicycert)

The PrefixPolicycert carries policy information sourced from route originators. It provides a specific set of policy regarding one or more prefix blocks. The owner of the prefix block creates it. There is only one valid PrefixPolicycert for each prefix block at any given time.

PrefixPolicycerts are verified with a digital signature provided by the autonomous system generating the policy. It does not contain a public key. Rather, it binds a particular policy to a particular identity (i.e., autonomous system).

5.1 Format

This certificate is formatted as a series of TLVs. Each TLV will include a type, which is treated as a 16 bit (two octet) unsigned integer, a length, which is also two octets, and a variable length data field. TLVs MUST be placed in the PrefixPolicycert in type order.

5.1.1 PrefixPolicycert Header

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +---------------+---------------+-------------------------------+
   | Cert. Marker  |    Type Id    | Length                        |
   +---------------+---------------+-------------------------------+
   | TLVs
   +----------------
```

o   Certificate Marker: "162(0xa2), identifying this as an soBGP certificate.

o   Type ID: "2(0x02), identifying this as an PrefixPolicycert.

o   Length: Set to the length of the TLVs.

o   TLVs: The Type/Length/Value attributes making up an PrefixPolicycert.

5.1.2 The Originating Autonomous System

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +--------------------------------+-------------------------------+
    | TLV Type                       | Length                        |
    +--------------------------------+-------------------------------+
    | Originating Autonomous System                                  |
    +----------------------------------------------------------------+
```

- TLV type: 1 (0x0001)

- Length: Set to 4.

- Originating Autonomous System: (4 octets), the autonomous system which originated this certificate. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

5.1.3 The Serial Number

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +--------------------------------+-------------------------------+
    | TLV Type                       | Length                        |
    +--------------------------------+-------------------------------+
    | Serial Number                                                  |
    +----------------------------------------------------------------+
```

- TLV type: 2 (0x0002)

- Length: Set to 4.

- Serial Number: (4 octets), A serial number which identifies this PrefixPolicycert, taken from a 32 bit number space.

5.1.4 Authorizing AS Entitycert Uniform Resource Locator

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +--------------------------------+-------------------------------+
    | TLV Type                       | Length                        |
    +--------------------------------+-------------------------------+
    | URL                                                            |
    +-----------------
```

- TLV type: 3 (0x0003)

- Length: Denotes the length of the URL in octets.

- URL: A uniform resource locator indicating a location where the Authorizing AS?s Entitycert can be found.

An PrefixPolicycert may omit this TLV. However, an implementation is REQUIRED to correctly parse them if they are present. A receiving device MAY choose to ignore the URL TLV.

5.1.5 Authcert

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Authentication Certificate                                    |
   +----------------
```

- TLV type: 4 (0x0004)

- Length: Set to the length of the Authentication Certificate.

- Authentication Certificate containing a prefix block for which the PrefixPolicycert applies.

One or more Authcert TLVs MUST be included in the PrefixPolicycert.

5.1.6 Policies

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Options                       | SubTVs
   +-------------------------------+--------------
```

- TLV type: 5 (0x0005)

- Length: Set to the sum of the Options size (2) and the length of the SubTVs.

- Options: (2 octets), a bit field describing various policies which should be applied to the prefixes indicated.

- SubTVs: (variable length), zero or more fields, the length of which is determined by the type, as described below.

5.1.6.1 Option bits

The options bit field describes policies that should be applied to the address prefix described in the TLV. These options are:

- Bit 0: Path Check. If this bit is set, the receiver should not accept any prefix for which the path cannot be verified as described in the section Verifying the Path, below.

- Bit 1: Second Hop Check. If this bit is set, the receiver should not accept any prefix for which the second entry in the AS PATH cannot be verified as described in the section Verifying the Second Hop, below.

- Bits 2-15: Reserved for future use.

