

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 17, 2013

M. Lepinski, Ed.  
BBN  
July 16, 2012

BGPSEC Protocol Specification  
draft-ietf-sidr-bgpsec-protocol-04

## Abstract

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

## Requirements Language

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

## Status of this Memo

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

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

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

This Internet-Draft will expire on January 17, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. BGPSEC Negotiation . . . . .	3
3. The BGPSEC_Path_Signatures Attribute . . . . .	6
3.1. Secure_Path . . . . .	8
3.2. Additional_Info . . . . .	9
3.3. Signature_Block . . . . .	11
4. Generating a BGPSEC Update . . . . .	12
4.1. Originating a New BGPSEC Update . . . . .	13
4.2. Propagating a Route Advertisement . . . . .	16
4.3. Reconstructing the AS_Path Attribute . . . . .	19
4.4. Processing Instructions for Confederation Members . . . . .	19
5. Processing a Received BGPSEC Update . . . . .	21
5.1. Validation Algorithm . . . . .	22
6. Algorithms and Extensibility . . . . .	26
6.1. Algorithm Suite Considerations . . . . .	26
6.2. Extensibility Considerations . . . . .	27
7. Security Considerations . . . . .	28
8. Contributors . . . . .	31
8.1. Authors . . . . .	31
8.2. Acknowledgements . . . . .	32
9. Normative References . . . . .	32
Author's Address . . . . .	33

## 1. Introduction

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

1. The route was originated by an AS that has been explicitly authorized by the holder of the IP address prefix to originate route advertisements for that prefix.
2. Every AS listed in the AS\_Path attribute of the update explicitly authorized the advertisement of the route to the subsequent AS in the AS\_Path.

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

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

## 2. BGPSEC Negotiation

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

This capability has capability code : TBD

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

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

## Capability Value:

0	1	2	3	4	5	6	7
-----							
	Send		Receive		Reserved		Version
-----							
	AFI						
-----							
-----							
	Reserved						
-----							
	SAFI						
-----							

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

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

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

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

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

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

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

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

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

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

- o The BGP speaker understands the BGPSEC\_Path\_Signatures attribute (see Section 3).
- o The BGP speaker supports 4-byte AS numbers (see RFC 4893).

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

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

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

### 3. The BGPSEC\_Path\_Signatures Attribute

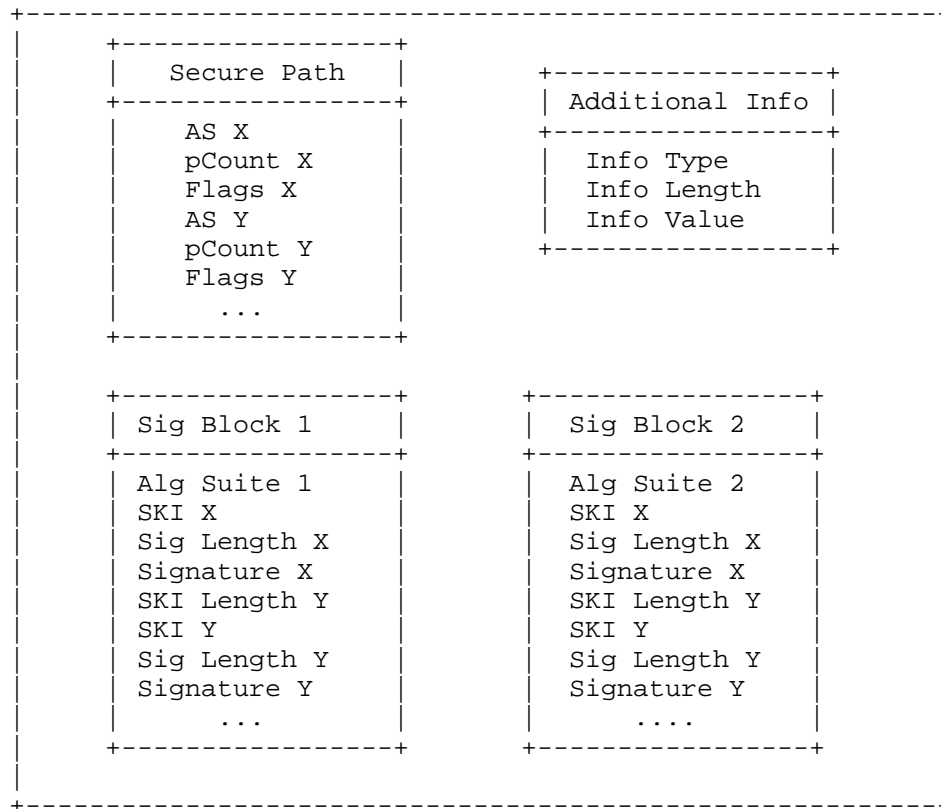
The BGPSEC\_Path\_Signatures attribute is a new optional (non-transitive) BGP path attribute.

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

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

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

High-Level Diagram of the BGPSEC\_Path\_Signatures Attribute  
BGPSEC\_Path\_Signatures



The following is a more detailed explanation of the format of the BGPSEC\_Path\_Signatures attribute.

BGPSEC\_Path\_Signatures Attribute

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

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

be contained in the AS\_PATH attribute. A BGPSEC update message containing the BGPSEC\_PATH\_SIGNATURES attribute MUST NOT contain the AS\_PATH attribute. The format of the Secure\_Path is described below in Section 3.1.

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

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

### 3.1. Secure\_Path

Here we provide a detailed description of the Secure\_Path information in the BGPSEC\_Path\_Signatures attribute.

#### Secure\_Path

```
+-----+
| Secure_Path Length           (2 octets) |
+-----+
| One or More Secure_Path Segments (variable) |
+-----+
```

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

The Secure\_Path contains one Secure\_Path segment for each (distinct) Autonomous System in the path to the NLRI specified in the update



message.

#### Secure\_Path Segment

AS Number	(4 octets)	
pCount	(1 octet)	
Flags	(1 octet)	

The AS Number is the AS number of the BGP speaker that added this Secure\_Path segment to the BGPSEC\_Path\_Signatures attribute. (See Section 4 for more information on populating this field.)

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

The first bit of the Flags field is the Entering\_Confed flag. The Entering\_Confed flag is set to one in the Secure\_Path Segment corresponding to the first Autonomous System in a confederation [3]. (That is, the Secure\_Path Segment corresponding to the AS that would otherwise have created an AS\_Path segment of type AS\_Confed\_Sequence in a non-BGPSEC update message.) In all other cases the Entering\_Confed flag is set to zero.

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

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

### 3.2. Additional\_Info

Here we provide a detailed description of the Additional\_Info in the BGPSEC\_Path\_Signatures attribute.

## Additional\_Info

Info Type	(1 octet)	
Info Length	(1 octet)	
Info Value	(variable)	

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

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

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

Implementations compliant with this specification MUST set the Info Type to zero in BGPSEC update messages for route advertisements that they originate (see Section 4.1 for more details). When an implementation compliant with this specification receives a BGPSEC update message with an Info Type field that it does not understand (i.e., an Info Type other than zero), the implementation MUST use the Additional\_Info when it verifies digital signatures (as per Section 5.1). However, other than signature verification, the implementation MUST ignore the Info Value field when it does not understand the Info Type.

EDITOR'S NOTE: In a previous version of this document there was an Expire Time that was used to provide protection against replay of old (stale) digital signatures or failure to propagate a withdrawal message. This mechanism was removed from the current version of the document. Please see the SIDR mailing list for discussions related to protection against replay attacks. Depending on the result of discussions within the SIDR working group this Additional Info field could at some future point be used to re-introduce Expire Time, or some other octets used in a future replay protection mechanism. The authors believe that the current instructions whereby the sender uses a null Additional\_Info type and the receiver ignores Additional\_Info types that it does not understand provides an opportunity to use

these octets in the future in a backwards-compatible fashion.

### 3.3. Signature\_Block

Here we provide a detailed description of the Signature\_Blocks in the BGPSEC\_Path\_Signatures attribute.

#### Signature\_Block

	Algorithm Suite Identifier	(1 octet)	
	Signature_Block Length	(2 octets)	
	Sequence of Signature Segments	(variable)	

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

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

A Signature\_Block has exactly one Signature Segment for each Secure\_Path Segment in the Secure\_Path portion of the BGPSEC\_Path\_Signatures Attribute. (That is, one Signature Segment for each distinct AS on the path for the NLRI in the Update message.)

#### Signature Segments

	Subject Key Identifier	(20 octets)	
	Signature Length	(2 octets)	
	Signature	(variable)	

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

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

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

#### 4. Generating a BGPSEC Update

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

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

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

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

update message for each NLRI.

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

#### 4.1. Originating a New BGPSEC Update

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

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

Note that the BGPSEC\_Path\_Signatures attribute and the AS4\_Path attribute are mutually exclusive. That is, any update message containing the BGPSEC\_Path\_Signatures attribute MUST NOT contain the AS4\_Path attribute nor the AS\_Path attribute. The information that would be contained in the AS4\_Path (or AS\_Path) attribute is instead conveyed in the Secure\_Path portion of the BGPSEC\_Path\_Signatures attribute.

Note that the Resource PKI enables the legitimate holder of IP address prefix(es) to issue a signed object, called a Route Origination Authorization (ROA), that authorizes a given AS to originate routes to a given set of prefixes (see [7]). Note that validation of a BGPSEC update message will fail (i.e., the validation algorithm, specified in Section 5.1, returns 'Not Good') unless there exists a valid ROA authorizing the first AS in the Secure\_Path portion of the BGPSEC\_Path\_Signatures attribute to originate routes to the prefix being advertised. Therefore, a BGPSEC speaker SHOULD NOT originate a BGPSEC update advertising a route for a given prefix

unless there exists a valid ROA authorizing the BGPSEC speaker's AS to originate routes to this prefix.

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

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

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

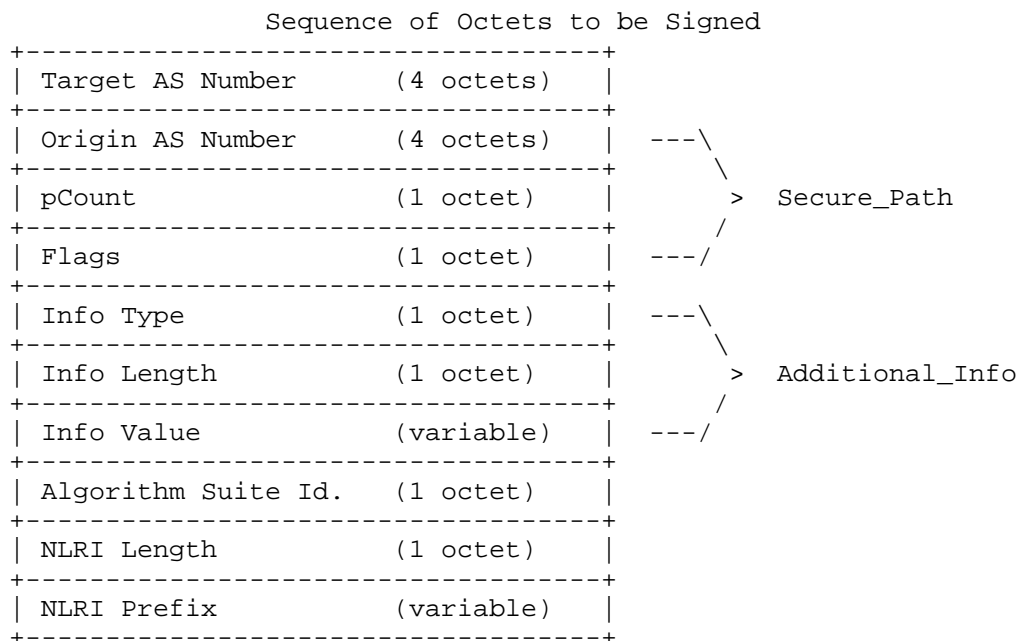
Typically, a BGPSEC speaker will use only a single algorithm suite, and thus create only a single Signature\_Block in the BGPSEC\_Path\_Signatures attribute. However, to ensure backwards compatibility during a period of transition from a 'current' algorithm suite to a 'new' algorithm suite, it will be necessary to originate update messages that contain a Signature\_Block for both the 'current' and the 'new' algorithm suites (see Section 6.1).

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

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

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

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



- o Apply to this octet sequence the digest algorithm (for the algorithm suite of this Signature\_Block) to obtain a digest value.
- o Apply to this digest value the signature algorithm, (for the algorithm suite of this Signature\_Block) to obtain the digital signature. Then populate the Signature Field with this digital signature.

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

#### 4.2. Propagating a Route Advertisement

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

If a BGPSEC router has received only non-BGPSEC update messages (without the BGPSEC\_Path\_Signatures attribute), containing the AS\_Path attribute, from a peer for a given prefix and if it chooses to propagate that peer's route for the prefix, then it **MUST NOT** attach any BGPSEC\_Path\_Signatures attribute to the corresponding update being propagated. (Note that a BGPSEC router may also receive a non-BGPSEC update message from an internal peer without the AS\_Path attribute, i.e., with just the NLRI in it. In that case, the prefix is originating from that AS and hence the BGPSEC speaker **SHOULD** sign and forward the update to its external peers, as specified in Section 4.1.)

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

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

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



BGP updates.

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

The pCount is typically set to the value 1. A BGPSEC speaker may set the pCount field to a value greater than 1. (See Section 4.1 for a discussion of setting pCount to a value greater than 1.) A route server that participates in the BGP control path, but does not act as a transit AS in the data plane, may choose to set pCount to 0. This option enables the route server to participate in BGPSEC and obtain the associated security guarantees without increasing the effective length of the AS path. (Note that BGPSEC speakers compute the effective length of the AS path by summing the pCount values in the BGPSEC\_Path\_Signatures attribute, see Section 5.) However, when a route server sets the pCount value to 0, it still inserts its AS number into the Secure\_Path segment, as this information is needed to validate the signature added by the route server. Note that the option of setting pCount to 0 is intended only for use by route servers that desire not to increase the effective AS-PATH length of routes they advertise. The pCount field SHOULD NOT be set to 0 in other circumstances. BGPSEC speakers SHOULD drop incoming update messages with pCount set to zero in cases where the BGPSEC speaker does not expect its peer to set pCount to zero (i.e., cases where the peer is not acting as a route server).

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

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

If the received BGPSEC update message contains two Signature\_Blocks and the BGPSEC speaker supports both of the corresponding algorithms suites, then the new update message generated by the BGPSEC speaker SHOULD include both of the Signature\_Blocks. If the received BGPSEC

update message contains two Signature\_Blocks and the BGPSEC speaker only supports one of the two corresponding algorithm suites, then the BGPSEC speaker MUST remove the Signature\_Block corresponding to the algorithm suite that it does not understand. If the BGPSEC speaker does not support the algorithm suites in any of the Signature\_Blocks contained in the received update message, then the BGPSEC speaker MUST NOT propagate the route advertisement with the BGPSEC\_Path\_Signatures attribute (i.e., propagate it as an unsigned BGP update message).

Note that in the case where there are two Signature\_Blocks (corresponding to different algorithm suites) that the validation algorithm (see Section 5.1) deems a BGPSEC update message to be 'Good' if there is at least one supported algorithm suite (and corresponding Signature\_Block) that is deemed 'Good'. This means that a 'Good' BGPSEC update message may contain a Signature\_Block which is not deemed 'Good' (e.g., contains signatures that the BGPSEC does not successfully verify). Nonetheless, such Signature\_Blocks MUST NOT be removed. (See Section 7 for a discussion of the security ramifications of this design choice.)

