Internet Engineering Task Force
Internet Draft
draft-ietf-pkix-x509-ipaddr-as-extn-00.txt
Expires August 2002

Charles Lynn
Stephen Kent
Karen Seo
BBN Technologies
February 2002

X.509 Extensions for IP Addresses and AS Identifiers

Status of this Memo

Abstract

This document defines two private X.509 v3 certificate extensions. The first binds a list of IP address blocks, or prefixes, to the subject of a certificate. The second binds a list of Autonomous System Identifiers to the subject of a certificate. These extensions may be used to convey the authorization of the subject to use the IP addresses and Autonomous System identifiers contained in the extensions.

Please send comments on this draft to the ietf-pkix@imc.org mail list.

Table of Contents

1. Introduction

This document defines two private X.509 v3 certificate extensions. The first binds a list of IP address blocks, or prefixes, to the subject (private key holder) of a certificate. The second binds a list of Autonomous System (AS) Identifiers to the subject of a certificate. These extensions convey that the subject "owns" or is authorized to use the IP address blocks and AS Identifiers contained in the extensions. The issuer of the certificate would typically be the entity (e.g., IANA, a regional registry, ISP) who owns the set of IP address blocks and AS Identifiers from which the subject's IP address blocks or AS Identifiers have been taken and who made the delegation of the resources to the subject. These certificates provide a scalable means of verifying the ownership of IP address prefixes and AS Identifiers, e.g., they can be used by routing protocols such as Secure BGP [S-BGP] to verify legitimacy/correctness of routing information.

It is assumed that the reader is familiar with the terms and concepts described in [PKIX-1], [RFC2459] (PKIX X.509 certificate profile), [RFC791] (IPv4), [RFC2373] (IPv6). Some relevant terms include:

advertise - (see [RFC1771]).

delegate - Transfer ownership of an IP address block or AS identifier through issuance of a certificate to the new owner.

downstream service provider (DSP) - Second or lower tier internet service provider.

initial octet - the first octet in the value of a DER encoded BIT STRING [X.690].

IP v4 address (IPv4) - a 32-bit identifier written as four decimal numbers, each in the range 0 to 255, separated by "."s. 10.5.0.5 is an example.

IP v6 address (IPv6) - a 128-bit identifier written as eight hexadecimal quantities, each in the range 0 to ffff, separated by ":"s. 2001:0:2:3:0:0:0:1 is an example. One string of :0: quantities may be replaced by "::", thus 2001:0:2:3::1 represents the same address as the immediately preceding example. (See [RFC2373]).

own - for an IP address prefix, being authorized to specify the AS that may originate advertisement of the prefix throughout the Internet. For an Autonomous System Identifier, being authorized to operate a network(s) that identifies itself to other network operators using that Autonomous System Identifier. Or, for either, being authorized to delegate ownership to another entity.

subsequent octets - the second through last octets in the value of a DER encoded BIT STRING [X.690].

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. IP Address Delegation Extension

This extension conveys the delegation of ownership of IP addresses to the subject by binding those addresses to a public key belonging to the subject.

2.1. Context

IP address space is currently managed by a hierarchy nominally rooted at ICANN, but managed by Internet Regional Registries (e.g., APNIC, ARIN, and RIPE). ICANN delegates IP address space to the Registries,

who in turn delegate IP address space to internet service providers (ISPs), who delegate IP address space to down stream providers (DSPs), etc. Any level in the hierarchy can also delegate IP address space to organizations who are end entities, i.e., organizations who will not be re-delegating any of their space to other organizations. The IP address delegation extension is intended to enable verification of this ownership of IP address blocks, i.e., of the authorization of an entity to use or delegate IP address space. Accordingly, it makes sense to take advantage of the inherent authoritativeness of the existing hierarchy for delegating IP address space. Thus the PKI hierarchy for issuing certificates with this extension SHOULD parallel the IP address delegation hierarchy. The roots of the PKI hierarchy will be the regional Internet Registries (i.e., APNIC, ARIN, RIPE, etc.), the next level down will be the ISPs, etc. An example of one use of this extension is a router using it to verify the authorization of an organization to originate a BGP UPDATE advertising a path to a particular IP address block [S-BGP].