5.1.6.2 SubTVs

The Authcert Policy subTVs provide optional policy information for the block of addresses included in the Authcert indicated; each subTV is of a fixed length, as determined by its type.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TV Type                       | Data....
   +-------------------------------+------------------------
```

- TV Type: (2 octets), An unsigned integer indicating the type of subTV

  Types defined within this specification are:

  - Type 1: Must Include AS, 4 octets of data, an AS which must be included in the AS path of any prefix falling within this block of addresses.

  - Type 2: OR Include AS, 4 octets of data, at least one of the included OR Include AS' must be included in the AS path of any prefix falling within this block of addresses.

  - Type 3: Maximum Prefix Length, 1 octet of data, the maximum length of any prefix allowed within this block of prefixes.

5.1.7 Signature

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
| TLV Type                      | Length                        |
+-------------------------------+-------------------------------+
| Signature Type                | Number of Issuers             |
+-------------------------------+-------------------------------+
| Entity Certificate Issuer Autonomous System                   |
+-------------------------------+-------------------------------+
| Entity Certificate Serial Number                              |
+-------------------------------+-------------------------------+
| ...                                                           |
+---------------------------------------------------------------+
| Signature                                                     |
+------------------
```

o   TLV type: 65535 (0xFFFF)

o   Length: (2 octets), unsigned integer denoting the length of the payload bytes which follow.

o   Signature Type: (2 octets), unsigned integer denoting the type of signature (the algorithm used to build this signature). Each possible signing algorithm is assigned an integer from this field. Signature type 1 is defined as an RSA encryption of a SHA1 digest.

o   Number of Issuers (2 octets): The number of Entitycert references included in the signature payload. If more than one Entitycert reference follows, all Entitycerts MUST contain the same public key for the same authorizing autonomous system.

o   Entity Certificate Issuer Autonomous System: (4 octets), the autonomous system of the entity that provided the Entitycert to the AS issuing the PrefixPolicycert. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

o   Entity Certificate Serial Number: (4 octets), the Entitycert serial number containing the public key of the AS issuing the PrefixPolicycert.

o   Signature: The signature itself.

The signature is calculated using the private key of the authorizing entity across all the TLVs within the PrefixPolicycert. The Signature TLV MUST be appended as the last TLV in the PrefixPolicycert after the signature has been computed.

5.2 Creation

An PrefixPolicycert is created by an autonomous system for prefix blocks that it owns. An autonomous system creates it with the following steps:

- Build an PrefixPolicycert by adding TLVs containing its own AS number, a unique sequence number, policy related to one or more prefix blocks, and the Authcert or Authcerts defining the prefix blocks to which this policy applies.
- Sign the PrefixPolicycert by hashing and encrypting the PrefixPolicycert TLVs. Place the signature (and other required) information in a Signature TLV, and append it to the PrefixPolicycert.

5.2.1 Certificate Uniqueness

Digital certificates are created as uniquely named objects, which allows them to be uniquely identified. A PrefixPolicycert is uniquely identified by the pair of Authorized Originator and Serial Number TLV values.

5.2.2 Certificate Encoding

PrefixPolicycert distributed in [SOBGP-BGP] are distributed in TLV form. However if they are manually distributed (e.g., through electronic mail) they may need to be encoded into ASCII. PrefixPolicycert SHOULD be base64 encoded as described in Section 4.3 of [RFC1421].

5.3 Distribution

PrefixPolicycerts may be distributed using any number of methods, for example:

- maintained in a directory maintained by the issuing autonomous system,
- distributed via some out of band mechanism, or
- distributed within BGP using extensions defined in [SOBGP-BGP].

To ensure interoperability, an autonomous system SHOULD distribute its PrefixPolicycerts within BGP.

5.4 Validation

The Authcert included in the Authcert TLV MUST be validated as correct before the Policy TLV can be accepted. Thus, the Authcert should be extracted from the PrefixPolicycert and validated before the PrefixPolicycert is validated.

The PrefixPolicycert is validated using the following steps.

- Identify the Entitycert that signed the PrefixPolicycert. The correct Entitycert is uniquely identified with the Entity Certificate Issuer Autonomous System and Entity Certificate Serial Number contained in the Signature TLV. The Entity Certificate Issuer Autonomous System is compared with the AS number in the Entitycert IssuerAltName field. The Entity Certificate Serial Number is compared with the Entitycert CertificateSerialNumber.
- Obtain the Entitycert that signed the Authcert, and validate it. The Entitycert may be in a local cache (already received via BGP extensions), retrieved using the URL in the Authcert, or through other means. If an entity does not have the validating public key it MUST NOT assume the PrefixPolicycert is valid.
- Verify that the autonomous system identifier in SubjectAltname matches the Authorized Originator TLV value of the PrefixPolicycert.
- Hash the PrefixPolicycert TLVs.
- Extract the signature from the PrefixPolicycert.
- Extract the public key from the Entitycert, and use it to decrypt the signature.
- Validate that the computed hash matches the decrypted hash. If the hashes do not match, the PrefixPolicycert MUST be discarded.