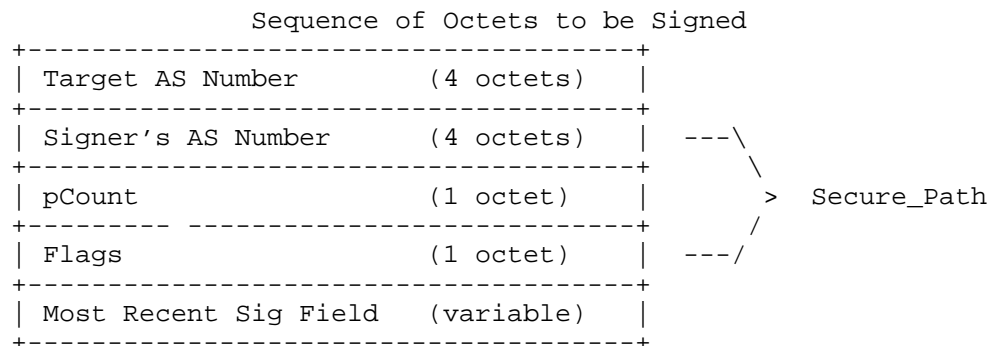
For each Signature\_Block corresponding to an algorithm suite that the BGPSEC speaker does support, the BGPSEC speaker then adds a new Signature Segment to the Signature\_Block. This Signature Segment is prepended to the list of Signature Segments (placed in the first position) so that the list of Signature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

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

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

- o Construct a sequence of octets by concatenating the Target AS number, the Secure\_Path segment that is being added by the BGPSEC speaker constructing the signature, and the signature field of the most recent Signature Segment (the one corresponding to AS from whom the BGPSEC speaker's AS received the announcement). Note that the Target AS number is the AS number announced by the peer

in the OPEN message of the BGP session within which the BGPSEC update message is sent.



- o Apply to this octet sequence the digest algorithm (for the algorithm suite of this Signature\_Block) to obtain a digest value.
- o Apply to this digest value the signature algorithm, (for the algorithm suite of this Signature\_Block) to obtain the digital signature. Then populate the Signature Field with this digital signature.

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

#### 4.3. Reconstructing the AS\_Path Attribute

EDITOR'S NOTE: This is a place-holder section. Given that BGPSEC update messages do not contain the AS\_Path attribute, this document needs to include a clearly specified algorithm for reconstructing the AS\_Path attribute from the data in the BGPSEC\_Path\_Signatures attribute. (For example, when propagating a path received via BGPSEC to a non-BGPSEC peer.) This algorithm for reconstructing the AS\_Path will appear in the next version of this document. In essence, the algorithm is: For each Secure\_Path Segment put into the AS\_Path pCount copies of the AS number field of the segment --- and if you see the Entering\_Confed flag is set to one, then add an AS\_Confed\_Sequence to the AS\_Path.

#### 4.4. Processing Instructions for Confederation Members

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

When a confederation member sends a BGPSEC update message to a peer

who is a member of the same confederation, the confederation member puts its (private) Member-AS Number (as opposed to the public AS Confederation Identifier) in the AS Number field of the Secure\_Path Segment that it adds to the BGPSEC update message.

Furthermore, when sending a BGPSEC update message to a peer who is a member of the same confederation, the first confederation member to add a Secure\_Path Segment to a BGPSEC update message sets the Entering\_Cofed flag in the Flags field to be one in the Secure\_Path Segment that it produces. In the case where the route advertisement originates within a confederation, the member AS that originates the route and sends it to a peer is the member AS that sets its Entering\_Confed flag to one. In the case where the route advertisement is received from outside the confederation, it is the member AS that receives the route advertisement from a peer outside the confederation and propagates it to a peer inside the confederation that sets its Entering\_Confed flag to one. Note that this is the same of the confederation member who would have added a new AS\_Path segment of type AS\_Confed\_Sequence to the AS\_Path attribute in a non-BGPSEC update message.

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

- o First, search through the Secure\_Path, going from most recently added segment to least recently added segment, and find first Secure\_Path Segment with the Entering\_Confed flag set to one. (That is, of all the Secure\_Path Segments with the Entering\_Confed flag set to one, find the one that was most recently added.)
- o Second, remove the Secure\_Path Segment found in previous step along with all more recently added Secure\_Path Segments. Keep a count of the number of segments removed in this fashion.
- o Third, starting with the most recently added Signature Segment, remove a number of Signature Segments equal to the number of Secure\_Path Segments removed in the previous step. (That is, remove the K most recently added signature segments, where K is the number of Secure\_Path Segments removed in the previous step.)
- o Finally, add a Secure\_Path Segment containing, in the AS field, the AS Confederation Identifier (the public AS number of the confederation) as well as a corresponding Signature Segment. Note that all fields other than the AS field are populated as per

Sections 4.1 and 4.2.

When validating a received BGPSEC update message, confederation members must make the following adjustment to the algorithm presented in Section 5.1. That is, when a confederation member is processing (validating) a Signature Segment and its corresponding Secure\_Path Segment, the confederation member must note that when a signature is produced by a BGPSEC speaker outside of a confederation, the Target AS will always be the AS Confederation Identifier (the public AS number of the confederation) as opposed to the Member-AS Number. To handle this case, when processing a current Secure\_Path Segment, if the next most recently added Secure\_Path segment has the Entering\_Confed flag set then, when computing the digest for the current Secure\_Path segment, take the Target AS Number to be the AS Confederation Identifier of the validating BGPSEC speaker's own confederation. (Note that the algorithm in Section 5.1 processes Secure\_Path Segments in order from most recently added to least recently added, therefore the algorithm encounters the Entering\_Confed flag immediately before it encounters the Secure\_Path segment that requires using the AS Confederation Identifier to validate.)

## 5. Processing a Received BGPSEC Update

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

- o For each valid RPKI end-entity certificate containing an AS Number extension, the AS Number, Public Key and Subject Key Identifier are required,
- o For each valid ROA, the AS Number and the list of IP address prefixes.

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

To validate a BGPSEC update message containing the BGPSEC\_Path\_Signatures attribute, the recipient performs the

validation steps specified in Section 5.1. The validation procedure results in one of two states: 'Good' and 'Not Good'.

It is expected that the output of the validation procedure will be used as an input to BGP route selection. However, BGP route selection and thus the handling of the two validation states is a matter of local policy, and shall be handled using existing local policy mechanisms. It is expected that BGP peers will generally prefer routes received via 'Good' BGPSEC update messages over routes received via 'Not Good' BGPSEC update messages as well as routes received via update messages that do not contain the BGPSEC\_Path\_Signatures attribute. However, BGPSEC specifies no changes to the BGP decision process and leaves to the operator the selection of an appropriate policy mechanism to achieve the operator's desired results within the BGP decision process.

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

Upon receiving a BGPSEC update message, a BGPSEC speaker SHOULD sum the pCount values within BGPSEC\_Path\_Signatures attribute to determine the effective length of the AS Path. The BGPSEC speaker SHOULD use this sum of pCount values in precisely the same way as it uses the length of the AS Path in non-BGPSEC update messages.

### 5.1. Validation Algorithm

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

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

- o Check to ensure that the entire BGPSEC\_Path\_Signatures attribute is syntactically correct (conforms to the specification in this

document).

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

If there are two `Signature_Blocks` within the `BGPSEC_Path_Signatures` attribute and one of them is poorly formed (or contains the wrong number of `Signature` segments), then the recipient should log that an error occurred, strip off that particular `Signature_Block` and process the update message as though it arrived with a single `Signature_Block`. If the `BGPSEC_Path_Signatures` attribute contains a syntax error that is not local to one of two `Signature_Blocks`, then the recipient should log that an error occurred and drop the update message containing the error. Similarly, if an update message contains both the `BGPSEC_Path_Signatures` attribute and either an `AS_Path` or `AS4_Path` attribute, then the recipient should log that an error occurred and drop the update message containing the error.

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

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

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

make use of this correspondence.

- o (Step I): Locate the public key needed to verify the signature (in the current Signature segment). To do this, consult the valid RPKI end-entity certificate data and look up all valid (AS, SKI, Public Key) triples in which the AS matches the AS number in the corresponding Secure\_Path segment. Of these triples that match the AS number, check whether there is an SKI that matches the value in the Subject Key Identifier field of the Signature segment. If this check finds no such matching SKI value, then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature-List Block.
- o (Step II): Compute the digest function (for the given algorithm suite) on the appropriate data. If the segment is not the (least recently added) segment corresponding to the origin AS, then the digest function should be computed on the following sequence of octets:

Sequence of Octets to be Hashed

AS Number of Target AS	(4 octets)	<div style="display: flex; align-items: center;"> <div style="text-align: right; margin-right: 5px;"> <div style="border-left: 1px dashed black; height: 100%;"></div> <div style="margin-left: 5px;"> <div style="text-align: right;">---</div> <div style="text-align: center;">&gt;</div> <div style="text-align: left;">---</div> </div> </div> </div>	Secure_Path
AS Number	(4 octets)		
pCount	(1 octet)		
Flags	(1 octet)		
Sig Field in the Next Segment	(variable)		

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

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



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

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

Sequence of Octets to be Hashed			
AS Number of Target AS (4 octets)			
Origin AS Number (4 octets)	---	\	> Secure_Path
pCount (1 octet)		/	
Flags (1 octet)	---	/	
Info Type (1 octet)	---	\	> Additional_Info
Info Length (1 octet)		/	
Info Value (variable)	---	/	
Algorithm Suite Id. (1 octet)			
NLRI Length (1 octet)			
NLRI Prefix (variable)			

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

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

- o (Step III): Use the signature validation algorithm (for the given algorithm suite) to verify the signature in the current segment. That is, invoke the signature validation algorithm on the following three inputs: the value of the Signature field in the current segment; the digest value computed in Step II above; and the public key obtained from the valid RPKI data in Step I above. If the signature validation algorithm determines that the signature is invalid, then mark the entire Signature-List Block as 'Not Good' and proceed to the next Signature\_Block. If the signature validation algorithm determines that the signature is valid, then continue processing Signature-Segments (within the current Signature-List Block).

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

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

## 6. Algorithms and Extensibility

### 6.1. Algorithm Suite Considerations

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

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

It is anticipated that in the future the mandatory algorithm suites document will be updated to specify a transition from the 'current' algorithm suite to the 'new' algorithm suite. During the period of

transition (likely a small number of years), all BGPSEC update messages SHOULD simultaneously use both the 'current' algorithm suite and the 'new' algorithm suite. (Note that Sections 3 and 4 specify how the BGPSEC\_Path\_Signatures attribute can contain signatures, in parallel, for two algorithm suites.) Once the transition is complete, use of the old 'current' algorithm will be deprecated, use of the 'new' algorithm will be mandatory, and a subsequent 'even newer' algorithm suite may be specified as recommend to implement. Once the transition has successfully been completed in this manner, BGPSEC speakers SHOULD include only a single Signature\_Block (corresponding to the 'new' algorithm).

## 6.2. Extensibility Considerations

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

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

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

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

## 7. Security Considerations

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

A BGPSEC speaker who receives a valid BGPSEC update message, containing a route advertisement for a given prefix, is provided with the following security guarantees:

- o The origin AS number corresponds to an autonomous system that has been authorized by the IP address space holder to originate route advertisements for the given prefix.
- o For each AS number in the AS Path, a BGPSEC speaker authorized by the holder of the AS number intentionally chose (in accordance with local policy) to propagate the route advertisement to the next AS in the Secure\_Path.

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

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

Note that there may be cases where a BGPSEC speaker deems 'Good' (as per the validation algorithm in Section 5.1) a BGPSEC update message that contains both a 'Good' and a 'Not Good' Signature\_Block. That is, the update message contains two sets of signatures corresponding to two algorithm suites, and one set of signatures verifies correctly and the other set of signatures fails to verify. In this case, the

protocol specifies that if the BGPSEC speaker propagates the route advertisement received in such an update message then the BGPSEC speaker SHOULD add its signature to each of the Signature\_Blocks using both the corresponding algorithm suite. Thus the BGPSEC speaker creates a signature using both algorithm suites and creates a new update message that contains both the 'Good' and the 'Not Good' set of signatures (from its own vantage point).

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

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

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

vantage points in the network, a BGPSEC update that was deemed 'Not Good' at an upstream BGPSEC speaker may indeed be deemed 'Good' at another BGP speaker downstream.

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

- o The BGPSEC speaker received the given route advertisement with the indicated NLRI and Secure\_Path; and
- o The BGPSEC speaker chose to propagate an advertisement for this route to the peer (implicitly) indicated by the 'Target AS'

The BGPSEC update validation procedure is a potential target for denial of service attacks against a BGPSEC speaker. To mitigate the effectiveness of such denial of service attacks, BGPSEC speakers should implement an update validation algorithm that performs expensive checks (e.g., signature verification) after performing less expensive checks (e.g., syntax checks). The validation algorithm specified in Section 5.1 was chosen so as to perform checks which are likely to be expensive after checks that are likely to be inexpensive. However, the relative cost of performing required validation steps may vary between implementations, and thus the algorithm specified in Section 5.1 may not provide the best denial of service protection for all implementations.

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

Finally, BGPSEC does not provide protection against all attacks at the transport layer. An adversary on the path between a BGPSEC speaker and its peer is able to perform attacks such as modifying valid BGPSEC updates to cause them to fail validation, injecting (unsigned) BGP update messages without BGPSEC\_Path\_Signature attributes, or injecting BGPSEC update messages with BGPSEC\_Path\_Signature attributes that fail validation, or causing the

peer to tear-down the BGP session. Therefore, BGPSEC implementations MUST support appropriate transport security mechanisms.

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

## 8. Contributors

### 8.1. Authors

Rob Austein  
Dragon Research Labs  
sra@hactrn.net

Steven Bellovin  
Columbia University  
smb@cs.columbia.edu

Randy Bush  
Internet Initiative Japan  
randy@psg.com

Russ Housley  
Vigil Security  
housley@vigilsec.com

Matt Lepinski  
BBN Technologies  
lepinski@bbn.com

Stephen Kent  
BBN Technologies  
kent@bbn.com

Warren Kumari  
Google  
warren@kumari.net

Doug Montgomery

USA National Institute of Standards and Technology  
dougmn@nist.gov

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

Samuel Weiler  
Cobham  
weiler+ietf@watson.org

## 8.2. Acknowledgements

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

## 9. Normative References

- [1] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4", RFC 4271, January 2006.
- [2] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.
- [3] Traina, P., McPherson, D., and J. Scudder, "Autonomous System Confederations for BGP", RFC 5065, August 2007.
- [4] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, February 2009.
- [5] Kumari, W. and K. Sriram, "Recommendation for Not Using AS\_SET and AS\_CONFED\_SET in BGP", RFC 6472, December 2011.
- [6] Lepinski, M. and S. Kent, "Recommendation for Not Using AS\_SET and AS\_CONFED\_SET in BGP", RFC 6480, February 2012.
- [7] Lepinski, M., Kent, S., and D. Kong, "Recommendation for Not Using AS\_SET and AS\_CONFED\_SET in BGP", RFC 6482, February 2012.
- [8] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.



- [9] Patel, K., Ward, D., and R. Bush, "Extended Message support for BGP", March 2011.
- [10] Kent, S., "Threat Model for BGP Path Security", February 2012.
- [11] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPSEC Router Certificates, Certificate Revocation Lists, and Certification Requests", December 2011.
- [12] Turner, S., "BGP Algorithms, Key Formats, & Signature Formats", March 2012.
- [13] Bush, R. and R. Austein, "The RPKI/Router Protocol", February 2012.