2.2. Specification

2.2.1. OID

The OID for this extension is id-pe-ipAddrBlock.

```
id-pe-ipAddrBlock  OBJECT IDENTIFIER ::= { id-pe 7 }
```

where [RFC2459] defines


```
id-pkix  OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
           dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

id-pe    OBJECT IDENTIFIER ::= { id-pkix 1 }
```

2.2.2. Criticality

This extension can be CRITICAL or NOT CRITICAL at the discretion of the CA issuing the certificate. The intended use of this extension is to connote ownership of the block(s) of IP addresses identified in

the extension. A CA might well mark the extension as CRITICAL to convey the notion that a relying party must understand the semantics of the extension to make use of the certificate. Newly created applications that would make use of certificates containing this extension would be expected to recognize the extension. However, many common application implementations (e.g., browsers) that might make use of certificates that contain this extension, (as clients not as replying parties) do not tolerate CRITICAL private extensions, and thus a CA may choose to not mark this extension as CRITICAL, to avoid compatibility problems with these application implementations.

2.2.3. Syntax

```
id-pe-ipAddrBlock OBJECT IDENTIFIER ::= { id-pe 7 }

IPAddrBlocks        ::= SEQUENCE OF IPAddressFamily

IPAddressFamily     ::= SEQUENCE {
   addressFamily        OCTET STRING (SIZE (2..3)), -- AFI & opt SAFI
   ipAddressChoice      IPAddressChoice }

IPAddressChoice     ::= CHOICE {
   inherit              BOOLEAN, -- Inherit from Issuer
   addressesOrRanges    SEQUENCE OF IPAddressOrRange }

IPAddressOrRange    ::= CHOICE {
   addressPrefix        IPAddress,
   addressRange         IPAddressRange }

IPAddressRange      ::= SEQUENCE {
   min                  IPAddress,
   max                  IPAddress }

IPAddress           ::= BIT STRING
```


2.2.3.1. Type IPAddrBlocks

The IPAddrBlocks type is a sequence of IPAddressFamily types.

2.2.3.2. Type IPAddressFamily

The IPAddressFamily type is a sequence containing an addressFamily and ipAddressChoice element.

2.2.3.3. Element addressFamily

The addressFamily element is an OCTET STRING containing a two-octet Address Family Identifier (AFI), in network byte order, optionally followed by a one-octet Subsequent Address Family Identifier (SAFI). AFI's and SAFI's are specified in [IANA] and [RFC2283], respectively.

There MUST be only one IPAddressFamily sequence per unique combination of AFI and SAFI. Each sequence MUST be ordered by ascending addressFamily values (treating the octets as unsigned quantities). An addressFamily without a SAFI MUST precede one that contains a SAFI. When both IPv4 and IPv6 addresses are specified, the IPv4 addresses MUST precede the IPv6 addresses (since the IPv4 AFI of 0001 is less than the IPv6 AFI of 0002).

2.2.3.4. Element ipAddressChoice and Type IPAddressChoice

The ipAddressChoice element is of type IPAddressChoice. The IPAddressChoice type is a CHOICE of either an inherit or addressesOrRanges element.

2.2.3.5. Element inherit

If the IPAddressChoice choice contains the inherit element, then the BOOLEAN MUST be TRUE. In this case, the set of authorized IP addresses for the specified AFI and optional SAFI is taken from the Issuer's certificate, or the Issuer's Issuer's certificate, recursively, until a certificate containing an IPAddressChoice containing an addressesOrRanges element is located. If no authorization is being granted for a particular AFI and optional SAFI, then there SHOULD NOT be an IPAddressFamily member for that AFI/SAFI in the IPAddrBlocks sequence; i.e., the AFI/SAFI should be omitted rather than setting inherit BOOLEAN to FALSE.