Once a PrefixPolicycert has been validated, any PrefixPolicycert that matches the following criteria MUST be discarded:
- has a lower serial number from the same originating AS, and
- includes an Authcert with the same prefix block

5.5 Revocation

Any entity issuing an PrefixPolicycert MUST keep a revocation list. The entity MAY use any form for propagating that revocation list.

Because BGP routers do not necessarily have synchronized clocks, PrefixPolicycert do not carry expiration times, and thus do not expire. Revocation is only method of invalidating an PrefixPolicycert.

Revocation information may be represented as a "validation list". A validation list includes lists of both valid and invalid (i.e., revoked) certificates. Any number not appearing in the list MUST be considered invalid. Validation list may be more efficient than a pure revocation list for PrefixPolicycerts in the case where a large number of serial numbers have been revoked by an issuer.

An autonomous system SHOULD include an PrefixPolicycert validation list in their AS Policy Certificate (distributed using [SOBGP-BGP]). This allows autonomous systems that cannot route to the issuing autonomous system to verify that the Entitycert has not been revoked.

6.0 AS Policy Certificates (ASPolicycert)

The ASPolicycert provides a specific set of policy relating to an autonomous system. An administrative entity within the autonomous system creates it. There is only one valid ASPolicycert for each autonomous system at any given time.

ASPolicycerts are verified with a digital signature from the autonomous system generating the policy. It does not contain a public key. Rather, it binds a particular policy to a particular identity (i.e., autonomous system).

6.1 Format

This certificate is formatted as a series of TLVs. Each TLV will include a type, which is treated as a 16 bit (two octet) unsigned integer, a length, which is also two octets, and a variable length data field. TLVs MUST be placed in the ASPolicycert in type order.

6.1.1 ASPolicycert Header

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +---------------+---------------+-------------------------------+
    | Cert. Marker  |    Type Id    | Length                        |
    +---------------+---------------+-------------------------------+
    | TLVs
    +----------------
```

o   Certificate Marker: "162(0xa2), identifying this as an soBGP certificate.

- Type ID: "3(0x03), identifying this as an ASPolicycert.

- Length: Set to the length of the TLVs.

- TLVs: The Type/Length/Value attributes making up an ASPolicycert.

6.1.2 The Originating Autonomous System

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
| TLV Type                      | Length                        |
+-------------------------------+-------------------------------+
| Originating Autonomous System                                 |
+---------------------------------------------------------------+
```

- TLV type: 1 (0x0001)

- Length: Set to 4.

- Originating Autonomous System: (4 octets), the autonomous system which originated this certificate. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

6.1.3 The Serial Number

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
| TLV Type                      | Length                        |
+-------------------------------+-------------------------------+
| Serial Number                                                 |
+---------------------------------------------------------------+
```

- TLV type: 2 (0x0002)

- Length: Set to 4.

- Serial Number: (4 octets), A serial number which identifies this ASPolicycert, taken from a 32 bit number space.

6.1.4 Authorizing AS Entitycert Uniform Resource Locator

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-------------------------------+-------------------------------+
    | TLV Type                      | Length                        |
    +-------------------------------+-------------------------------+
    | URL                                                           |
    +----------------
```

- TLV type: 3 (0x0003)

- Length: Denotes the length of the URL in octets.

- URL: A uniform resource locator indicating a location where the Authorizing AS?s Entitycert can be found.

An PrefixPolicycert may omit this TLV. However, an implementation is REQUIRED to correctly parse them if they are present. A receiving device MAY choose to ignore the URL TLV.