#### Author's Address

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

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



Secure Inter-Domain Routing (sidr)  
raft  
3

Kent, S. Internet D  
Kong, D. Expires: January 201  
Seo, K. Intended Status: BCP

BBN Technologies

July 9, 2012

Template for a Certification Practice State  
ment (CPS) for the Resource PKI (RPKI)

draft-ietf-sidr-cps-00.txt Status of this Memo This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as work in progress." The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html> The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html> This Internet-Draft will expire on January 31, 2013. Abstract This document contains a template to be used for creating a Certification Practice Statement (CPS) for an Organization that is part of the Resource Public Key Infrastructure (RPKI), e.g., a resource allocation registry or an ISP. Conventions used in this document The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119]. Kong, Seo & Kent Expires January 2013

[Page 1]

Internet-Draft	Template CPS for the RPKI	July 2012	Table of Contents
Contents	Preface.....	7	1
1. Introduction.....	8	1.1. Overview.....	8
1.2. Document Name and Identification.....	9	1.3. PKI Participants.....	9
1.3.1. Certification Authorities.....	9	1.3.2. Registration Authorities.....	9
1.3.3. Subscribers.....	10	1.3.4. Relying Parties.....	10
1.4. Certificate Usage.....	10	1.4.1. Appropriate Certificate Uses.....	10
1.4.2. Prohibited Certificate Uses.....	10	1.5. Policy Administration.....	11
1.5.1. Organization administering the document.....	11	1.5.2. Contact Person.....	11
1.5.3. Person Determining CPS Suitability for the Policy.....	11	1.5.4. CP S Approval Procedures.....	11
1.6. Definitions and Acronyms.....	11	2. Publication and Repository Responsibilities.....	14
2.1. Repositories.....	14	2.2. Publication of Certification Information.....	14
2.3. Time or Frequency of Publication.....	14	2.4. Access Controls on Repositories.....	14
3. Identification And Authentication.....	15	3.1. Naming.....	15
3.1.1. Types of Names.....	15	3.1.2. Need for Names to be Meaningful.....	15
3.1.3. Anonymity or Pseudonymity of Subscribers.....	15	3.1.4. Rules for Interpreting Various Name Forms.....	15
3.1.5. Uniqueness of Names.....	15	3.2. Initial Identity Validation.....	16
3.2.1. Method to Prove Possession of Private Key.....	16	3.2.2. Authentication of Organization Identity.....	16
3.2.3. Authentication of Individual Identity.....	16	3.2.4. Non-verified Subscriber Information.....	17
3.2.5. Validation of Authority.....	17	3.2.6. Criteria for Interoperation.....	17
3.3. Identification and Authentication for Routine Re-key Requests.....	17	3.3.1. Identification and Authentication for Routine Re-key after Revocation.....	18
3.4. Identification and Authentication for Revocation Request.....	18	4. Certificate Life-Cycle Operational Requirements.....	19

Internet-Draft	Template CPS for the RPKI	July 2012	4.1.
Certificate Application.....19			4.1.1. Who C
an Submit a Certificate Application.....19			4.1.2. Enrollment Proce
ss and Responsibilities.....19	4.2. Certificate Application Processi		
ng.....19	4.2.1. Performing Identification and Authenti		
cation Functions19	4.2.2. Approval or Rejection of Certificate Applicatio		
ns....19	4.2.3. Time to Process Certificate Applications.....20		
4.3. Certificate Issuance.....20			4.3.1.
CA Actions During Certificate Issuance.....20			4.3.2. Notificati
on to Subscriber by the CA of Issuance of	Certificate.....		
.....20	4.3.3. Notification of Certificate Issuanc		
e by the CA to Other	Entities.....		
.....20	4.4. Certificate Acceptance.....20		
4.4.1. Conduct Constituting Certificate Acceptance.....20			4.4.
2. Publication of the Certificate by the CA.....20			4.5. Key Pair and
Certificate Usage.....20			4.5.1. Subscriber Private
Key and Certificate Usage.....21			4.5.2. Relying Party Public Key and C
ertificate Usage.....21	4.6. Certificate Renewal.....		
.....21	4.6.1. Circumstance for Certificate Renewal.....		
21	4.6.2. Who May Request Renewal.....21		
4.6.3. Processing Certificate Renewal Requests.....22			4.6.4. Noti
fication of New Certificate Issuance to Subscriber22			4.6.5. Conduct Const
ituting Acceptance of a Renewal Certificate	.....		
.....22	4.6.6. Publication of the Renewal Certif		
icate by the CA....22	4.6.7. Notification of Certificate Issuance by the		
CA to Other	Entities.....22		
4.7. Certificate Re-key.....22			4.
7.1. Circumstance for Certificate Re-key.....22			4.7.2. Who Ma
y Request Certification of a New Public Key...23			4.7.3. Processing Certif
icate Re-keying Requests.....23			4.7.4. Notification of New Certific
ate Issuance to Subscriber23	4.7.5. Conduct Constituting Acceptance of a		
Re-keyed	Certificate.....23		
4.7.6. Publication of the Re-keyed Certificate by the CA...23			4.7.7
. Notification of Certificate Issuance by the CA to Other	Entities.....		
.....24	4.8. Certificate Modification		
.....24	4.8.1. Circumstance for Certificate M		
odification.....24	4.8.2. Who May Request Certificate modification.		
.....24	4.8.3. Processing Certificate Modification Requests.....		
24	4.8.4. Notification of Modified Certificate Issuance to		Subscr
iber.....24			
nstituting Acceptance of Modified Certificate	.....		4.8.5. Conduct Co
.....25	Kong, Seo & Kent	Expires January 2013	

Internet-Draft	Template CPS for the RPKI	July 2012	4
4.8.6. Publication of the Modified Certificate by the CA...	25	4.8.7. Notification of Certificate Issuance by the CA to Other Entities.....	
.....25		4.9. Certificate Revocation and Suspension.....	
.....25		4.9.1. Circumstances for Revocation.....	
.....25		4.9.2. Who Can Request Revocation.....	
.....25		4.9.3. Procedure for Revocation Request.....	25
4.9.4. Revocation Request Grace Period.....	26	4.9.5. Time Within Which CA Must Process the Revocation Request .....	
.....26		4.9.6. Revocation Checking Requirement for Relying Parties.....	26
.....26		4.9.7. CRL Issuance Frequency.....	
.....26		4.9.8. Maximum Latency for CRLs.....	
.....26		4.10. Certificate Status Services.....	2
6	5. Facility, Management, and Operational Controls.....	27	5.1. Physical Controls.....
	location and construction.....	27	5.1.1. Site location and construction.....
	.....27		5.1.2. Physical access.....
	.....27		5.1.3. Power and air conditioning.....
	.....27		5.1.4. Water exposures.....
	.....27		5.1.5. Fire prevention and protection.....
	.....27		5.1.6. Media storage.....
	5.1.7. Waste disposal.....	27	5.1.8. Off-site backup.....
	.....27		5.2. Procedural Controls.....
	.....27		5.2.1. Trusted roles.....
	.....27		5.2.2. Number of persons required per task.....
	.....27		5.2.3. Identification and authentication for each role.....
	.....27		5.2.4. Roles requiring separation of duties.....
	5.3. Personnel Controls.....	27	5.3.1. Qualifications, experience, and clearance requirements.....
	.....28		5.3.2. Background check procedures.....
	.....28		5.3.3. Training requirements.....
	.....28		5.3.4. Retraining frequency and requirements.....
	.....28		5.3.5. Job rotation frequency and sequence.....
	.....28		5.3.6. Sanctions for unauthorized actions.....
	.....28		5.3.7. Independent contractor requirements.....
	5.3.8. Documentation supplied to personnel.....	28	5.4. Audit Logging Procedures.....
	.....28		5.4.1. Types of Events Recorded.....
	.....28		5.4.2. Frequency of Processing Log.....
	.....28		5.4.3. Retention Period for Audit Log.....
	.....29		5.4.4. Protection of Audit Log.....
	.....29		5.4.5. Audit Log Backup Procedures.....
	5.4.6. Audit Collection System (Internal vs. External) [OMITTED]...	29	5.4.7. Notification to Event-causing Subject [OMITTED].....
	.....29		Kong, Seo & Kent Expires January 20

Internet-Draft	Template CPS for the RPKI	July 2012	5
4.8. Vulnerability Assessments.....	29	5.5. Records archival [OMITTED].....	29
.....	29	5.6. Key Changeover.....	29
[OMITTED].....	29	5.7. Compromise and disaster recovery [OMITTED].....	29
.....	29	5.8. CA or RA Termination.....	29
...30	6. Technical Security Controls.....		30
	6.1. Key Pair Generation and Installation.....		30
	6.1.1. Key Pair Generation.....		30
private Key Delivery to Subscriber.....	30	6.1.2. Public Key Delivery to Certificate Issuer.....	30
.....	30	6.1.3. CA Public Key Delivery to Relying Parties.....	30
.....	30	6.1.4. Key Sizes.....	30
Checking31	6.1.5. Public Key Parameters Generation and Quality		31
	6.1.6. Key Usage Purposes (as per X.509 v3 Key Usage Field)		31
	6.1.7. Private Key Protection and Cryptographic Module Engineering		31
ls.....	31	6.2.1. Cryptographic module standards and controls.....	31
of m) Multi-Person Control.....	31	6.2.2. Private Key Escrow.....	31
.....	31	6.2.3. Private Key Backup.....	31
.....	31	6.2.4. Private Key Archival.....	31
.32	6.2.5. Private Key Transfer into or from a Cryptographic Module		32
	.....		32
private Key Storage on Cryptographic Module.....	32	6.2.6. Method of Activating Private Key.....	32
.....	32	6.2.7. Method of Deactivating Private Key.....	32
.....	32	6.2.8. Method of Destroying Private Key.....	32
.....	32	6.2.9. Cryptographic Module Rating.....	32
.....	32	6.3. Other aspects of Key Pair Management.....	32
	6.3.1. Public Key Archival.....		32
2. Certificate Operational Periods and Key Pair Usage		Periods.....	33
.....	33	6.4. Activation data.....	33
.....	33	6.4.1. Activation Data Generation and Installation.....	33
.....	33	6.4.2. Activation data protection.....	33
.....	33	6.4.3. Other Aspects of Activation Data.....	33
	6.5. Computer Security Controls.....		33
life cycle Technical Controls.....	33	6.6. System Development Controls.....	33
.....	33	6.6.1. Security Management Controls.....	33
.....	34	6.6.2. Life Cycle Security Controls.....	34
.....	34	6.7. Network Security Controls.....	34
34	6.8. Time-stamping.....		35
7. Certificate and CRL Profiles.....	35	8. Compliance Audit and Other Assessments.....	36
And Legal Matters.....	37	9. Other Business And Legal Matters.....	37
.....	38	9.1. Fees.....	38
013	Kong, Seo & Kent	Expires January 2	

Internet-Draft	Template CPS for the RPKI	July 2012	9
9.1.1. Certificate issuance or renewal fees.....	38	9.1.2. Fees	
for other services (if applicable).....	38	9.1.3. Refund policy...	
.....	38	9.2. Financial responsibility.....	
.....	38	9.2.1. Insurance coverage.....	
.....	38	9.2.2. Other assets.....	
.....	38	9.2.3. Insurance or warranty coverage for end-entities.....	38
9.3. Confidentiality of business information.....	38	9.3.1. S	
cope of confidential information.....	38	9.3.2. Information	
not within the scope of confidential		information.....	
.....	38	9.3.3. Responsibility to protect confidential i	
nformation..	38	9.4. Privacy of personal information.....	
.....	38	9.4.1. Privacy plan.....	38
9.4.2. Information treated as private.....	38	9.4.3. Inf	
ormation not deemed private.....	38	9.4.4. Responsibility	
to protect private information.....	38	9.4.5. Notice and consent to use	
private information.....	38	9.4.6. Disclosure pursuant to judicial or a	
dministrative		process.....	
38	9.4.7. Other information disclosure circumstances.....	38	9.5
. Intellectual property rights (if applicable).....	38	9.6. Represent	
ations and warranties.....	38	9.6.1. CA representati	
ons and warranties.....	38	9.6.2. Subscriber representations	
and warranties.....	39	9.6.3. Relying party representations and war	
ranties.....	39	9.7. Disclaimers of warranties.....	
....	39	9.8. Limitations of liability.....	39
9.9. Indemnities.....	39	9.10. Term	
and termination.....	39	9.10.1. Term.....	
.....	39	9.10.2. Termination.....	
.....	39	9.10.3. Effect of termination and surviv	
al.....	39	9.11. Individual notices and communications with parti	
cipants.	39	9.12. Amendments.....	39
9.12.1. Procedure for amendment.....	39	9.1	
2.2. Notification mechanism and period.....	39	9.13. Dispute res	
olution provisions.....	39	9.14. Governing law.....	
.....	39	9.15. Compliance with applicable law...	
.....	39	9.16. Miscellaneous provisions.....	
.....	39	9.16.1. Entire agreement.....	
.....	39	9.16.2. Assignment.....	39
9.16.3. Severability.....	39	9.16.4. En	
forcement (attorneys' fees and waiver of rights).	39	9.16.5. Force Majeure	
.....	39	10. Security Considerations.....	
.....	39	Kong, Seo & Kent	Expires January 2013



Internet-Draft                      Template CPS for the RPKI                      July 2012    11. IAN

A Considerations.....40    12. Acknowledgment

S.....41    13. References.....

.....41    13.1. Normative References.....

.....41    13.2. Informative References.....

.....41    Author's Addresses.....

...42    Copyright Statement.....42

Prefac

e    This document contains a template to be used for creating a    Certification P

ractice Statement (CPS) for an Organization that is    part of the Resource Public

Key Infrastructure (RPKI). The user of    this document should:    1. substitut

e a title page for page 1 saying, e.g., "<Name of    Organization> Certificat

ion Practice Statement for the Resource    Public Key Infrastructure (RPKI)"

with date, author, etc. There    is no expectation that a CPS will be publish

ed as an RFC.    2. leave the table of contents intact    3. delete this Prefac

e, headers and footers (but keep page numbers)    4. fill in the information ind

icated below by <text in angle    brackets>    5. delete sections 10, 11, 12

, 13.1, Acknowledgments, Author's    Addresses, Intellectual Property Stateme

nt, Disclaimer of    Validity, Copyright Statement, Acknowledgments; leaving

a    reference section with just the references in 13.2    6. update the tab

le of contents to reflect the changes required by    steps 4 and 5 above .

This document has been generated to complement the Certificate Policy    (CP) for

the RPKI [RFC6484]. Like the RPKI CP, it is is based on the    template specified

in RFC 3647. A number of sections contained in the    template were omitted from t

his CPS because they did not apply to    this PKI. However, we have retained the s

ection numbering scheme    employed in the RFC to facilitate comparison with the s

ection    numbering scheme employed in that RFC and in the RPKI CP. Kong, Seo & Ken

t    Expires January 2013    [Page 7]

Internet-Draft                      Template CPS for the RPKI                      July 2012

1. Introduction      This document is the Certification Practice Statement (CPS) of <Name of Organization>. It describes the practices employed by the <Name of Organization> Certification Authority (CA) in the Resource Public Key Infrastructure (RPKI). These practices are defined in accordance with the requirements of the Certificate Policy (CP, [RFC6484]) for the RPKI. The RPKI is designed to support validation of claims by current holders of Internet Number Resources (INRs, see definition in Section 1.7) in accordance with the records of the organizations that act as CAs in this PKI. The ability to verify such claims is essential to ensuring the unique, unambiguous distribution of these resources. This PKI parallels the existing INR distribution hierarchy. These resources are distributed by the Internet Assigned Numbers Authority (IANA) to the Regional Internet Registries. In some regions, National Internet Registries (NIRs) form a tier of the hierarchy below the RIRs for internet number resource (INR) distribution. Internet Service Providers (ISPs) and network subscribers form additional tiers below registries.

1.1. Overview      This CPS describes:

- 1.1.1. Participants
- 1.1.2. Publication of the certificates and CRLs
- 1.1.3. How certificates are issued, managed, and revoked
- 1.1.4. Facility management (physical security, personnel, audit, etc.)
- 1.1.5. Key management
- 1.1.6. Audit procedures
- 1.1.7. Business and legal issues

This PKI encompasses several types of certificates (see [RFC6480] for more details):

Kong, Seo & Kent                      Expires January 2013                      [Page 8]

Internet-Draft                      Template CPS for the RPKI                      July 2012   . CA certificates for each organization distributing INRs and for each subscriber INR holder) . End entity (EE) certificates for organizations to use to validate digital signatures on RPKI-signed objects (see definition in Section 1.7).

. In the future, the PKI also may include end entity certificates in support of access control for the repository system as described in 2.4.1.2. Document Name and Identification   The name of this document is "<Name of Organization>'s Certification Practice Statement for the Resource Public Key Infrastructure (RPKI)".