2.2.3.6. Element addressesOrRanges

The addressesOrRanges element is a sequence of IPAddressOrRange types. The addressPrefixes and addressRange elements MUST be sorted using the representation IP address/prefix length. Note that the bytes in this representation (a.b.c.d/length for IPv4 or s:t:u:v:w:x:y:z/length for IPv6) are not in the same order as occurs in a DER encoded BIT STRING. For example, given two addressPrefixes:

IP addr/length	DER encoding
10.32.0.0/12	03 03 04 0a 20
10.64.0.0/16	03 03 00 0a 40

the prefix 10.32.0.0/12 MUST come before the prefix 10.64.0.0/16 since 32 is less than 64; whereas if one were to sort by the DER BIT STRINGs, the order would be reversed as the unused bits octet would sort in the opposite order. Any pair of IPAddressOrRange choices in an extension MUST NOT overlap each other. Any contiguous address prefixes or ranges MUST be combined into a single range or, when possible, a single prefix.

2.2.3.7. Type IPAddressOrRange

The IPAddressOrRange type is a CHOICE of either an addressPrefix (an IP address Prefix) or an addressRange (an IP address range) element.

2.2.3.8. Element addressPrefix and Type IPAddress

The addressPrefix element is an IPAddress type. The IPAddress type defines a range of IP addresses in which the most significant (left-most) N bits of the address remain constant while the remaining bits (32 - N for IPv4, or 128 - N for IPv6) may be either zero or one. A prefix is written as the constant octets followed by a "/" and the number of constant bits (N). For example, the IPv4 prefix 10.64/12 corresponds to the addresses 10.64.0.0 to 10.79.255.255 while 10.64/11 corresponds to 10.64.0.0 to 10.95.255.255. The IPv6 prefix 2001:0:2/48 represents addresses 2001:0:2:: to 2001:0:2:ffff:ffff:ffff:ffff:ffff.

An IP address prefix is encoded as a BIT STRING. The DER encoding of

a BIT STRING uses the initial octet of the string to specify how many of the least significant bits of the last subsequent octet are unused. DER encoding specifies that these unused bits MUST be set to zero. The special case of all IP address blocks, i.e., a prefix of all zero bits -- "0/0", MUST be encoded per DER with a length octet of one, an initial octet of zero, and no subsequent octets -- 0x03, 0x01, 0x00. Note that the number of trailing zero bits is significant for IP addresses. For example, the DER encoding of

10.64/12, 0x03, 0x03, 0x04, 0x0a, 0x40, is different than 10.64/11, encoded as 0x03, 0x03, 0x05, 0x0a, 0x40.

2.2.3.9. Element addressRange and Type IPAddressRange

The addressRange element is of type IPAddressRange. The IPAddressRange type consists of a SEQUENCE containing a minimum (element min) and maximum (element max) IP address. Each IP address is encoded as a BIT STRING. The semantic interpretation of the minimum address in an IPAddressRange is that all the unspecified bits (for the full length of the IP address) are zero-bits (0). The semantic interpretation of the maximum address is that all the unspecified bits are one-bits (1).

Note that an IP address prefix can be encoded as a range, where the minimum and maximum values would be identical. However, a range of IP addresses MUST, whenever possible, be encoded as a single prefix and NOT be encoded as a range.

1) Address ranges (bit strings) should be sorted into ascending order by most-significant address bits

2) Contiguous prefixes and/or ranges MUST be combined into a single prefix (whenever possible) or range.

   Let "LMBx" denote the "Left Most Bits of x".

3) If a range is of the form minimum IP address = <n LMBp><zeros>
   and maximum IP address = <n LMBp><ones>,
   where n >= 0, then the prefix form MUST be used:
   BIT STRING ((8 - (n mod 8)) mod 8) <n LMBp><zero pad last byte>
   else the min/max form MUST be used.

Example:
    128.0.0.0   = 1000 0000.0000 0000.0000 0000.0000 0000
 to 143.255 255 255 = 1000 1111.1111 1111.1111 1111.1111 1111
BIT STRING 4 128   -- 1000

4) A min/max form with minimum IP address = <(i - 1) LMBn><1><zeros>
                  and maximum IP address = <(j - 1) LMBx><0><ones>
MUST be encoded as:
SEQUENCE {
   BIT STRING ((8 - (i mod 8)) mod 8) <i LMBn><zero pad last byte>
   BIT STRING ((8 - (j mod 8)) mod 8) <j LMBx><zero pad last byte>
}

I.e., all trailing zero bits are removed from the min and all trailing 1 bits are removed from the max.