6.1.5 Attached Transit Autonomous Systems

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-------------------------------+-------------------------------+
    | TLV Type                      | Length                        |
    +-------------------------------+---------------+---------------+
    | Address Family Identifier     |   RESERVED    | Subsequent AFI |
    +-------------------------------+---------------+---------------+
    | Autonomous Systems                                            |
    +-----------------
```

- TLV type: 4 (0x0004)

- Length: Set to 4 + 4 octets for each autonomous system in the list.

- Address Family Identifier: This field carries the identity a the Network Layer protocol. Presently defined values for this field are specified in RFC 1700 (see the Address Family Numbers section).

- RESERVED: Set to 0.

- Subsequent AFI: This field provides additional information about the type of the Network Layer protocol.

- Autonomous Systems: List of autonomous systems which are connected to the originating autonomous system through some form of peering arrangement and which may transit traffic from the origin AS. Each AS number takes four octets. AS number values containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

One or more Attached Transit AS TLVs may be included in the Policy Certificate. Each type 4 TLV indicates an AS which is connected to the AS which originates this ASPolicycert through a BGP peering relationship.

6.1.6 Attached Non-transit Autonomous Systems

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
| TLV Type                      | Length                        |
+-------------------------------+---------------+---------------+
| Address Family Identifier     |   RESERVED    | Subsequent AFI|
+-------------------------------+---------------+---------------+
| Autonomous Systems                                            |
+------------------
```

o   TLV type: 5 (0x0005)

o   Length: Set to 4 + 4 octets for each autonomous system in the list.

o   Address Family Identifier: This field carries the identity a the Network Layer protocol. Presently defined values for this field are specified in RFC 1700 (see the Address Family Numbers section).

o   RESERVED: Set to 0.

o   Subsequent AFI: This field provides additional information about the type of the Network Layer protocol.

o   Autonomous Systems: List of autonomous systems which are connected to the originating autonomous system through some form of peering arrangement and which may not transit traffic from the origin AS. Each AS number takes four octets. AS number values containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

One or more Attached Non-Transit AS TLVs may be included in the

ASPolicycert. Each type 5 TLV indicates an AS which is connected to the AS which originates this ASPolicycert through a BGP peering relationship.

6.1.7 Revoked Entity Certificate List

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Entity Certificate Revocation List
   +----------------
```

- TLV type: 6 (0x0006)

- Length: (2 octets), length of TLV data (the list of revoked Entity Certificates) in octets

- Entity Certificate Revocation List: A revocation list created by the autonomous system, which includes a list of revoked Entity Certificates issued by this autonomous system. The format of the revocation list MUST be as defined in [RFC3280].

A single Revoked Entity Certificate List TLV MAY be included in an ASPolicycert, or it may be omitted.

When an Entity Certificate Revocation List is received, all currently held Entitycerts from this issuer MUST be checked against the validity list. Entitycerts found to be invalid MUST be deleted.

6.1.8 Authorization Certificate Validity List

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Validity Ranges
   +----------------
```

- TLV type: 7 (0x0007)

- Length: (2 octets), length of TLV data (the list of revoked Authorization Certificates) in octets

- Validity Ranges: A list of validity subTVs defining which serial numbers are valid and invalid. Validity ranges are interpreted in order until a match is found. For more information on validity lists, see Section 4.5.

A single TLV of this type MAY be included in an ASPolicycert, or it may be omitted.

When an Authorization Certificate Validity List is received, all currently held Authcerts from this issuer MUST be checked against the validity list. Authcerts found to be invalid MUST be deleted.

6.1.8.1 Validity Ranges

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | subTV Type                    | Size of Range                 |
   +-------------------------------+-------------------------------+
   | Lowest Authorization Serial Number                            |
   +---------------------------------------------------------------+
```

- subTV type: (2 octets).

SubTV type	Value
VALID	0
INVALID	1

- Size of Range: (2 octets). Number of contiguous serial numbers defining a range.

- Lowest Authorization Serial Number (4 octets). The lowest value in the range.

6.1.9 Prefix Policy Certificate Validity List

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-------------------------------+-------------------------------+
   | TLV Type                      | Length                        |
   +-------------------------------+-------------------------------+
   | Validity Ranges
   +----------------
```

- TLV type: 8 (0x0008)

- o Length: (2 octets), length of TLV data (the list of revoked Authorization Certificates) in octets

- o Validity Ranges: A list of validity subTVs (as defined in the previous section) defining which PrefixPolicycert serial numbers are valid and invalid. Validity ranges are interpreted in order until a match is found.. For more information on validity lists, see Section 5.5.

A single TLV of this type MAY be included in an ASPolicycert, or it may be omitted.

When an Prefix Policy Validity List is received, all currently held PrefixPolicycerts from this issuer MUST be checked against the validity list. PrefixPolicycerts found to be invalid MUST be deleted.