1.3. PKI Participants   Note that in a PKI, the term "subscriber" refers to an individual or organization that is a subject of a certificate issued by a CA. The term is used in this fashion throughout this document, without qualification, and should not be confused with the networking use of the term to refer to an individual or organization that receives service from an ISP. In such cases the term "network subscriber" will be used. Also note that, for brevity, this document always refers to PKI participants as organizations or entities, even though some of them are individuals.

1.3.1. Certification Authorities   <Describe the CAs that you will operate for the RPKI. One approach is to operate two CAs: one designated "offline" and the other designated "production." The offline CA is the top level CA for the <Name of Organization> portion of the RPKI. It provides a secure revocation and recovery capability in case the production CA is compromised or becomes unavailable. Thus the offline CA issues certificates only to instances of the production CA; and the CRLs it issues are used to revoke only certificates issued to the production CA. The production CA is used to issue RPKI certificates to <Name of Organization> members, to whom INRs have been distributed.>

1.3.2. Registration Authorities   <Describe how the registration authority function is handled for the CA(s) that you operate. The RPKI does not require establishment or use of a separate registration authority (RA) in conjunction with the

Kong, Seo & Kent   Expires January 2013

Internet-Draft                      Template CPS for the RPKI                      July 2012      CA func

tion. The RA function MUST be provided by the same entity      operating as a CA, e. g., entities listed in Section 1.3.1. An entity      acting as a CA in this PKI already has a formal relationship with      each organization to which it distributes INRs. These organizations      already perform the RA function implicitly since they already assume      responsibility for distributing INRs.>                      1.3.3. Subscribers

Organizations receiving INR allocations from this CA are subscribers      in the RPKI.                      1.3.4. Relying Parties      Entities or individuals that act in reliance on certificates or RPKI-      signed objects issued under this PKI are relying parties. Relying      parties may or may not be subscribers within this PKI. (See Section 1.7 for the definition of an RPKI-signed object.)                      1.3.5. Other Participants

<Specify the entity that operates a repository holding certificates,      CRLs, and other RPKI-signed objects issued by this Organization, and      provide a URL for the repository.>1.4. Certificate Usage                      1.4.1. Appropriate Certificate Uses

The certificates issued under this hierarchy are for authorization in      support of validation of claims of current holdings of INRs. Additional uses of the certificates, consistent with the basic goal      cited above, are also permitted under the RPKI CP [RFC6484].      Some of the certificates that may be issued under this PKI could be      used to support operation of this infrastructure, e.g., access control for the repository system as described in 2.4. Such uses also      are permitted under the RPKI certificate policy.                      1.4.2. Prohibited Certificate Uses

Any uses other than those described in Section 1.4.1 are prohibited.Kong, Se o & Kent                      Expires January 2013                      [Page 10]

Internet-Draft                      Template CPS for the RPKI                      July 2012

1.5. Policy Administration                      1.5.1. Organization administering the document                      This CPS is administered by <Name of Organization>                      1.5.2. Contact Person                      <Insert Organization contact info here>                      1.5.3. Person Determining CPS Suitability for the Policy                      Not applicable. Each organization issuing a certificate in this PKI is attesting to the distribution of INRs to the holder of the private key corresponding to the public key in the certificate. The issuing organizations are the same organizations as the ones that perform the distribution hence they are authoritative with respect to the accuracy of this binding.                      1.5.4. CPS Approval Procedures                      Not applicable. Each organization issuing a certificate in this PKI is attesting to the distribution of INRs to the holder of the private key corresponding to the public key in the certificate. The issuing organizations are the same organizations as the ones that perform the distribution, hence they are authoritative with respect to the accuracy of this binding.

1.6. Definitions and Acronyms

BPKI - Business PKI: A BPKI is an optional additional PKI used by an Organization to identify members to whom RPKI certificates can be issued.

CP - Certificate Policy. A CP is a named set of rules that indicates the applicability of a certificate to a particular community and/or class of applications with common security requirements.                      The CP for the RPKI is [RFC6384].

CPS - Certification Practice Statement. A CPS is a document that specifies the practices that a Certification Authority employs in issuing certificates.

Kong, Seo & Kent                      Expires January 2013

Internet-Draft                      Template CPS for the RPKI                      July 2012

Distribution of INRs - A process of distribution of the INRs along the respective number hierarchy. IANA distributes blocks of IP addresses and Autonomous System Numbers to the five Regional Internet Registries (RIRs). RIRs distribute smaller address blocks and Autonomous System Numbers to organizations within their service regions, who in turn distribute IP addresses to their customers. IANA - Internet Assigned Numbers Authority. IANA is responsible for global coordination of the Internet Protocol addressing systems and Autonomous System (AS) numbers used for routing internet traffic. IANA distributes INRs to Regional Internet Registries (RIRs). INRs - Internet Number Resources. INRs are number values for three protocol parameter sets, namely:

- . IP Version 4 addresses,
- . IP version 6 addresses, and
- . Identifiers used in Internet inter-domain routing, currently Border Gateway Protocol-4 Autonomous System numbers.

ISP - Internet Service Provider. An ISP is an organization managing and selling Internet services to other organizations. NIR - National Internet Registry. An NIR is an organization that manages the distribution of INRs for a portion of the geopolitical area covered by a Regional Registry. NIRs form an optional second tier in the tree scheme used to manage INR distribution. RIR - Regional Internet Registry. An RIR is an organization that manages the distribution of INRs for a geopolitical area. RPKI-signed object - An RPKI-signed object is a digitally signed data object (other than a certificate or CRL) declared to be such by a standards track RFC, and that can be validated using certificates issued under this PKI. The content and format of these data constructs depend on the context in which validation of claims of current holdings of INRs takes place. Examples of Kong, Seo & Kent Expires January 2013

[Page 12]

Internet-Draft                      Template CPS for the RPKI                      July 2012                      the  
se objects are repository manifests [RFC6486] and Route                      Origin Authorizatio  
ns (ROAs) [RFC6482].Kong, Seo & Kent                      Expires January 2013  
[Page 13]

Internet-Draft                      Template CPS for the RPKI                      July 20122. Publication and Repository Responsibilities

2.1. Repositories      As per the CP, certificates, CRLs and RPKI-signed objects must be made available for downloading by all relying parties, to enable them to validate this data. The <Name of Organization> RPKI CA will publish certificates, CRLs, and RPKI-signed objects via a repository that is accessible via RSYNC at <insert URL here>. This repository will conform to the structure described in [RFC6481].

2.2. Publication of Certification Information      <Name of Organization> will publish certificates, CRLs and RPKI-signed objects issued by it to a repository that operates as part of a world-wide distributed system of RPKI repositories.

2.3. Time or Frequency of Publication      <Describe here your procedures for publication (to the global repository system) of the certificates, CRLs and RPKI-signed objects that you issue. If you choose to outsource publication of PKI data, you still need to provide this information for relying parties. This should include the period of time within which a certificate will be published after the CA issues the certificate, and the period of time within which a CA will publish a CRL with an entry for a revoked certificate, after the CA revokes that certificate.> The <Name of Organization> CA will publish its CRL prior to the nextScheduledUpdate value in the scheduled CRL previously issued by the CA.

2.4. Access Controls on Repositories      <Describe the access controls used by the Organization to ensure that only authorized parties can modify repository data, and any controls used to mitigate denial of service attacks against the repository. If the Organization offers repository services to its subscribers, then describe here the protocol(s) that it supports for publishing signed objects from subscribers.>

Expires January 2013

[Page 14]

Kong, Seo & Kent



Internet-Draft                      Template CPS for the RPKI                      July 20123. Identification And Authentication3.1. Naming                      3.1.1. Types of Names                      The subject of each certificate issued by this Organization is identified by an X.500 Distinguished Name (DN). The distinguished name will consist of a single Common Name (CN) attribute with a value generated by <Name of Organization>. Optionally, the serialNumber attribute may be included along with the common name (to form a terminal relative distinguished name set), to distinguish among successive instances of certificates associated with the same entity.                      3.1.2. Need for Names to be Meaningful                      The subject name in each subscriber certificate will be unique relative to all certificates issued by <Name of Organization>. However, there is no guarantee that the subject name will be globally unique in this PKI. Also, the name of the subscriber will not be "meaningful" in the conventional, human-readable sense. The rationale here is that these certificates are used for authorization in support of applications that make use of attestations of INR holdings. They are not used to identify subjects.                      3.1.3. Anonymity or Pseudonymity of Subscribers                      Although Subject names in certificates issued by this Organization need not be meaningful, and may appear "random," anonymity is not a function of this PKI; thus no explicit support for this feature is provided.                      3.1.4. Rules for Interpreting Various Name Forms                      None                      3.1.5 . Uniqueness of Names                      <Name of Organization> certifies subject names that are unique among the certificates that it issues. Although it is desirable that these subject names be unique throughout the PKI, to facilitate certificate path discovery, such uniqueness is neither mandated nor enforced through technical means. <Name of Organization> generates subject names to minimize the chances that two entities in the RPKI will beKong, Seo & Kent                      Expires January 2013

Internet-Draft                      Template CPS for the RPKI                      July 2012      assigne  
d the same name. Specifically, <insert subject name generation      description here  
, or cite RFC 6487.>                      3.1.6. Recognition, Authentication, and Role of Radem  
arks      Because the Subject names are not intended to be meaningful, <Name of      Or  
ganization> makes no provision to either recognize or authenticate      trademarks,  
service marks, etc.3.2. Initial Identity Validation                      3.2.1. Method to Prove  
Possession of Private Key      <Describe the method whereby each subscriber will be  
required to      demonstrate proof-of-possession (PoP) of the private key      correspo  
nding to the public key in the certificate, prior to      certificate issuance.>  
3.2.2. Authentication of Organization Identity      Certificates issued under thi  
s PKI do not attest to the      organizational identity of subscribers. However, cer  
tificates are      issued to subscribers in a fashion that preserves the accuracy of  
distributions of INRs as represented in <Name of Organization's>      records.  
<Describe the procedures that will be used to ensure that each RPKI      certificate  
that is issued, accurately reflects your records with      regard to the organizati  
on to which you have distributed (or sub-      distributed) the INRs identified in t  
he certificate. For example, a      BPKI certificate could be used to authenticate a  
certificate request      that serves as a link to the <Name of Organization's> subs  
criber      database that maintains the INR distribution records. The certificate  
request could be matched against the database record for the      subscriber in ques  
tion, and an RPKI certificate would be issued only      if the INRs requested were a  
subset of those held by the subscriber.      The specific procedures employed for t  
his purpose should be      commensurate with any you already employ in the maintenanc  
e of INR      distribution.>                      3.2.3. Authentication of Individual Identity      C  
ertificates issued under this PKI do not attest to the individual      identity of a  
subscriber. However, <Name of Organization> maintains      contact information for  
each subscriber in support of certificate      renewal, re-key, and revocation.Kong,  
Seo & Kent                      Expires January 2013                      [Page 16]

Internet-Draft                      Template CPS for the RPKI                      July 2012      <Describe the procedures that are used to identify at least one individual as a representative of each subscriber. This is done in support of issuance, renewal, and revocation of the certificate issued to the organization. For example, one might say "The <Name of Organization> BPKI (see Section 3.2.6) issues certificates that MUST be used to identify individuals who represent <Name of Organization> subscribers." The procedures should be commensurate with those you already employ in authenticating individuals as representatives for INR holders. Note that this authentication is solely for use by you in dealing with the organizations to which you distribute (or sub-distribute) INRs, and thus must not be relied upon outside of this CA/subscriber relationship.>

3.2.4. Non-verified Subscriber Information      No non-verified subscriber data is included in certificates issued under this certificate policy except for Subject Information Access (SIA) extensions [RFC6487].

3.2.5. Validation of Authority      <Describe the procedures used to verify that an individual claiming to represent a subscriber, is authorized to represent that subscriber in this context. For example, one could say, "Only an individual to whom a BPKI certificate (see Section 3.2.6) has been issued may request issuance of an RPKI certificate. Each certificate issuance request is verified using the BPKI." The procedures should be commensurate with those you already employ in authenticating individuals as representatives of subscribers.>

3.2.6. Criteria for Interoperation      The RPKI is neither intended nor designed to interoperate with any other PKI. <If you operate a separate, additional PKI for business purposes, e.g., a BPKI, then describe (or reference) how the BPKI is used to authenticate subscribers and to enable them to manage their resource distributions.>

3.3. Identification and Authentication for Re-key Requests

3.3.1. Identification and Authentication for Routine Re-key      <Describe the conditions under which routine re-key is required and the manner by which it is requested. Describe the procedures that are used to ensure that a subscriber requesting routine re-key is the legitimate holder of the certificate to be re-keyed. State the approach for establishing PoP of the private key corresponding to the

Kong, Seo & Kent                      Expires January 2013

Internet-Draft                      Template CPS for the RPKI                      July 2012    new public key. If you operate a BPKI, describe how that BPKI is used to authenticate routine re-key requests.>                      3.3.2. Identification and Authentication for Re-key after                      Revocation    <Describe the procedures used to ensure that an organization requesting a re-key after revocation is the legitimate holder of the INRs in the certificate being re-keyed. This should also include the method employed for verifying PoP of the private key corresponding to the new public key. If you operate a BPKI, describe how that BPKI is used to authenticate re-key requests. With respect to authentication of the subscriber, the procedures should be commensurate with those you already employ in the maintenance of INR distribution records.>3.4. Identification and Authentication for Revocation Request                      <Describe the procedures used by an RPKI subscriber to make a revocation request. Describe the manner by which it is ensured that the subscriber requesting revocation is the subject of the certificate (or an authorized representative thereof) to be revoked. Note that there may be different procedures for the case where the legitimate subject still possesses the original private key as opposed to the case when it no longer has access to that key. These procedures should be commensurate with those you already employ in the maintenance of subscriber records.>Kong, Seo & Kent                      Expires January 2013                      [Page 18]

Internet-Draft                      Template CPS for the RPKI                      July 20124. Certificate Life-Cycle Operational Requirements4.1. Certificate Application                      4.1.1. Who Can Submit a Certificate Application                      Any subscriber who holds INRs distributed by this Organization may submit a certificate application to this CA.

4.1.2. Enrollment Process and Responsibilities                      <Describe your enrollment process for issuing certificates both for initial deployment of the PKI and as an ongoing process. Note that most of the certificates in this PKI are issued as part of your normal business practices, as an adjunct to INR distribution, and thus a separate application to request a certificate may not be necessary. If so, reference should be made to where these practices are documented.>

4.2. Certificate Application Processing                      <Describe the certificate request/response processing that you will employ. You should make use of existing standards for certificate application processing (see [RFC6487]).>

4.2.1. Performing Identification and Authentication Functions                      <Describe your practices for identification and authentication of certificate applicants. Often, existing practices employed by you to identify and authenticate organizations can be used as the basis for issuance of certificates to these subscribers. Reference can be made to documentation of such existing practices.>

4.2.2. Approval or Rejection of Certificate Applications                      <Describe your practices for approval or rejection of applications and refer to documentation of existing business practices relevant to this process. Note that according to the CP, certificate applications will be approved based on the normal business practices of the entity operating the CA, based on the CA's records of subscribers. The CP also says that each CA will follow the procedures specified in 3.2.1 to verify that the requester holds the private key corresponding to the public key that will be bound to the certificate the CA issues to the requester.>

Kong, Seo & Kent                      Expires January 2013                      [Page 19]

Internet-Draft	Template CPS for the RPKI	July 2012	4.2
----------------	---------------------------	-----------	-----

.3. Time to Process Certificate Applications <Specify here your expected time frame for processing certificate applications.>4.3. Certificate Issuance 4

.3.1. CA Actions During Certificate Issuance <Describe your procedures for issuance and publication of a certificate.> 4.3.2. Notification to Subscriber by the CA of Issuance of Certificate <Name of Organization> will notify the subscriber when the certificate is published. <Describe here your procedures for notifying a subscriber when a certificate has been published.> 4

.3.3. Notification of Certificate Issuance by the CA to Other Entities <Describe here any other entities that will be notified when a certificate is published.>4.4. Certificate Acceptance 4.4.1. Conduct Constituting Certificate Acceptance When a certificate is issued, the <name of Organization> CA will publish it to the repository and notify the subscriber. <This may be done without subscriber review and acceptance. State your policy with respect to subscriber certificate acceptance here.> 4.4.2. Publication of the Certificate by the CA Certificates will be published at <insert repository URL here> once issued, following the conduct described in 4.4.1. This will be done within <specify the timeframe within which the certificate will be placed in the repository and the subscriber will be notified>.<Describe any additional procedures with respect to publication of the certificate here.>4.5. Key Pair and Certificate Usage A summary of the use model for the RPKI is provided below.Kong, Seo & Kent Expires January 2013 [Page 20]

4.5.1. Subscriber Private Key and Certificate Usage The certificates issued by <Name of Organization> to subordinate INR holders are CA certificates. The private key associated with each of these certificates is used to sign subordinate (CA or EE) certificates and CRLs.