Example:

        129.64.0.0   = 1000 0001.0100 0000.0000 0000.0000 0000
     to 143.255.255.255 = 1000 1111.1111 1111.1111 1111.1111 1111
SEQUENCE {
   BIT STRING 6 129 64 -- 1000 0001.01
   BIT STRING 4 128   -- 1000
}

To simplify the comparison of IP address blocks when performing certificate path validation, a maximum IP address MUST contain at least one bit whose value is 1, i.e., the subsequent octets may neither be omitted nor all zero.

NOTE: this specification could require that the least significant bit in the encoding of the max BIT STRING be a 1. This would insure that a broken ASN.1 DER encoder that removes all trailing zero bits, when DER encoding a BIT STRING, does not silently change the semantics of the max element.

SHOULD THE SPECIFICATION REQUIRE THIS DEFENSIVE ACTION?

2.3. IP Address Delegation Extension Certification Path Validation

Certification path validation of a certificate containing the IP address delegation extension requires additional processing. As each

certificate in a path is validated, the IP addresses in the IP address delegation extension of that certificate must be subsumed by IP addresses in the IP address delegation extension in the issuer's certificate.

3. Autonomous System Identifier Delegation Extension

This extension conveys the delegation of ownership of Autonomous System (AS) identifiers to the subject by binding those AS identifiers to a public key belonging to the subject.

3.1. Context

AS identifier delegation is currently managed by a hierarchy with roots at ICANN and the Internet Registries (APNIC, ARIN, RIPE, etc.). ICANN delegates AS identifiers to the Registries, who in turn delegate AS identifiers to organizations who are end entities, i.e., will not be re-delegating any of their identifiers to other organizations. The AS identifier delegation extension is intended to enable verification of this ownership of AS identifiers, i.e., of the authorization of an entity to use these AS identifiers. Accordingly, it makes sense to take advantage of the inherent authoritativeness of the existing hierarchy for delegating AS identifiers. Thus the PKI hierarchy for issuing certificates with this extension SHOULD

parallel the AS identifier delegation hierarchy. The roots of the PKI hierarchy will be the regional Internet Registries (i.e., APNIC, ARIN, RIPE, etc.). An example of one use of this extension is a router using it to verify the authorization of an organization to prepend an AS Number to the AS_PATH attribute of a BGP UPDATE [S-BGP].

3.2. Specification

3.2.1. OID

The OID for this extension is id-pe-autonomousSysId.

```
id-pe-autonomousSysId  OBJECT IDENTIFIER ::= { id-pe 8 }
```

where [RFC2459] defines

```
id-pkix  OBJECT IDENTIFIER ::= { iso(1) identified-organization(3)
           dod(6) internet(1) security(5) mechanisms(5) pkix(7) }

id-pe    OBJECT IDENTIFIER ::= { id-pkix 1 }
```

3.2.2. Criticality

This extension can be CRITICAL or NOT CRITICAL at the discretion of the CA issuing the certificate. The intended use of this extension is to connote ownership of the AS identifiers in the extension. A CA might well mark the extension as CRITICAL to convey the notion that a relying party must understand the semantics of the extension to make use of the certificate. Newly created applications that would make use of certificates containing this extension would be expected to recognize the extension. However, many common application implementations (e.g., browsers) that might make use of certificates that contain this extension, (as clients not as replying parties) do not tolerate CRITICAL private extensions, and thus a CA may choose to not mark this extension as CRITICAL, to avoid compatibility problems with these application implementations.