6.1.10 Most Recent AS Policy Certificate Uniform Resource Locator

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
| TLV Type                      | Length                        |
+-------------------------------+-------------------------------+
| URL                                                           |
+----------------
```

- o TLV type: 9 (0x0009)

- o Length: Denotes the length of the URL in octets.

- o URL: A uniform resource locator indicating a location where the most recent AS Policy Certificate can be found. This is useful for a receiver to verify that they have the most recent AS Policy Certificate for an AS.

An PrefixPolicycert may omit this TLV. However, an implementation is REQUIRED to correctly parse them if they are present. A receiving device MAY choose to ignore the URL TLV.

6.1.11 Signature

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

```
   +--------------------------------+--------------------------------+
   | TLV Type                       | Length                         |
   +--------------------------------+--------------------------------+
   | Signature Type                 | Number of Issuers              |
   +--------------------------------+--------------------------------+
   | Entity Certificate Issuer Autonomous System                     |
   +--------------------------------+--------------------------------+
   | Entity Certificate Serial Number                                |
   +--------------------------------+--------------------------------+
   | ...                                                             |
   +-----------------------------------------------------------------+
   | Signature                                                       |
   +------------------
```

o   TLV type: 65535 (0xFFFF)

o   Length: (2 octets), unsigned integer denoting the length of the payload bytes which follow.

o   Signature Type: (2 octets), unsigned integer denoting the type of signature (the algorithm used to build this signature). Each possible signing algorithm is assigned an integer from this field. Signature type 1 is defined as an RSA encryption of a SHA1 digest.

o   Number of Issuers (2 octets): The number of Entitycert references included in the signature payload. If more than one Entitycert reference follows, all Entitycerts MUST contain the same public key for the same authorizing autonomous system.

o   Entity Certificate Issuer Autonomous System: (4 octets), the autonomous system of the entity that provided the Entitycert to the AS issuing the PrefixPolicycert. AS numbers containing only two octets should be placed in the least significant octets of this four-octet field (the two rightmost octets).

o   Entity Certificate Serial Number: (4 octets), the Entitycert serial number containing the public key of the AS issuing the PrefixPolicycert.

o   Signature: The signature itself.

The signature is calculated using the private key of the authorizing entity across all the TLVs within the ASPolicycert. The Signature TLV MUST be appended as the last TLV in the ASPolicycert after the signature has been computed.

6.2 Creation

An ASPolicycert is created by an autonomous system in order to relay its own policy. An autonomous system creates it with the following steps:

- Build an ASPolicycert by adding TLVs containing its own AS number, a unique sequence number, and policy related to the autonomous system.
- Sign the ASPolicycert by hashing and encrypting the ASPolicycert TLVs. Place the signature (and other required) information in a Signature TLV, and append it to the ASPolicycert.

6.2.1 Certificate Uniqueness

Digital certificates are created as uniquely named objects, which allows them to be uniquely identified. An ASPolicycert is uniquely identified by the pair of Authorized Originator and Serial Number TLV values.

6.2.2 Certificate Encoding

ASPolicycert distributed in [SOBGP-BGP] are distributed in TLV form. However if they are manually distributed (e.g., through electronic mail) they may need to be encoded into ASCII. ASPolicycert SHOULD be base64 encoded following Section 4.3 of [RFC1421].

6.3 Distribution

ASPolicycert may be distributed using any number of methods, for example:

- maintained in a directory maintained by the issuing autonomous system,
- distributed via some out of band mechanism, or
- distributed within BGP using extensions defined in [SOBGP-BGP].

To ensure interoperability, an autonomous system SHOULD distribute its ASPolicycert within BGP.

6.4 Validation

The ASPolicycert is validated using the following steps.

- Identify the Entitycert that signed the ASPolicycert. The correct Entitycert is uniquely identified with the Entity Certificate Issuer Autonomous System and Entity Certificate Serial Number contained in the Signature TLV. The Entity Certificate Issuer Autonomous System is compared with the AS number in the Entitycert IssuerAltName field. The Entity Certificate Serial Number is compared with the Entitycert CertificateSerialNumber.