4.5.2. Relying Party Public Key and Certificate Usage The primary relying parties in this PKI are organizations that use RPKI EE certificates to verify RPKI-signed objects. Relying parties are referred to Section 4.5.2 of [RFC6484] for additional guidance with respect to acts of reliance on RPKI certificates.

4.6. Certificate Renewal 4.6.1. Circumstance for Certificate Renewal As per the RPKI CP, a certificate will be processed for renewal based on its expiration date or a renewal request from the certificate subject. The request may be implicit, a side effect of renewing a resource holding agreement, or may be explicit. If <Name of Organization> initiates the renewal process based on the certificate expiration date, then <Name of Organization> will notify the subscriber <insert the period of advance warning, e.g., "2 weeks in advance of the expiration date", or the general policy, e.g., "in conjunction with notification of service expiration".> The validity interval of the new (renewed) certificate will overlap that of the previous certificate by <insert length of overlap period, e.g., 1 week>, to ensure uninterrupted coverage. Certificate renewal will incorporate the same public key as the previous certificate, unless the private key has been reported as compromised. If a new key pair is being used, the stipulations of Section 4.7 will apply.

4.6.2. Who May Request Renewal The subscriber or <Name of Organization> may initiate the renewal process. <For the case of the subscriber, describe the procedures that will be used to ensure that the requester is the legitimate holder of the INRs in the certificate being renewed. This should also include the method employed for verifying PoP of the private key corresponding to the public key in the certificate being renewed or the new public key if the public key is being changed. With respect to authentication of the subscriber, the procedures should be commensurate with those you already employ in the maintenance of INR>.

Internet-Draft                      Template CPS for the RPKI                      July 2012    distrib  
ution records. If you operate a BPKI for this, describe how    that business-based  
PKI is used to authenticate renewal requests and    refer to 3.2.6.>                      4.6.3.  
Processing Certificate Renewal Requests    <Describe your procedures for handling  
certificate renewal requests. Describe how you verify that the requester is the  
subscriber or is    authorized by the subscriber, and that the certificate in quest  
ion    has not been revoked.>                      4.6.4. Notification of New Certificate Issuance  
to Subscriber    <Name of Organization> will notify the subscriber when the    cer  
tificate is published. <Describe your procedure for notification    of new certifi  
cate issuance to the subscriber. This should be    consistent with 4.3.2.>                      4  
.6.5. Conduct Constituting Acceptance of a Renewal Certificate    See Section 4.4.  
1 <If you employ a different policy from that    specified in Section 4.4.1, descr  
ibe it here.>                      4.6.6. Publication of the Renewal Certificate by the CA    See  
Section 4.4.2.                      4.6.7. Notification of Certificate Issuance by the CA to Ot  
her    Entities    See Section 4.4.3.4.7. Certificate Re-key                      4.7.1. Ci  
rcumstance for Certificate Re-key    As per the RPKI CP, re-key of a certificate w  
ill be performed only    when required, based on:    1. knowledge or suspicion of c  
ompromise or loss of the associated    private key, or    2. the expiration of t  
he cryptographic lifetime of the associated key    pairKong, Seo & Kent  
Expires January 2013                      [Page 22]



If a certificate is revoked to replace the RFC 3779 extensions, the replacement certificate will incorporate the same public key, not a new key, unless the subscriber requests a re-key at the same time. If the re-key is based on a suspected compromise, then the previous certificate will be revoked. Section 5.6 of the Certificate Policy notes that when a CA signs a certificate, the signing key should have a validity period that exceeds the validity period of the certificate.

This places additional constraints on when a subscriber should request a re-key.

4.7.2. Who May Request Certification of a New Public Key Only the holder or a certificate may request a re-key. In addition, <Name of Organization> may initiate a re-key based on a verified compromise report. <If the subscriber (certificate Subject) requests the rekey, describe how authentication is effected, e.g., using the <Name of Registry> BPKI. Describe how a compromise report received from other than a subscriber is verified.>

4.7.3. Processing Certificate Re-keying Requests <Describe your process for handling re-keying requests. As per the RPKI CP, this should be consistent with the process described in Section 4.3. So reference can be made to that section.>

4.7.4. Notification of New Certificate Issuance to Subscriber <Describe your policy regarding notifying the subscriber re: availability of the new re-keyed certificate. This should be consistent with the notification process for any new certificate issuance (see Section 4.3.2).>

4.7.5. Conduct Constituting Acceptance of a Re-keyed Certificate When a re-keyed certificate is issued, the CA will publish it in the repository and notify the subscriber. See Section 4.4.1.

4.7.6. Publication of the Re-keyed Certificate by the CA <Describe your policy regarding publication of the new certificate. This should be consistent with the publication process for any new certificate (see Section 4.4.2).>

t

Expires January 2013

[Page 23]

Kong, Seo & Ken

4.7. Notification of Certificate Issuance by the CA to Other Entities See Section 4.4.3.4.8. Certificate Modification 4.8.1. Circumstance for Certificate Modification

As per the RPKI CP, modification of a certificate occurs to implement changes to the RFC 3779 extension values or the SIA extension in a certificate. A subscriber can request a certificate modification when this information in a currently valid certificate has changed, as a result of changes in the INR holdings of the subscriber or a change of the repository publication point data. If a subscriber is to receive a distribution of INRs in addition to a current distribution, and if the subscriber does not request that a new certificate be issued containing only these additional INRs, then this is accomplished through a certificate modification. When a certificate modification is approved, a new certificate is issued. The new certificate will contain the same public key and the same expiration date as the original certificate, but with the incidental information corrected and/or the INR distribution expanded. When previously distributed INRs are to be removed from a certificate, then the old certificate will be revoked and a new certificate (reflecting the new distribution) issued.

4.8.2. Who May Request Certificate modification The subscriber or <Name of Organization> may initiate the certificate modification process. <For the case of the subscriber, state here what steps will be taken to verify the identity and authorization of the entity requesting the modification.>

4.8.3. Processing Certificate Modification Requests <Describe your procedures for verification of the modification request and procedures for the issuance of a new certificate. These should be consistent with the processes described in Sections 4.2 and 4.3.1.>

4.8.4. Notification of Modified Certificate Issuance to Subscriber <Describe your procedure for notifying the subscriber about the issuance of a modified certificate. This should be consistent with

Kong, Seo & Kent Expires January 2013 [Page 24]

Internet-Draft                      Template CPS for the RPKI                      July 2012                      the not  
 ification process for any new certificate (see Section 4.3.2).>                      4.8.5. Co  
 nduct Constituting Acceptance of Modified Certificate                      When a modified certifica  
 te is issued, <Name of Organization> will                      publish it to the repository and noti  
 fy the subscriber. See Section 4.4.1.                      4.8.6. Publication of the Modified c  
 ertificate by the CA                      <Describe your procedure for publication of a modified c  
 ertificate. This should be consistent with the publication process for any new  
 certificate (see Section 4.4.2).>                      4.8.7. Notification of Certificate Issu  
 ance by the CA to Other                      Entities                      See Section 4.4.3.4.9. Certificate Re  
 vocation and Suspension                      4.9.1. Circumstances for Revocation                      As per the RP  
 KI CP, certificates can be revoked for several reasons. Either <Name of Organiz  
 ation> or the subject may choose to end the                      relationship expressed in the certi  
 ficate, thus creating cause to                      revoke the certificate. If one or more of the IN  
 Rs bound to the                      public key in the certificate are no longer associated with the  
 subject, that too constitutes a basis for revocation. A certificate                      also may  
 be revoked due to loss or compromise of the private key                      corresponding to the p  
 ublic key in the certificate. Finally, a                      certificate may be revoked in order t  
 o invalidate data signed by the                      private key associated with that certificate.  
 4.9.2. Who Can Request Revocation                      The subscriber or <Name of Organization>  
 may request a revocation.                      <For the case of the subscriber, describe what steps  
 will be taken to                      verify the identity and authorization of the entity requestin  
 g the                      revocation.>                      4.9.3. Procedure for Revocation Request                      <Describe yo  
 ur process for handling a certificate revocation request. This should include:K  
 ong, Seo & Kent                      Expires January 2013                      [Page 25]

Internet-Draft                      Template CPS for the RPKI                      July 2012      o    Proc  
edure to be used by the subscriber to request a revocation      o    Procedure for not  
ification of the subscriber when the revocation                      is initiated by <Name of Org  
anization>.>                      4.9.4. Revocation Request Grace Period      A subscriber is requi  
red request revocation as soon as possible after      the need for revocation has be  
en identified.                      4.9.5. Time Within Which CA Must Process the Revocation Requ  
est      <Describe your policy on the time period within which you will      process a  
revocation request.>                      4.9.6. Revocation Checking Requirement for Relying Par  
ties      As per the RPKI CP, a relying party is responsible for acquiring and      che  
cking the most recent, scheduled CRL from the issuer of the      certificate, whenev  
er the relying party validates a certificate.                      4.9.7. CRL Issuance Frequency  
    <State the CRL issuance frequency for the CRLs that you publish.>      Each CRL c  
ontains a nextScheduledUpdate value and a new CRL will be      published at or befor  
e that time. <Name of Organization> will set the      nextScheduledUpdate value when  
it issues a CRL, to signal when the      next scheduled CRL will be issued.                      4  
.9.8. Maximum Latency for CRLs      A CRL will be published to the repository system  
within <state the      maximum latency> after generation.4.10. Certificate Status S  
ervices      <Name of Organization> does not support OCSP or SCVP. <Name of      Organi  
zation> issues CRLs.Kong, Seo & Kent                      Expires January 2013

[Page 27]

Internet-Draft                      Template CPS for the RPKI                      July 2012                      5.3

5.1. Qualifications, experience, and clearance requirements                      5.3.2. Background check procedures                      5.3.3. Training requirements                      5.3.4. Retraining frequency and requirements                      5.3.5. Job rotation frequency and sequence                      5.3.6. Sanctions for unauthorized actions                      5.3.7. Independent contractor requirements                      5.3.8. Documentation supplied to personnel

5.4. Audit Logging Procedures                      <As per the CP, describe in the following sections the details of how you implement audit logging.>

5.4.1. Types of Events Recorded                      Audit records will be generated for the basic operations of the certification authority computing equipment. Audit records will include the date, time, responsible user or process, and summary content data relating to the event. Auditable events include:

- . Access to CA computing equipment (e.g., logon, logout)
- . Messages received requesting CA actions (e.g., certificate requests, certificate revocation requests, compromise notifications)
- . Certificate creation, modification, revocation, or renewal actions
- . Posting of any material to a repository
- . Any attempts to change or delete audit data

<List here any additional types of events that will be audited.>

5.4.2. Frequency of Processing Log                      <Describe your procedures for review of audit logs.>

Kong, Seo & Kent                      Expires January 2013                      [Page 28]

Internet-Draft                      Template CPS for the RPKI                      July 2012                      5.4

.3. Retention Period for Audit Log    <Describe your policies for retention of audit logs.>                      5.4.4. Protection of Audit Log    <Describe your policies for protection of the audit logs.>                      5.4.5. Audit Log Backup Procedures    <Describe your policies for backup of the audit logs.>                      5.4.6. Audit Collection System (Internal vs. External) [OMITTED]                      5.4.7. Notification to Event-causing Subject [OMITTED]                      5.4.8. Vulnerability Assessments    <Describe any vulnerability assessments that you will apply (or have already applied) to the PKI subsystems. This should include whether such assessments have taken place and any procedures or plans to perform or repeat/reassess vulnerabilities in the future.>5.5. Records archival [OMITTED]5.6. Key Changeover    The <Name of Organization> CA certificate will contain a validity period that is at least as long as that of any certificate being issued under that certificate. When <Name of Organization> CA changes keys it will follow the procedures described in [RFC6489].5.7. Compromise and disaster recovery [OMITTED]5.8. CA or RA Termination    <Describe your policy for management of your CA's INR distributions in case of its own termination.>Kong, Seo & Kent                      Expires January 2013                      [Page 29]

Internet-Draft                      Template CPS for the RPKI                      July 20126. Technical  
al Security Controls    This section describes the security controls used by <Name  
of    Organization>.6.1. Key Pair Generation and Installation                      6.1.1. Key Pa  
ir Generation    <Describe the procedures used to generate the CA key pair, and, i  
f    applicable, key pairs for subscribers. In most instances, public-key    pairs  
will be generated by the subscriber, i.e., the organization    receiving the dist  
ribution of INRs. However, your procedures may    include one for generating key  
pairs on behalf of your subscribers if    they so request.>                      6.1.2. Private K  
ey Delivery to Subscriber    <If the procedures in 6.1.1 include providing key pai  
r generation    services for subscribers, describe the means by which private keys  
are delivered to subscribers in a secure fashion. Otherwise say this    is not  
applicable.>                      6.1.3. Public Key Delivery to Certificate Issuer    <Describe t  
he procedures that will be used to deliver a subscriber's    public keys to the <N  
ame of Organization> RPKI CA. These procedures    should ensure that the public k  
ey has not been altered during transit    and that the subscriber possesses the pr  
ivate key corresponding to    the transferred public key. >                      6.1.4. CA Public  
Key Delivery to Relying Parties    CA public keys for all entities (other than tr  
ust anchors) are    contained in certificates issued by other CAs and will be publ  
ished    to the RPKI repository system. Relying parties will download these    cert  
ificates from this system. Public key values and associated data    for (putative)  
trust anchors will be distributed out of band and    accepted by relying parties  
on the basis of locally-defined criteria,    e.g., embedded in path validation sof  
tware that will be made    available to the Internet community.                      6.1.5. Key S  
izes    The key sizes used in this PKI are as specified in [RFC6485].Kong, Seo & K  
ent                      Expires January 2013                      [Page 30]



6. Public Key Parameters Generation and Quality Checking The public key algorithms and parameters used in this PKI are as specified in [RFC6485]. <If the procedures in 6.1.1 include subscriber key pair generation, EITHER insert here text specifying that the subscriber is responsible for performing checks on the quality of its key pair and saying that <Name of Organization> is not responsible for performing such checks for subscribers OR describe the procedures used by the CA for checking the quality of these subscriber key pairs.> 6.1.7. Key Usage Purposes (as per X.509 v3 Key Usage Field) The Key usage extension bit values employed in RPKI certificates are specified in [RFC6487]. 6.2. Private Key Protection and Cryptographic Module Engineering Controls 6.2.1. Cryptographic module standards and controls <Describe the standards and controls employed for the CA cryptographic module, e.g., it was evaluated under FIPS 140-2/3, at level 2 or 3 [FIPS]>. 6.2.2. Private Key (n out of m) Multi-Person Control <If you choose to use multi-person controls to constrain access to your CA's private keys, then insert the following text. "There will be private key <insert here n> out of <insert here m> multi-person control."> 6.2.3. Private Key Escrow <No private key escrow procedures are required for the RPKI, but if the CA chooses to employ escrow, state so here.> 6.2.4. Private Key Backup <Describe the procedures used for backing up your CA's private key. The following aspects should be included. (1) The copying should be done under the same multi-party control as is used for controlling the original private key. (2) At least one copy should be kept at an off-site location for disaster recovery purposes.>Kong, Seo & Kent Expires January 2013 [Page 31]

Internet-Draft                      Template CPS for the RPKI                      July 2012                      6.2

6.2.5. Private Key Archival      See sections 6.2.3 and 6.2.4                      6.2.6. Private Key Transfer into or from a Cryptographic Module      The private key for <Name of Organization>'s production CA <if appropriate, change "production CA" to "production and offline CAs"> will be generated by the cryptographic module specified in 6.2.1.      The private keys will never leave the module except in encrypted form for backup and/or transfer to a new module.                      6.2.7. Private Key Storage on Cryptographic Module      The private key for <Name of Organization>'s production CA <if appropriate, change "production CA" to "production and offline CAs"> will be stored in the cryptographic module. It will be protected from unauthorized use <say how here>.                      6.2.8. Method of Activating Private Key      <Describe the mechanisms and data used to activate your CA's private key.>                      6.2.9. Method of Deactivating Private Key      <Describe process and procedure for private key deactivation here.>.                      6.2.10. Method of Destroying Private Key      <Describe the method used for destroying your CA's private key, e.g., when it is superseded. This will depend on the particular module.>                      6.2.11. Cryptographic Module Rating      <Describe the rating of the cryptographic module used by the CA, if applicable.>6.3. Other aspects of Key Pair Management                      6.3.1. Public Key Archival      <Because this PKI does not support non-repudiation, there is no need to archive public keys. If keys are not archived, say so. If they are, describe the archive processes and procedures.>Kong, Seo & Kent                      Expires January 2013

Internet-Draft                      Template CPS for the RPKI                      July 2012                      6.3

6.2. Certificate Operational Periods and Key Pair Usage                      Periods                      The <Name of Organization> CA's key pair will have a validity interval of <insert number of years - These key pairs and certificates should have reasonably long validity intervals, e.g., 10 years, to minimize the disruption caused by key change over.>

6.4. Activation data                      6.4.1. Activation Data Generation and Installation                      <Describe how activation data for your CA will be generated.>                      6.4.2. Activation data protection                      Activation data for the CA private key will be protected by <Describe your procedures here>.