3.2.3. Syntax


```
id-pe -autonomousSysId  OBJECT IDENTIFIER ::= { id-pe 8 }

ASIdentifiers      ::= SEQUENCE {
    asnum               [0] EXPLICIT ASIdentifierChoice OPTIONAL,
    rdi                 [1] EXPLICIT ASIdentifierChoice OPTIONAL}

ASIdentifierChoice ::= CHOICE {
```

```
      inherit              BOOLEAN, -- Inherit from Issuer
      asIdsOrRanges        SEQUENCE OF ASIdOrRange }

   ASIdOrRange         ::= CHOICE {
       id                   ASId,
       range                ASRange }

   ASRange             ::= SEQUENCE {
       min                  ASId,
       max                  ASId }

   ASId                ::= INTEGER
```

3.2.3.1. Type ASIdentifiers

The ASIdentifiers type is a SEQUENCE containing one or more forms of Autonomous System identifiers -- AS numbers (in the asnum element) or Routing Domain Identifiers (in the rdi element). When the ASIdentifiers type contains multiple forms of identifiers, the asnum entry will precede the rdi entry. AS numbers are used by BGP and Routing Domain Identifiers are specified in the IDRP.

3.2.3.2. Elements asnum, rdi, and Type ASIdentifierChoice

The asnum and rdi elements are both of type ASIdentifierChoice. The ASIdentifierChoice type is a CHOICE of either the inherit or asIdsOrRanges element.

3.2.3.3. Element inherit

If the ASIdentifierChoice choice contains the inherit element, then the BOOLEAN MUST be TRUE. In this case, the set of authorized AS identifiers is taken from the Issuer's certificate, or the Issuer's Issuer's certificate, recursively, until a certificate containing an ASIdentifierChoice containing an sasIdsOrRanges element is located. If no authorization is being granted for a particular form of AS identifier then there SHOULD NOT be an asnum/rdi member in the ASIdentifiers sequence; i.e., the member should be omitted rather than setting inherit BOOLEAN to FALSE.

3.2.3.4. Element asIdsOrRanges

The asIdsOrRanges element is a SEQUENCE of ASIdOrRange types. Any pair of items in the asIdsOrRanges SEQUENCE MUST NOT overlap.

3.2.3.5. Type ASIdOrRange

The ASIdOrRange type is a CHOICE of either a single integer (ASId) or a single sequence (ASRange).

3.2.3.6. Element id

The id element has type ASId.

3.2.3.7. Element range

The range element has type ASRange.

3.2.3.8. Type ASRange

The ASRange type is a SEQUENCE of a min and a max element and is used to specify a range of AS identifier values.

3.2.3.9. Elements min and max

The min and max elements have type ASId. The min element is used to specify the value of the minimum AS identifier in the range and the max elements specifies the value of the maximum AS identifier in the range.

3.2.3.10. Type ASId

The ASId type is an INTEGER.

3.3. Autonomous System Identifier Delegation Extension Certification Path Validation

Certification path validation of a certificate containing the Autonomous System identifier delegation extension requires additional processing. As each certificate in a path is validated, the AS identifiers in the Autonomous System identifier delegation extension of that certificate must be subsumed by the AS identifiers in the

Autonomous System identifier delegation extension in the issuer's certificate.

4. Security Considerations

This specification describes two private X.509 extensions. Since X.509 certificates are digitally signed, no additional integrity service is necessary. Certificates with these extensions need not be kept secret, and unrestricted and anonymous access to these certificates has no security implications.

However, security factors outside the scope of this specification will affect the assurance provided to certificate users. This section highlights critical issues that should be considered by implementors, administrators, and users.

These extensions represent authorization information, i.e., ownership of IP addresses and/or AS identifiers. They were developed to support a secure version of BGP, but may be employed in other contexts. In the secure BGP context, certificates containing these extensions function as capabilities, i.e., the certificate asserts that the holder of the private key (the Subject) owns the IP addresses and/or AS identifiers represented in the extension(s). As a result of this capability model, the Subject field is largely irrelevant for security purposes, contrary to common PKI conventions.

5. Acknowledgements

The authors would like to acknowledge the contributions to this specification by Charles Gardiner and Russ Housley.

Providing feedback could get your name here!