- Obtain the Entitycert that signed the ASPolicycert, and validate it. The Entitycert may be in a local cache (already received via BGP extensions), retrieved using the URL in the Authcert, or through other means. If an entity does not have the validating public key it MUST NOT assume the ASPolicycert is valid.
- Verify that the autonomous system identifier in SubjectAltname matches the Authorized Originator TLV value of the ASPolicycert.
- Hash the ASPolicycert TLVs.
- Extract the signature from the ASPolicycert.
- Extract the public key from the Entitycert, and use it to decrypt the signature.
- Validate that the computed hash matches the decrypted hash. If the hashes do not match, the ASPolicycert MUST be discarded.

Once an ASPolicycert has been validated, any ASPolicycert with a lower serial number from the same originating AS MUST be discarded.

6.5 Revocation

Each ASPolicycert issued by an autonomous system overrides any previously issued ASPolicycerts from this autonomous system. Therefore, revocation is not required.

If present, a receiver has the opportunity of using the Most Recent AS Policy Certificate URL in the ASPolicycert to verify that they have the most recent policy certificate.

7.0 Security Considerations

This document describes the format of authentication, authorization, and policy certificates used to with [SOBGP-BGP]. Each certificate type is digitally signed, and therefore requires no external protection to ensure its integrity. There are no restrictions on how they may be distributed. Revocation schemes are defined for all certificate types.

The following sections describe the security considerations of each of those objects.

7.1 Entitycerts

Entitycerts provide authentication, providing a binding of an identity (i.e., autonomous system number) to a public key. The authenticity of the binding is verified with a digital signature, where the public key of the certificate issuer has been previously accepted as valid. Issuer public keys can either be manually configured, or are verified through the use of another issuer's trusted public key in a "web of trust" built by the receiver.

Certificate issuers MUST maintain certificate revocation lists (CRLs). Entities verifying Entitycerts SHOULD reference the certificate revocation lists whenever possible. (Mandating the consultation of a CRL as part of the verification process is not possible, because the CRL may not be available at the time verification is performed. For example, if the issuer maintains the CRL on a directory server to which routing is not yet setup.) Issuers SHOULD distribute their CRLs within their AS Policy Certificates to increase the likelihood of a receiver having the CRL available.

Self-signed Entitycerts may be necessary in order to start a chain of trust. However self-signed Entitycerts MUST be manually validated as accurate before the enclosed public key is used, else the "web of trust" breaks down.

7.2 Authcerts

Authcerts provide authorization, where the issuer of a prefix block certifies that it has given that prefix block to a specific autonomous system. Receivers use the Authcert to validate announcements received in BGP UPDATE messages.

The authenticity of Authcerts is verified with a digital signature, where the public key of the certificate issuer is distributed in an Entitycert. Before a receiver can verify the Authcert, they MUST first check that the verifying Entitycert is authentic.

The Authcert issuer MUST keep an Authcert validation list describing which certificates are valid, and which are invalid. The receivers of an Authcert SHOULD consult the Authcert validation list to ensure that the authorization has not been revoked.

Autonomous systems may need to authorize their own use of prefix blocks if the autonomous system that issued their prefix blocks does not issue them an Authcert. However, such self-generated Authcerts are dangerous, since unrestricted use of self-signed Authcerts defeats the goal of authorization. Thus an entity MUST accept self-generated Authcerts only from autonomous systems that have been explicitly configured as trusted to claim authorization without the confirmation of a third party.

7.3 PrefixPolicycerts

PrefixPolicycerts bind policy generated by an autonomous system for prefix blocks that they advertise. This policy is bound to a particular Authcert, which verifies that they are authorized to advertise those prefix blocks.

PrefixPolicycerts are verified with a digital signature, where the public key of the certificate issuer is distributed in an Entitycert. Before a receiver can verify the PrefixPolicycert, they MUST first verify that the verifying Entitycert is authentic.

7.4 ASPolicycerts

ASPolicycerts contain policy generated by an autonomous system, and contain policy about the autonomous system itself. The policy includes its neighbor autonomous systems, which can be used by other entities to validate valid inter-connections. The policy can also include revocation and validation lists (Authcert, PrefixPolicycert).