6.4.3. Other Aspects of Activation Data                      <Add here any details you wish to provide with regard to the activation data for your CA. If there are none, say "None.">

6.5. Computer Security Controls                      <Describe your security requirements for the computers used to support this PKI, e.g., requirements for authenticated logins, audit capabilities, etc. These requirements should be commensurate with those used for the computers used for managing distribution of INRs.>

6.6. Life cycle Technical Controls                      6.6.1. System Development Controls                      <Describe any system development controls that apply to the PKI systems, e.g., use of Trusted System Development Methodology (TS DM).>

6.6.2. Security Management Controls                      <Describe the security management controls that will be used for the RPKI software and equipment employed by the CA. These security measures should be commensurate with those used for the systems used by the CAs for managing and distributing INRs.>

Kong, Seo & Kent  
Expires January 2013                      [Page 33]

Internet-Draft                      Template CPS for the RPKI                      July 2012                      6.6

.3. Life Cycle Security Controls    <Describe how the equipment (hardware and software) used for RPKI functions will be procured, installed, maintained, and updated. This should be done in a fashion commensurate with the way in which equipment for the management and distribution of INRs is handled. >6.7. Network Security Controls    <Describe the network security controls that will be used for CA operation. These should be commensurate with the network security controls employed for the computers used for managing distribution of INRs.>6.8. Time-stamping    The RPKI does not make use of time stamping.

Kong, Seo & Kent                      Expires January 2013                      [Page 34]



Internet-Draft                      Template CPS for the RPKI                      July 20128. Compliance Audit and Other Assessments      <List here any audit and other assessments used to ensure the security of the administration of INRs. These are sufficient for the RPKI systems.>Kong, Seo & Kent                      Expires January 2013  
[Page 36]

Internet-Draft                      Template CPS for the RPKI                      July 20129. Other B  
usiness And Legal Matters      <The sections below are optional. Fill them in as app  
ropriate for    your organization. The CP says that CAs should cover 9.1 to 9.11 a  
nd    9.13 to 9.17 although not every CA will choose to do so. Note that    the man  
ner in which you manage your business and legal matters for    this PKI should be  
commensurate with the way in which you manage    business and legal matters for th  
e distribution of INRs.>Kong, Seo & Kent                      Expires January 2013  
[Page 37]

Internet-Draft                      Template CPS for the RPKI                      July 2012

9.1.1. Certificate issuance or renewal fees                      9.1.2. Fees for other services (if applicable)                      9.1.3. Refund policy

9.2.1. Insurance coverage                      9.2.2. Other assets                      9.2.3. Insurance or warranty coverage for end-entities

9.3.1. Scope of confidential information                      9.3.2. Information not within the scope of confidential information                      9.3.3. Responsibility to protect confidential information

9.4. Privacy of personal information                      9.4.1. Privacy plan                      9.4.2. Information treated as private                      9.4.3. Information not deemed private                      9.4.4. Responsibility to protect private information                      9.4.5. Notice and consent to use private information                      9.4.6. Disclosure pursuant to judicial or administrative process                      9.4.7. Other information disclosure circumstances

9.5. Intellectual property rights (if applicable)

9.6. Representations and warranties                      9.6.1. CA representations and warranties

Kent                      Expires January 2013                      [Page 38]



Internet-Draft	Template CPS for the RPKI	July 2012	9.6
.2. Subscriber representations and warranties		9.6.3. Relying party representations and warranties	
9.9. Indemnities	9.7. Disclaimers of warranties	9.8. Limitations of liability	
9.10. Term and termination	9.10.1. Term	9.10.2. Termination	
9.10.3. Effect of termination and survival	9.11. Individual notices and communications with participants	9.12. Amendments	
9.12.1. Procedure for a amendment	9.12.2. Notification mechanism and period	9.13. Dispute resolution provisions	
9.14. Governing law	9.15. Compliance with applicable law	9.16. Miscellaneous provisions	
9.16.1. Entire agreement	9.16.2. Assignment	9.16.3. Severability	
9.16.4. Enforcement (attorneys' fees and waiver of rights)	9.16.5. Force Majeure	10. Security Considerations	
Kong, Seo & Kent	E		
Expires January 2013	[Page 39]		

Internet-Draft                      Template CPS for the RPKI                      July 2012      The degree to which a relying party can trust the binding embodied in a certificate depends on several factors. These factors can include the practices followed by the certification authority (CA) in authenticating the subject; the CA's operating policy, procedures, and technical security controls, including the scope of the subscriber's responsibilities (for example, in protecting the private key), and the stated responsibilities and liability terms and conditions of the CA (for example, warranties, disclaimers of warranties, and limitations of liability). This document provides a framework to address the technical, procedural, personnel, and physical security aspects of Certification Authorities, Registration Authorities, repositories, subscribers, and relying party cryptographic modules, in order to ensure that the certificate generation, publication, renewal, re-key, usage, and revocation is done in a secure manner. Specifically, Section 3 Identification and Authentication (I&A); Section 4 Certificate Lifecycle Operational Requirements; Section 5 Facility Management, and Operational Controls; Section 6 Technical Security Controls; Section 7 Certificate and CRL Profiles; and Section 8 Compliance Audit and Other Assessments are oriented towards ensuring secure operation of the PKI entities such as CA, RA, repository, subscriber systems, and relying party systems.11. IANA Considerations None.12. Acknowledgments The authors would like to thank Matt Lepinski for help with the formatting, Ron Watro for assistance with the editing, and other members of the SIDR working group for reviewing this document.Kong, Seo & Kent                      Expires January 2013                      [Page 40]

Internet-Draft                      Template CPS for the RPKI                      July 2012

13.1. Normative References                      [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.                      [RFC3280] Housley, R., Polk, W. Ford, W., Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", BCP 14, RFC 2119, March 1997.                      [RFC6484] Kent, S., Kong, D., Seo, K., and Watro, R., "Certificate Policy (CP) for the Resource PKI (RPKI)," February 2012.                      [RFC6487] Huston, G., Michaelson, G., and Loomans, R., "A Profile for X.509 PKIX Resource Certificates," February 2012.                      [RFC6485] Huston, G., "A Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure," February 2012.

13.2. Informative References                      [BGP4] Y. Rekhter, T. Li (editors), A Border Gateway Protocol 4 (BGP-4). IETF RFC 1771, March 1995.                      [FIPS] Federal Information Processing Standards Publication 140-3 (FIPS-140-3), "Security Requirements for Cryptographic Modules", Information Technology Laboratory, National Institute of Standards and Technology, work in progress.                      [RSA] Rivest, R., Shamir, A., and Adelman, L. M. 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 2 (Feb.), 120-126.

Kong, Seo & Kent                      Expires January 2013

Internet-Draft                      Template CPS for the RPKI                      July 2012Author's A  
ddresses   Stephen Kent   BBN Technologies   10 Moulton Street   Cambridge MA 021  
38   USA   Phone: +1 (617) 873-3988   Email: skent@bbn.com   Derrick Kong   BBN T  
echnologies   10 Moulton Street   Cambridge MA 02138   USA   Phone: +1 (617) 873-  
1951   Email: dkong@bbn.com   Karen Seo   BBN Technologies   10 Moulton Street  
Cambridge MA 02138   USA   Phone: +1 (617) 873-3152   Email: kseo@bbn.comCopyrigh  
t Statement   Copyright (c) 2012 IETF Trust and the persons identified as the   d  
ocument authors.   All rights reserved.   This document is subject to BCP 78 and t  
he IETF Trust's Legal   Provisions Relating to IETF Documents   (<http://trustee.ietf.org/license-info>) in effect on the date of   publication of this document.   P  
lease review these documents   carefully, as they describe your rights and restri  
ctions with respect   to this document. Code Components extracted from this docum  
ent must   include Simplified BSD License text as described in Section 4.e of   t  
he Trust Legal Provisions and are provided without warranty as   described in the  
Simplified BSD License.Kong, Seo & Kent   Expires January 2013  
[Page 42]

Secure Inter-Domain Routing  
Internet-Draft  
Intended status: Standards Track  
Expires: December 6, 2012

M. Reynolds  
IPSw  
S. Kent  
BBN  
M. Lepinski  
BBN  
June 4, 2012

Local Trust Anchor Management for the Resource Public Key Infrastructure  
<draft-ietf-sidr-ltamgmt-05.txt>

#### Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on December 6, 2012.

#### Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

#### Abstract

This document describes a facility to enable a relying party (RP) to manage trust anchors (TAs) in the context of the Resource Public Key Infrastructure (RPKI). It is common to allow an RP to import TA material in the form of self-signed certificates. The facility described in this document allows an RP to impose constraints on such TAs. Because this mechanism is designed to operate in the RPKI context, the relevant constraints are the RFC 3779 extensions that bind address spaces and/or autonomous system (AS) numbers to entities. The primary motivation for this facility is to enable an RP to ensure that resource allocation information that it has acquired via some trusted channel is not overridden by the information acquired from the RPKI repository system or by the putative TAs that the RP imports. Specifically, the mechanism allows an RP to specify a set of bindings between public key identifiers and RFC 3779 extension data and will override any conflicting bindings expressed via the putative TAs and the certificates downloaded from the RPKI repository system. Although this mechanism is designed for local use by an RP, an entity that is accorded administrative control over a set of RPs may use this mechanism to convey its view of the RPKI to a set of RPs within its jurisdiction. The means by which this latter use case is effected is outside the scope of this document.

## Table of Contents

1	Introduction . . . . .	4
1.1	Terminology . . . . .	5
2	Overview of Certificate Processing . . . . .	5
2.1	Target Certificate Processing . . . . .	5
2.2	Perforation . . . . .	5
2.3	TA Re-parenting . . . . .	6
2.4	Paracertificates . . . . .	6
3	Format of the constraints file . . . . .	8
3.1	Relying party subsection . . . . .	8
3.2	Flags subsection . . . . .	8
3.3	Tags subsection . . . . .	9
3.3.1	Xvalidity_dates tag . . . . .	10
3.3.2	Xcrl_dp tag . . . . .	10
3.3.3	Xcp tag . . . . .	11
3.3.4	Xaia tag . . . . .	11
3.4	Blocks subsection . . . . .	12
4	Certificate Processing Algorithm . . . . .	13
4.1	Proofreading algorithm . . . . .	14
4.2	TA processing algorithm . . . . .	15
4.2.1	Preparatory processing (stage 0) . . . . .	16
4.2.2	Target processing (stage 1) . . . . .	17
4.2.3	Ancestor processing (stage 2) . . . . .	18
4.2.4	Tree processing (stage 3) . . . . .	19
4.2.5	TA re-parenting (stage 4) . . . . .	20
4.3	Discussion . . . . .	21
5	Implications for Path Discovery . . . . .	21
5.1	Two answers . . . . .	21
5.2	One answer . . . . .	22
5.3	No answer . . . . .	22
6	Implications for Revocation . . . . .	22
6.1	No state bits set . . . . .	23
6.2	ORIGINAL state bit set . . . . .	23
6.3	PARA state bit set . . . . .	23
6.4	Both ORIGINAL and PARA state bits set . . . . .	24
7	Security Considerations . . . . .	24
8	IANA Considerations . . . . .	24
9	Acknowledgements . . . . .	24
10	References . . . . .	24
10.1	Normative References . . . . .	24
10.2	Informative References . . . . .	25
	Authors' Addresses . . . . .	25
	Appendix A: Sample Constraints File . . . . .	26
	Appendix B: Optional Sorting Algorithm for Ancestor Processing . . . . .	27

## 1 Introduction

The Resource Public Key Infrastructure (RPKI) [I-D.sidr-arch] is a PKI in which certificates are issued to facilitate management of IP addresses and autonomous system number resources. Such resources are expressed in the form of X.509v3 "resource" certificates with extensions as defined by RFC 3779 [I-D.sidr-res-cert-prof]. Validation of a resource certificate is preceded by path discovery. Path discovery is effected by constructing a certificate path (upward) from a target certificate to a trust anchor. Path validation proceeds from the TA in question to the target certificate, using the public key from each certificate along the path to verify the signature of its subordinate certificate. In the RPKI it is anticipated that one or more putative TAs, aligned with the resource allocation hierarchy, will be available in the form of self-signed certificates configured by an RP. There are circumstances under which an RP may wish to override the resource specifications obtained through the RPKI distributed repository system [I-D.sidr-repos-struct]. This document describes a mechanism by which an RP may override any conflicting information expressed via the putative TAs and the certificates downloaded from the RPKI repository system.

To effect this local control, this document calls for a relying party to specify a set of bindings between public key identifiers and resources (IP resources and/or AS number resources) through a text file known as a constraints file. The constraints expressed in this file then take precedence over any competing claims expressed by resource certificates acquired from the distributed repository system. (The means by which a relying party acquires the key identifier and the RFC 3779 extension data used to populate the constraints file is outside the scope of this document.) The relying party also may use a local publication point (the root of a local directory tree that is made available as if it were a remote repository) as a source of certificates and CRLs (and other RPKI signed objects, e.g. ROAs and manifests) that do not appear in the RPKI repository system.

In order to allow reuse of existing, standard path validation mechanisms, the RP-imposed constraints are realized by having the RP itself represented as the only TA known in the local certificate validation context. To ensure that all RPKI certificates can be validated relative to this TA, this RP TA certificate must contain all-encompassing resource allocations, i.e. 0/0 for IPv4, 0::/0 for IPv6 and 0-4294967295 for AS numbers. Thus, a conforming implementation of this mechanism must be able to cause a self-signed certification authority (CA) certificate to be created with a locally generated key pair. It also must be able to issue CA certificates subordinate to this TA. Finally, a conforming implementation of this



mechanism must process the constraints file and modify certificates as needed in order to enforce the constraints asserted in the file.

The remainder of this document describes in detail the types of certificate modification that may occur, the semantics of the constraints file, and the implications of certificate modification on path discovery and revocation.

## 1.1 Terminology

It is assumed that the reader is familiar with the terms and concepts described in "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" [RFC5280] and "X.509 Extensions for IP Addresses and AS Identifiers" [RFC3779].