Appendix A -- Examples of IP Address Delegation Extensions

A non-critical X.509 v3 certificate extension that specifies:
IPv4 unicast address prefixes

1) 10.0.32/20 i.e., 10.0.32.0 to 10.0.47.255
2) 10.0.64/24 i.e., 10.0.64.0 to 10.0.64.255
3) 10.1/16 i.e., 10.1.0.0 to 10.1.255.255

4) 10.2.48/20 i.e., 10.2.48.0 to 10.2.63.255
5) 10.2.64/24 i.e., 10.2.64.0 to 10.2.64.255
6) 10.3/16 i.e., 10.3.0.0 to 10.3.255.255

and

7) inherits all IPv6 addresses from the Issuer's certificate

would be (in hexadecimal):


```
30 44                        Extension {
   06 08 2b06010505070107      extnID        1.3.6.1.5.5.7.1.7
                               critical      FALSE (thus omitted)
   04 38                       extnValue {
      30 36                      IPAddrBlocks {
         30 2b                     IPAddressFamily {
            04 03 0001  01           addressFamily: IPv4 Unicast
                                     IPAddressChoice {
            30 24                      addressesOrRanges {
                                         IPAddressOrRange {
               03 04 04 0a0020             addressPrefix 10.0.32/20
                                         } -- IPAddressOrRange
                                         IPAddressOrRange {
               03 04 00 0a0040             addressPrefix 10.0.64/24
                                         } -- IPAddressOrRange
                                         IPAddressOrRange {
               03 03 00 0a01               addressPrefix    10.1/16
                                         } -- IPAddressOrRange
                                         IPAddressOrRange {
               30 0c                       addressRange {
                  03 04 04 0a0230            min        10.2.48.0
                  03 04 00 0a0240            max        10.2.64.255
                                           } -- addressRange
                                         } -- IPAddressOrRange
                                         IPAddressOrRange {
               03 03 00 0a03               addressPrefix    10.3/16
                                         } -- IPAddressOrRange
                                       } -- addressesOrRanges
                                     } -- IPAddressChoice
                                   } -- IPAddressFamily
         30 07                     IPAddressFamily {
```

```
            04 02 0002              addressFamily: IPv6
                                    IPAddressChoice {
            01 01 ff                  inherit: TRUE from Issuer
                                    } -- IPAddressChoice
                                  } -- IPAddressFamily
                                } -- IPAddrBlocks
                              } -- extnValue
                            } -- Extension
```

This example illustrates how the prefixes and ranges are sorted.

+ Prefix 1 precedes prefix 2, even though the number of unused bits (4) in prefix 1 is larger than the number of unused bits (0) in prefix 2.

+ Prefix 2 precedes prefix 3 even though the number of octets (4) in the BIT STRING encoding of prefix 2 is larger than the number of octets (3) in the BIT STRING encoding of prefix 3.

+ Prefixes 4 and 5 are adjacent (representing the range of address

from 10.2.48.0 to 10.2.64.255), so MUST be combined into a range (since the range cannot be encoded by a single prefix).

+ Note that the six trailing zero bits in the max element of the range are significant to the semantic interpretation of the value (as all unused bits are interpreted to be 1's, not 0's). The four trailing zero bits in the min element are not significant and MUST be removed (thus the (4) unused bits in the encoding of the min element). (DER encoding requires that unused bits in the last subsequent octet be set to zero.)

+ The range formed by prefixes 4 and 5 precedes prefix 6 even though the SEQUENCE encoding for a range (30) is larger than the encoding for a BIT STRING (03) used to encode a prefix.

+ The IPv4 information precedes the IPv6 information since the address family identifier for IPv4 (0001) is less than the identifier for IPv6 (0002).