ASPolicycerts are verified with a digital signature, where the public key of the certificate issuer is distributed in an Entitycert. Before a receiver can verify the ASPolicycerts, they MUST first verify that the verifying Entitycert is authentic.

7.5 Entitycert Uniform Resource Locators

Authcerts, PrefixPolicycerts, and ASPolicycerts may contain a URL that references the Entitycert used to validate it. Care should be taken in evaluating the URL since it is not yet known to be valid and could be used to propagate a denial of service attack.

8.0 IANA Considerations

This document defines three certificate types, each of which contains a series of TLVs. IANA is expected to maintain a registry of all the values defined, according to the following sections.

8.1 Authorization Certificate

The Authorization Certificate Type Field:

- o Type values 1 through 4, 14 and 65535 are assigned in this document.

- o Type values 5 through 13 and 15 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.1.1 Signature Type

The Signature TLV Signature Type field:

- o Type values 1 is assigned in this document.

- o Type values 2 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.2 Prefix Policy Certificate

- o Type values 1 through 5, 14 and 65535 are assigned in this document.

- o Type values 6 through 13 and 15 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.2.1 Policies Type

The Policies Type has two name spaces: Options flags and SubTVs.

The Options Field:

- o Bits 0 and 1 are assigned in this document.

- o Bits 2 thru 7 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Bits 8 thru 15 are for "Private Use" as defined in RFC 2434 [RFC2434].

The subTV TV Type field:

- o TV Type values 1 through 3 are assigned in this document.

- o TV Type values 4 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o TV Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o TV Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.2.2 Signature Type

The Signature TLV Signature Type field:

- o Type values 1 is assigned in this document.

- o Type values 2 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.3 AS Policy Certificate

- o Type values 1 through 9, 14 and 65535 are assigned in this document.

- o Type values 10 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.3.1 Validity Ranges

- o Type values 1 through 2 are assigned in this document.

- o Type values 3 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

8.3.2 Signature Type

The Signature TLV Signature Type field:

- o Type values 1 is assigned in this document.

- o Type values 2 through 16575 MUST be assigned using the "IETF Consensus" policy defined in RFC 2434 [RFC2434].

- o Type values 16576 through 32895 SHOULD be assigned using the "Specification Required" policy defined in RFC 2434 [RFC2434].

- o Type values 32896 through 65534 are for "Private Use" as defined in RFC 2434 [RFC2434].

9.0 Acknowledgments

A large number of people contributed to or provided valuable feedback on this document; we've tried to include all of them here (in no particular order), but might have missed a few: James Ng, Russ White, Alvaro Retana, Dave Cook, John Scudder, David Ward, Martin Djernaes, Max Pritikin, Chris Lonvick, Tim Gage, Scott Fanning, Barry Friedman, Jim Duncan, Yi Yang, Robert Adams, Tony Tauber, Iljitsch van Beijnum, Ed Lewis, and Jonathan Natale.

10.0 References

10.1 Normative References

[ADDR-EXT] Lynn, C., Kent, S. and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", draft-ietf-pkix-x509-ipaddr-as-extn-03.txt, September 2003.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Level", BCP 14, RFC 2119, March 1997.

[RFC2434] Narten, T., and H. Alvestrand,, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, October 1998.

[RFC2858] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 2858, June 2000.

[RFC3279] Polk, T., et. al., " Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[RFC3280] Housley, R., et. al., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 3280, April 2002.

[SOBGP-BGP] Ng, J. (editor), "Extensions to BGP to Support Secure Origin BGP (soBGP)", draft-ng-sobgp-extensions-01.txt, June 2003.

[SOBGP-DEPLOY] White, R. (editor), ?Deployment Considerations for Secure Origin BGP (soBGP)?, draft-white-sobgp-bgp-deployment-01.txt, June 2003.

[SOBGP-RADIUS] Lonvick, C., ?RADIUS Attributes for soBGP Support?, draft-lonvick-sobgp-radius-03.txt, August 19, 2003.

[X.690] International Telecommunication Union, "ITU-T Recommendation X.660 Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 1997.

10.2 Informative References

[IAB-SC] Rescorla, E., B. Korver, and the Internet Architecture Board, "Guidelines for Writing RFC Text on Security Considerations", http://www.ietf.org/internet-drafts/draft-iab-sec-cons-03.txt, Work in progress, 2003.

[RFC3281] Farrell, S., and R. Housley, " An Internet Attribute Certificate Profile for Authorization", RFC 3281, April 2002.

Editor's Address

Brian Weis
Cisco Systems
170 W. Tasman Drive,
San Jose, CA 95134-1706, USA
(408) 526-4796
bew@cisco.com