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

## 2 Overview of Certificate Processing

The fundamental aspect of the facility described in this document is one of certificate modification. The constraints file, described in more detail in the next section, contains assertions about resources that are to be specially processed. As a result of this processing, certificates in the local copy of the RPKI repository are transformed into new certificates satisfying the resource constraints so specified. This enables the RP to override conflicting assertions about resource holdings as acquired from the RPKI repository system. Three forms of certificate modification can occur.

### 2.1 Target Certificate Processing

If a certificate is acquired from the RPKI repository system and its SKI is listed in the constraints file, it will be reissued directly under the RP TA certificate, with (possibly) modified RFC 3779 extensions. The modified extensions will include any RFC 3779 data expressed in the constraints file. In Section 4.2, target certificate processing corresponds to stage one of the algorithm.

### 2.2 Perforation

Any certificate acquired from the RPKI repository that contains an RFC 3779 extension that intersects the resource data in the constraints file will be reissued directly under the RP TA, with modified RFC 3779 extensions. We refer to the process of modifying the RFC 3779 extension in an affected certificate as "perforation" (because the process will create "holes" in these extensions). The

modified extensions will exclude any RFC 3779 data expressed in the constraints file. In the certificate processing algorithm described in Section 4.2, perforation corresponds to stage two of the algorithm ("ancestor processing") and also to stage three of the algorithm ("tree processing").

### 2.3 TA Re-parenting

For consistency, all valid, self-signed certificates that would have been regarded as TAs in the public RPKI certificate hierarchy, e.g. self-signed certificates issued by IANA or the RIRs, will be re-issued under the RP TA certificate. This processing is done even though all but one of these certificates might not intersect any resources specified in the constraints file. We refer to this reissuance as "re-parenting" since the Issuer (parent) of the certificate has been changed. In the certificate processing algorithm described in Section 4.2, TA re-parenting corresponds to stage four of the algorithm.

### 2.4 Paracertificates

If a certificate is subject to any of the three forms of processing just described, that certificate will be referred to as an "original" certificate and the processed (output) certificate will be referred to as a paracertificate. When an original certificate is transformed into a paracertificate all the fields and extensions from the original certificate will be retained, except as indicated in Table 1, below.

Original Certificate Field	Action
Version	unchanged
Serial number	created per note A
Signature	replaced if needed with RP's signing alg
Issuer	replaced with RP's name
Validity dates	replaced per note B
Subject	unchanged
Subject public key info	unchanged
Extensions	
Subject key identifier	unchanged
Key usage	unchanged
Basic constraints	unchanged
CRL distribution points	replaced per note B
Certificate policy	replaced per note B
Authority info access	replaced per note B
Authority key ident	replaced with RP's
IP address block	modified as described
AS number block	modified as described
Subject info access	unchanged
All other extensions	unchanged
Signature Algorithm	same as above
Signature value	new

Table 1 Certificate Field Modifications

Note A. The serial number will be created by concatenating the current time (the number of seconds since Jan 1, 1970) with a count of the certificates created in the current run.

Note B. These fields are derived (as described in section 3.3 below) from parameters in the constraints file (if present); otherwise, they take on values from the certificates from which the paracertificates are derived.

### 3 Format of the constraints file

This section describes a general model for the syntax of the constraints file. The model described below is nominal; implementations need not match details of this model as presented, but the external behavior of implementations MUST correspond to the externally observable characteristics of this model in order to be compliant.

The constraints file consists of four logical subsections: the relying party subsection, the flags subsection, the tags subsection and the blocks subsection. The relying party subsection and the blocks subsection are REQUIRED and MUST be present; the flags and tags subsections are OPTIONAL. Each subsection is described in more detail below. Note that the semicolon (;) character acts as the comment character, to enable annotating constraints files. All characters from a semicolon to the end of that line are ignored. In addition, lines consisting only of whitespace are ignored. The subsections MUST occur in the order indicated. An example constraints file is given in Appendix A.

#### 3.1 Relying party subsection

The relying party subsection is a REQUIRED subsection of the constraints file. It MUST be the first subsection of the constraints file, and it MUST consist of two lines of the form:

```
PRIVATEKEYMETHOD      value [ ... value ]
TOPLEVELCERTIFICATE   value
```

The first line provides guidance to the certificate processing algorithm on the method that will be used to gain access to the RP's private key. This line consists of the string literal PRIVATEKEYMETHOD, followed by one or more whitespace delimited string values. These values are passed to the certificate processing algorithm as described below. Note that this entry, as for all entries in the constraints file, is case sensitive.

The second line of this subsection consists of the string literal TOPLEVELCERTIFICATE, followed by exactly one string value. This value is the name of a file containing the relying party's TA certificate. The file name is passed to the certificate processing algorithm as described below.

#### 3.2 Flags subsection

The flags subsection of the constraints file is an OPTIONAL subsection. If present it MUST immediately follow the relying party

subsection. The flags subsection consists of one or more lines of the form

```
CONTROL  flagname  booleanvalue
```

Each such line is referred to as a control line. Each control line MUST contain exactly three whitespace delimited strings. The first string MUST be the literal CONTROL. The second string MUST be one of the following three literals:

```
resource_nounion
intersection_always
treegrowth
```

The third string denotes a boolean value, and MUST be one of the literals TRUE or FALSE. Control flags influence the global operation of the certificate processing algorithm; the semantics of the flags is described in detail in Section 4.2. Note that each flag has a default value, so that if the corresponding CONTROL line does not appear in the constraints file, the algorithm flag is considered to take the corresponding default value. The default value for each flag is FALSE. Thus, if any flag is not named in a control line it takes the value FALSE. Further, if the flags subsection is absent, all three flags take the value FALSE.

### 3.3 Tags subsection

The tags subsection is an OPTIONAL subsection in the constraints file. If present it MUST immediately follow the relying party subsection (if the flags subsection is absent) or the flags subsection (if it is present). The tags subsection consists of one or more lines of the form

```
TAG  tagname  tagvalue [ ... tagvalue ]
```

Each such line is referred to as a tag line. Each tag line MUST consist of at least three whitespace delimited string values, the first of which must be the literal TAG. The second string value gives the name of the tag, and subsequent string(s) give the value(s) of the tag. The tag name MUST be one of the following four string literals:

```
Xvalidity_dates
Xcrldp
Xcp
Xaia
```

The purpose of the tag lines is to provide an indication of the means

by which paracertificate fields, specifically those indicated above under "Note B", are constructed. Each tag has a default, so that if the corresponding tag line is not present in the constraints file, the default behavior is used when constructing the paracertificates. The syntax and semantics of each tag line is described next.

Note that the tag lines are considered to be global; the action of each tag line (or the default action, if that tag line is not present) applies to all paracertificates that are created as part of the certificate processing algorithm.

### 3.3.1 Xvalidity\_dates tag

This tag line is used to control the value of the notBefore and notAfter fields in paracertificates. If this tag line is specified and there is a single tagvalue which is the literal string C, the paracertificate validity interval is copied from the original certificate validity interval from which it is derived. If this tag is specified and there is a single tagvalue which is the literal string R, the paracertificate validity interval is copied from the validity interval of the relying party's top level (TA) certificate. If this tag is specified and the tagvalue is neither of these literals, then exactly two tagvalues MUST be specified. Each must be a Generalized Time string of the form YYYYMMDDHHMMSSZ. The first tagvalue is assigned to the notBefore field and the second tagvalue is assigned to the notAfter field. It MUST be the case that the tagvalues may be parsed as valid Generalized Time strings such that notBefore is less than notAfter, and also such that notAfter represents a time in the future (i.e., the paracertificate has not already expired).

If this tag line is not present in the constraints file the default behavior is to copy the validity interval from the original certificate to the corresponding paracertificate.

### 3.3.2 Xcrl\_dp tag

This tag line is used to control the value of the CRL distribution point extension in paracertificates. If this tag line is specified and there is a single tagvalue that is the string literal C, the CRLDP of the paracertificate is copied from the CRLDP of the original certificate from which it is derived. If this tag line is specified and there is a single tagvalue that is the string literal R, the CRLDP of the paracertificate is copied from the CRLDP of the RP's top level certificate. If this tag line is specified and there is a single tagvalue that is not one of these two reserved literals, or if there is more than one tagvalue, then each tagvalue is interpreted as a URI that will be placed in the CRLDP sequence in the

paracertificate.

If this tag line is not present in the constraints file the default behavior is to copy the CRLDP from the original certificate into the corresponding paracertificate.

### 3.3.3 Xcp tag

This tag line is used to control the value of the policyQualifierId field in paracertificates. If this tag line is specified there MUST be exactly one tagvalue. If the tagvalue is the string literal C, the paracertificate value is copied from the value in the corresponding original certificate. If the tagvalue is the string literal R, the paracertificate value is copied from the value in the RP's top level TA certificate. If the tagvalue is the string literal D, the paracertificate value is set to the default OID. If the tagvalue is not one of these reserved string literals, then the tagvalue MUST be an OID specified using the standard dotted notation. The value in the paracertificate's policyQualifierId field is set to this OID. Note the RFC 5280 specifies that only a single policy may be specified in a certificate, so only a single tagvalue is permitted in this tag line, even though the CertificatePolicy field is an ASN.1 sequence.

If this tag line is not specified the default behavior is to use the default OID in creating the paracertificate.

This option permits the RP to convert a value of the policyQualifierId field in a certificate (that would not be in conformance with the RPKI CP) to a conforming value in the paracertificate. This conversion enables use of RPKI validation software that checks the policy field against that specified in the RPKI CP [ID.sidr-res-cert-prof].

### 3.3.4 Xaia tag

This tag line is used to control the value of the Authority Information Access (AIA) extension in the paracertificate. If this tag line is present then it MUST have exactly one tagvalue. If this tagvalue is the string literal C, then the AIA field in the paracertificate is copied from the AIA field in the original certificate from which it is derived. If this tag line is present and the tagvalue is not the reserved string literal, then the tagvalue MUST be a URI. This URI is set as the AIA extension of the paracertificates that are created.

If this tag line is not specified the default behavior is to use copy the AIA field from the original certificate to the AIA field of the paracertificate.

### 3.4 Blocks subsection

The blocks subsection is a REQUIRED subsection of the constraints file. If the tags subsection is present, the blocks subsection MUST appear immediately after it. If the tags subsection is absent, but the flags subsection is present, the block subsection MUST appear immediately after it. Otherwise, the blocks subsection MUST appear immediately after the relying party subsection. The blocks subsection consists of one or more blocks, known as target blocks. A target block is used to specify an association between a certificate (given by a hash of its public key information) and a set of resource assertions. Each target block contains four regions, an SKI region, an IPv4 region, an IPv6 region and an AS number region. All regions are REQUIRED to be present in a target block.

The SKI region contains a single line beginning with the string literal SKI and followed by forty hexadecimal characters giving the subject key identifier of a certificate, known as the target certificate. The hex character string MAY contain embedded whitespace or colon characters (included to improve readability), which are ignored. The IPv4 region consists of a line containing only the string literal IPv4. This line is followed by zero or more lines containing IPv4 prefixes in the format described in RFC 3779. The IPv6 region consists of a line containing only the string literal IPv6, followed by zero or more lines containing IPv6 prefixes using the format described in RFC 3513. (The presence of the IPv4 and IPv6 literals is to simplify parsing of the constraints file.) Finally, the AS number region consists of a line containing only the string literal AS#, followed by zero or more lines containing AS numbers (one per line). The AS numbers are specified in decimal notation as recommended in RFC 5396. A target block is terminated by either the end of the constraints file, or by the beginning of the next target block, as signaled by its opening SKI region line. An example target block is shown below. See also the complete constraints file example given in Appendix A. Note that whitespace, as always, is ignored.

```
SKI 00:12:33:44:00:BA:BA:DE:EB:EE:00:99:88:77:66:55:44:33:22:11
IPv4
  10.2.3/24
  10.8/16
IPv6
  1:2:3:4:5:6/112
AS#
  123
  567
```

The blocks subsection MUST contain at least one target block. Note that it is OPTIONAL that the SKI refer to a certificate that is known



or resolvable within the context of the local RPKI repository. Also, there is no REQUIRED or implied ordering of target blocks within the block subsection. As a result of the fact that blocks may occur in any order, it MAY result that the outcome of processing a constraints file depends on the order in which target blocks occur within the constraints file. The next section of this document contains a detailed description of the certificate processing algorithm.

#### 4 Certificate Processing Algorithm

The section describes the certificate processing algorithm through which paracertificates are created from original certificates in the local RPKI repository. For the purposes of describing this algorithm, it will be assumed that certificates may be persistently associated with state (or metadata) information. This state information will be further construed as having the form of any array of named bits that are associated with each certificate. No specific implementation of this functionality is mandated by this document. Any implementation that provides the indicated functionality is acceptable, and need not actually consist of a bit field associated with each certificate.

The state bits used in certificate processing are

NOCHAIN  
ORIGINAL  
PARA  
TARGET

If the NOCHAIN bit is set, this indicates that a full path between the given certificate and a TA has not yet been discovered. If the ORIGINAL bit is set, this indicates that the certificate in question has been processed by some part of the processing algorithm described in Section 4.2. If it was processed as part of stage one processing, as described in section 4.2.2, the TARGET bit will also be set. Finally, any paracertificate will have the PARA bit set.

At the beginning of algorithm processing each certificate in the local RPKI repository has the ORIGINAL, PARA and TARGET bits clear. If a certificate has a complete, validated path to a TA, or is itself a TA, then that certificate will have the NOCHAIN bit clear, otherwise it will have the NOCHAIN bit set. As the certificate processing algorithm is executed, the bit state of original certificates may change. In addition, since the certificate processing algorithm may also be creating paracertificates, it is responsible for actively setting or clearing the state of these four bits on those paracertificates.

The certificate processing algorithm consists of two sub-algorithms:

"proofreading" and "TA processing". Conceptually, the proofreading sub-algorithm performs syntactic checks on the constraints file, while the TA processing sub-algorithm performs the actual certificate transformation processing. If the proofreading sub-algorithm does not succeed in parsing the constraints file, the TA processing sub-algorithm is not executed. Note also that if the constraints file is not present, neither sub-algorithm is executed and the local RPKI repository is not modified. Each of the constituent algorithms will now be described in detail.

#### 4.1 Proofreading algorithm

The goal of the proofreading algorithm is to check the constraints file for syntactic errors, such as missing REQUIRED subsections, or malformed addresses such as 1.2.300/24. It also performs a set of heuristic checks, such as checking for prefixes that are too large (larger than /8). The proofreading algorithm SHOULD also examine resource regions (IPv4, IPv6 and AS# regions) within the blocks subsection, and reorder such resources within a region in ascending numeric order. On encountering any error the proofreading algorithm SHOULD provide an error message indicating the line on which the error occurred as well as informative text that is sufficiently descriptive as to allow the user to identify and correct the error. An implementation of the proofreading algorithm MUST NOT assume that it has access to the local RPKI repository (even read-only access). An implementation of the proofreading algorithm MUST NOT alter the local RPKI repository in any way; it also MUST NOT change any of the state/metadata information associated with certificates in that repository. (Recall that the processing described here is creating a copy of that local repository.) Finally, the proofreading algorithm MAY produce a transformed output file containing the same syntactic information as in the text version of the constraints file, so long as the format of the transformed file is understood by the TA processing algorithm.

The proofreading algorithm performs the following syntactic checks on the constraints file. It checks for the presence of the REQUIRED relying party subsection and the REQUIRED blocks subsection. It checks that the order of the two, three or four subsections is as stated above. It checks that the relying party subsection conforms to the specification given in section 3.1 above. If present, it checks that the tags and flags subsections conform to the specifications in sections 3.2 and 3.3 above. It then checks the blocks subsection. It splits the blocks subsection into constituent target blocks, as delimited by the SKI region line(s), and verifies that at least one target block is present. It verifies that each SKI region line contains exactly forty hexadecimal digits and contains no additional characters other than whitespace or colon characters. For each target

block, it then verifies the presence of the IPv4, IPv6 and AS# regions, and also verifies that at least one such resource is present. For each IPv4 prefix, IPv6 prefix and autonomous system number given, it checks that the indicated resource is syntactically valid according to the appropriate RFC definition, as described in section 3.4. It also verifies that no IPv4 resource has a prefix larger than /8. The proofreading algorithm SHOULD performing reordering within each of the three resource regions so that stated resources occur in ascending numerical order. If the proofreading algorithm has performed any reordering of information it MAY overwrite the constraints file. If it does so, however, it MUST preserve all information contained within the file, including information that is not parsed (such as comments). If the proofreading algorithm has performed any reordering of information but has not overwritten the constraints file, it MAY produce a transformed output file, as described above. If the proofreading algorithm has performed any reordering of information, but has neither overwritten the constraints file nor produced a transformed output file, it MUST provide an error message to the user indicating what reordering was performed.