An extension specifying the IPv6 prefix 2001:0:2/48 and the IPv4 prefixes 10/8 and 172.16/12, and which inherits all IPv4 multicast addresses from the issuer's certificate would be:

```
   30 3b                          Extension {
      06 08 2b06010505070107        extnID        1.3.6.1.5.5.7.1.7
                                    critical      FALSE (thus omitted)
      04 2f                         extnValue {
         30 2d                        IPAddrBlocks {
            30 10                       IPAddressFamily {
               04 03 0001 01              addressFamily: IPv4 Unicast
                                          IPAddressChoice {
               30 09                        addressesOrRanges {
                                              IPAddressOrRange {
                  03 02 00 0a                   addressPrefix    10/8
                                              } -- IPAddressOrRange
                                              IPAddressOrRange {
```

```
                  03 03 04 b010              addressPrefix    172.16/12
                                           } -- IPAddressOrRange
                                         } -- addressesOrRanges
                                       } -- IPAddressChoice
                                     } -- IPAddressFamily
            30 08                    IPAddressFamily {
               04 03 0001 02           addressFamily: IPv4 Multicast
                                       IPAddressChoice {
               01 01 ff                  inherit: TRUE from Issuer
                                       } -- IPAddressChoice
                                     } -- IPAddressFamily
            30 0f                    IPAddressFamily {
               04 02 0002              addressFamily: IPv6
                                       IPAddressChoice {
               30 09                     addressesOrRanges {
                                           IPAddressOrRange {
                  03 07 00 200100000002      addressPrefix    2001:0:2/48
                                           } -- IPAddressOrRange
                                         } -- addressesOrRanges
                                       } -- IPAddressChoice
                                     } -- IPAddressFamily
                                   } -- IPAddrBlocks
                                 } -- extnValue
                               } -- Extension
```


Appendix B -- Example of an AS Identifier Delegation Extension

An extension that specifies AS Numbers 135, 3000 to 3999, and 5001, and which inherits all Routing Domain Identifiers from the issuers

certificate would be (in hexadecimal):

```
30 29                        Extension {
   06 08 2b06010505070108      extnID        1.3.6.1.5.5.7.1.8
                               critical      FALSE (thus omitted)
   04 1d                       extnValue {
      30 1b                      ASIdentifiers {
         a0 14                     asnum
                                     ASIdentifierChoice {
            30 12                      asIdsOrRanges {
                                         ASIdOrRange {
               02 02 0087                  ASId
                                         } -- ASIdOrRange
                                         ASIdOrRange {
               30 08                       ASRange {
                  02 02 0bb8               min
                  02 02 0f9f               max
                                           } -- ASRange
                                         } -- ASIdOrRange
                                         ASIdOrRange {
               02 02 1389                  ASId
                                         } -- ASIdOrRange
                                       } -- asIdsOrRanges
                                     } -- ASIdentifierChoice
                                   } -- asnum
         a1 03                     rdi {
                                     ASIdentifierChoice {
            01 01 ff                   inherit
                                     } -- ASIdentifierChoice
                                   } -- rdi
                                 } -- ASIdentifiers
                               } -- extnValue
                             } -- Extension
```

References

[IANA] IANA web page, http://www.iana.org, has assignments for several number spaces, including "Address Family Numbers".

[PKIX-1] R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", draft-ietf-pkix-new-part1-08.txt, July 2001.

[PKIX-ALG] Bassham, L., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Representation of Public Keys and Digital Signatures," draft-ietf-pkix-ipki-pkalgs-00.txt, July 14, 2000.

[RFC1700] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. (see also http://www.iana.org/iana/assignments.html)

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", RFC 2026, BCP 00009, October 1996.

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

[RFC2373] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[RFC2459] Housley, R., Ford, W., Polk, W., Solo, D., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

[S-BGP] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," IEEE JSAC Special Issue on Network Security, April 2000.

[X.509] ITU-T Recommendation X.509 (1997 E): "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", June 1997.

[X.690] ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, "Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

Disclaimer

The views and specification here are those of the authors and are not necessarily those of their employers. The authors and their employers specifically disclaim responsibility for any problems

arising from correct or incorrect implementation or use of this specification.

Authors' Address

Charles Lynn
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
USA

Phone: +1 (617) 873-3367
Email: CLynn@BBN.Com

Stephen Kent
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
USA

Phone: +1 (617) 873-3988
Email: Kent@BBN.Com

Karen Seo
BBN Technologies
10 Moulton St.
Cambridge, MA 02138
USA

Phone: +1 (617) 873-3152
Email: KSeo@BBN.Com

Intellectual Property Rights

pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurance of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any

copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.