#### 4.2 TA processing algorithm

The TA processing algorithm acts on the constraints file (or the output file produced by the proofreading algorithm) and the contents of the local RPKI repository to produce paracertificates for the purpose of enforcing the resource allocations as expressed in the constraints file. The TA processing algorithm operates in five stages, a preparatory stage (stage 0), target processing (stage 1), ancestor processing (stage 2), tree processing (stage 3) and TA re-parenting (stage 4). Conceptually, during the preparatory stage the constraints (or proofreader output) file is read and a set of internal RP, tag and flag variables are set based on the contents of that file. (If the constraint file has not specified one or more of the tags and/or flags, those tags and flags are set to default values.) During target processing all certificates specified by a target block are processed, and the resources for those certificates are (potentially) expanded; for each target found a new paracertificate is manufactured with its various fields set, as shown in Table 1, using the values of the internal variables set in the preparatory stage and also, of course, the fields of the original certificate (and, potentially, fields of the RP's TA certificate). In stage 2 (ancestor) processing, all ancestors of the each target certificate are found, and the claimed resources are then removed (perforated). A new paracertificate with these diminished resources is crafted, with its fields generated based on internal variable settings, original certificate field values, and, potentially, the fields of the RP's TA certificate. In tree processing (stage 3), the

entire local RPKI repository is searching for any other certificates that have resources that intersect a target resource, and that were not otherwise processed during a preceding stage. Perforation is again performed for any such intersecting certificates, and paracertificates created as in stage 2. Finally, in the fourth and last stage, TA re-parenting, any TA certificates in the local RPKI repository that have not already been processed are now re-parented under the RP's TA certificate. This transformation will create paracertificates; however, these paracertificates may have RFC 3779 resources that were not altered during algorithm processing. The final output of algorithm processing will be threefold. First, the state/metadata information on some (original) certificates in the repository MAY be altered. Second, paracertificates will be created, with the appropriate metadata, and entered into the repository. Finally, the TA processing algorithm SHOULD produce a human readable log of its actions, indicating which paracertificates were created and why. The remainder of this section describes the processing stages of the algorithm in detail.

#### 4.2.1 Preparatory processing (stage 0)

During preparatory processing, the constraints file, or the corresponding output file of the proofreader algorithm, is read. Internal variables are set corresponding to each tag and flag, if present, or to their defaults, if absent. Internal variables are also set corresponding to the PRIVATEKEYMETHOD value string(s) and the TOPLEVELCERTIFICATE string. The TA processing algorithm is queried to determine if it supports the indicated private key access methodology. This query is performed in an implementation-specific manner. In particular, an implementation is free to vacuously return success to this query. The TA processing algorithm next uses the value string for the TOPLEVELCERTIFICATE to locate this certificate, again in an implementation-specific manner. The certificate in question may already be present in the local RPKI repository, or it may be located elsewhere. The implementation is also free to create the top level certificate at this time, and then assign to this newly-created certificate the name indicated. It is necessary only that, at the conclusion of this processing, a valid trust anchor certificate for the relying party has been created or otherwise obtained.

Some form of access to the RP's private key and top level certificate are required for subsequent correct operation of the algorithm. Therefore, stage 0 processing MUST terminate if one or both conditions are not satisfied. In the error case, the implementation SHOULD provide an error message of sufficient detail that the user can correct the error(s). If stage 0 processing does not succeed, no further stages of TA processing are executed.

#### 4.2.2 Target processing (stage 1)

During target processing, the TA processing algorithm reads all target blocks in the constraints file or corresponding proofreader output file. It then processes each target block in the order specified in the file. In the description that follows, except where noted, the operation of the algorithm on a single target block will be described. Note, however, that all stage 1 processing is executed before any processing in subsequent stages is performed.

The algorithm first obtains the SKI region of the target block. It then locates, in an implementation-dependent manner, the certificate the SKI extension field of which contains that value. Note that if paracertificates have been created by virtue of previous target blocks being processed, those paracertificates are not searched in attempting to locate a certificate with a matching SKI; only original certificates are searched. If more than one original certificate is found matching this SKI, there are two possible scenarios. If a resource holder has two certificates issued by the same CA, with overlapping validity intervals and the same key, but distinct subject names (typically, by virtue of the SerialNumber parts being different), then these two certificates are both considered to be (distinct) targets, and are both processed. If, however, a resource holder has certificates issued by two different CAs, containing different resources, but using the same key, there is no unambiguous method to decide which of the certificates is intended as the target. In this latter case the algorithm MUST issue a warning to that effect, mark the target block in question as unavailable for processing by subsequent stages and proceed to the next target block. If no certificate is found then the algorithm SHOULD issue a warning to that effect and proceed to process the next target block.

If a single original certificate is found matching the indicated SKI, then the algorithm takes the following actions. First, it sets the ORIGINAL state bit for the certificate found. Second, it sets the TARGET state bit for the certificate found. Third, it extracts the RFC 3779 resources from the certificate. If the global resource\_nounion flag is TRUE, it compares the extracted certificate resources with the resources specified in the constraints file. If the two resource sets are different, the algorithm SHOULD issue a warning noting the difference. An output resource set is then formed that is identical to the resource set extracted from the certificate. If, however, the resource\_nounion flag is FALSE, then the output resource set is calculated by forming the union of the resources extracted from the certificate and the resources specified for this target block in the constraints file. A paracertificate is then constructed according to Table 1, using fields from the original certificate, the tags that had been set during stage 0, and, if

necessary, fields from the RP's TA certificate. The RFC 3779 resources of the paracertificate are equated to the derived output resource set. The PARA state bit is set for the newly created paracertificate.

#### 4.2.3 Ancestor processing (stage 2)

The goal of ancestor processing is to discover all ancestors of target certificates and remove from those ancestors the resources specified in the target blocks corresponding to the targets being processed. Note that it is possible that, for a given chain from a target certificate to a trust anchor, another target might be encountered. This is handled by removing all the target resources of all descendants. The set of all targets that are descendants of the given certificate is formed. The union of all the target resources of the corresponding target blocks is computed, and this union is then removed from the shared ancestor.

In detail, the algorithm is as follows. First, all original target certificates processed during stage 1 processing are collected. Second, any such certificates that have the NOCHAIN state bit set are eliminated from the collection. (Note that, as a result of eliminating such certificates, the resulting collection may be empty, in which case this stage of algorithm processing terminates, and processing advances to stage 3.) Next, an implementation MAY sort the collection. The optional sorting algorithm is described in Appendix B. Note that all stage 2 processing is completed before any stage 3 processing.

Two levels of nested iteration are performed. The outer iteration is effected over all certificates in the collection; the inner iteration is over all ancestors of the designated certificate being processed. The first certificate in the collection is chosen, and a resource set R is initialized based on the resources of the target block for that certificate (since the certificate is in the collection, it must be a target certificate, and thus correspond to a target block). The parent of the certificate is then located using ordinary path discovery over original certificates only. The ancestor's certificate resources A are then extracted. These resources are then perforated with respect to R. That is, an output set of resources is created by forming the intersection I of A and R, and then taking the set difference  $A - I$  as the output resources. A paracertificate is then created containing resources that are these output resources, and containing other fields and extensions from the original certificate (and possibly the RP's TA certificate) according to the procedure given in Table 1. The PARA state bit is set on this paracertificate and the ORIGINAL state bit is set on A. If A is also a target certificate, as indicated by its TARGET state bit being set, then

there will already have been a paracertificate created for it. This previous paracertificate is destroyed in favor of the newly created paracertificate. In this case also, the set R is augmented by adding into it the set of resources of the target block for A. The algorithm then proceeds to process the parent of A. This inner iteration continues until the self-signed certificate at the root of the path is encountered and processed. The outer iteration then continues by clearing R and proceeding to the next certificate in the target collection.

Note that ancestor processing has the potential for order dependency, as mentioned earlier in this document. If sorting is not implemented, or if the sorting algorithm fails to completely process the collection of target certificates because the allotted maximum number of iterations has been realized, it may be the case that an ancestor of a certificate logically occurs before that certificate in the collection. Whenever an existing paracertificate is replaced by a newly created paracertificate during ancestor processing, the algorithm SHOULD alert the user, and SHOULD log sufficient detail such that the user is able to determine which resources were perforated from the original certificate in order to create the (new) paracertificate.

In addition, implementations MUST provide for conflict detection and notification during ancestor processing. In particular, if a certificate is encountered two or more times during any part of the ancestor processing algorithm, and the modifications dictated by the ancestor processing algorithm are in conflict, the implementation MUST refrain from processing that certificate. Further, the implementation MUST present the user with an error message that contains enough detail that the user can locate those directives in the constraints file that are creating the conflict. For example, during one stage of the processing algorithm it may be directed that resources R1 be added to a certificate C, while during a different stage of the processing algorithm it may be directed that resources R2 be removed from certificate C. If the resource sets R1 and R2 have a non-empty intersection, that is a conflict.

#### 4.2.4 Tree processing (stage 3)

The goal of tree processing is to locate other certificates the resources of which might conflict with the resources allocated to a target by virtue of their being mentioned in the constraints file. In this stage of processing, certificates that are not ancestors of any target are considered. In detail, the algorithm used is as follows. First, all target certificates are again collected. Second, all target certificates that have the NOCHAIN state bit set are eliminated from this collection. Third, if the intersection\_always

global flag is set, those target blocks that occur in the constraints file, but that did not correspond to a certificate in the local repository, are also added to the collection. In tree processing, unlike ancestor processing, this collection is not sorted. An iteration is now performed over each certificate (or set of target block resources) in the collection. Note that the collection may be empty, in which case this stage of algorithm processing terminates, and processing advances to stage 4. Note also that all stage 3 processing is performed before any stage 4 processing.

Given a certificate or target resource block, each top level original TA certificate is examined. If that TA certificate has an intersection with the target block resources, then the certificate is perforated with respect to those resources. A paracertificate is created based on the contents of the original certificate (and possibly the RP's TA certificate, as indicated in Table 1) using the perforated resources. The ORIGINAL state bit is set on the original certificate processed in this manner, and the PARA state bit is set on the paracertificate just created. An inner iteration then begins on the descendants of the original certificate just processed. There are two ways in which this iteration may proceed. If the treegrowth global flag is clear, then examination of the children proceeds until all children are exhausted, or until one child is found with intersecting resources. If the treegrowth global flag is set, all children are examined. Since a transfer of resources may be in process such that more than one child possesses intersecting resources, it is RECOMMENDED that the treegrowth flag be set. The inner iteration proceeds until all descendants have been examined and no further intersecting resources are found. The outer iteration then continues with the next certificate or target resource block in the collection. Note that unlike ancestor processing, there is no concept of a potentially cumulating resource collection R; only the resources in the target block are used for perforation.

#### 4.2.5 TA re-parenting (stage 4)

In the final stage of TA algorithm processing, all TA certificates (other than the RP's TA certificate) that have not already been processed in a previous stage are now processed. It will be the case that all such unprocessed TA certificates have no intersection with any target resource blocks. As such, in creating the corresponding paracertificates, the output resource set is identical to the input resource set. Other transformations as described in Table 1 are performed. The original TA certificates have the ORIGINAL state bit set; the newly created paracertificates have the PARA state bit set. Note that once stage four processing is completely, only a single TA certificate will remain in an unprocessed state, namely the relying party's own TA certificate.



### 4.3 Discussion

The algorithm described in this document effectively creates two coexisting certificate hierarchies: the original certificate hierarchy and the paracertificate hierarchy. Note that original certificates are not removed during any of the processing described in the previous section. Some original certificates may move from having no state bits set (or only the NOCHAIN state bit set) to having one or both of the ORIGINAL and TARGET state bits set. In addition, the NOCHAIN state bit will still be set if it was set before any processing. The paracertificate hierarchy, however, is intended to supersede the original hierarchy for the purposes of ROA validation. The presence of two hierarchies has implications for the handling of path discovery, and also for the handling of revocation. If one thinks of a certificate as being "named" by its SKI, then there can now be two certificates with the same name, one an original certificate and the other a paracertificate. The next two sections discuss the implications of this duality in detail. Before proceeding, it is worth noting that even without the existence of the paracertificate hierarchy, cases may exist in which two or more original certificates have the same SKI. As noted earlier, in Section 4.2.2, these cases may be subdivided into the case in which such certificates are distinguishable by virtue of having different subject names, but identical issuers and resource sets, versus all other cases. In the distinguishable case, the path discovery algorithm treats the original certificates as separate certificates, and processes them separately. In all other cases, the original certificates should be treated as indistinguishable, and path validation should fail.

## 5 Implications for Path Discovery

Path discovery proceeds from a child certificate C by asking for a parent certificate P such that the AKI of C is equal to the SKI of P. With one hierarchy this question would produce at most one answer. With two hierarchies, the original certificate hierarchy and the paracertificate hierarchy, the question may produce two answers, one answer, or no answer. Each of these cases is considered in turn.

### 5.1 Two answers

In this case, it SHOULD be the case that one of the matches is a certificate with the ORIGINAL state bit set and the PARA state bit clear, while the other match inversely has the ORIGINAL state bit clear and the PARA state bit set. If any other combination of ORIGINAL and PARA state bits obtains, the path discovery algorithm MUST alert the user. In addition, the path discovery algorithm SHOULD refrain from attempting to make a choice as to which of the two

certificates is the putative parent. In the no-error case, with the state bits as indicated, the certificate with the PARA state bit set is chosen as the parent P. Note this means, in effect, that all children of the original certificate have been re-parented under the paracertificate.

## 5.2 One answer

If the matching certificate has neither the ORIGINAL state bit set nor the PARA state bit set, this certificate is the parent. If the matching certificate has the PARA state bit set but the ORIGINAL state bit not set, this certificate is the parent. (This situation would arise, for example, if the original certificate had been revoked by its issuer but the paracertificate had not been revoked by the RP.) If the matching certificate has the ORIGINAL state bit set but the PARA state bit not set, this is not an error but it is a situation in which path discovery MUST be forced to fail. The parent P MUST be set to NULL, and the NOCHAIN state bit must be set on C and all its descendants; the user SHOULD be warned. Even if the RP has revoked the paracertificate, the original certificate MAY persist. Forcing path discovery to unsuccessfully terminate is a reflection of the RP's preference for path discovery to fail as opposed to using the original hierarchy. Finally, if the matching certificate has both the ORIGINAL and PARA state bits set, this is an error. The parent P MUST be set to NULL, and the user MUST be warned.

## 5.3 No answer

This situation occurs when C has no parent in either the original hierarchy or the paracertificate hierarchy. In this case the parent P is NULL and path discovery terminates unsuccessfully. The NOCHAIN state bit must be set on C and all its descendants.

## 6 Implications for Revocation

In a standard implementation of revocation in a PKI, a valid CRL names a (sibling) certificate by serial number. That certificate is revoked and is purged from the local RPKI repository. In the mechanism described in this document, the original certificate hierarchy and the paracertificate hierarchy are closely related. It can thus be asked how revocation is handled in the presence of these two hierarchies, in particular with regard to whether changes in one of the hierarchies triggers corresponding changes in the other hierarchy. There are four cases based on the state of the ORIGINAL and PARA bits. These will be discussed in the subsections below. It should be noted that the existence of two hierarchies presents a particular challenge with respect to revocation. If a CRL arrives and is processed, that can result in the destruction of one of the path

chains. In the case of a single hierarchy this would mean that certain objects would fail to validate. In the presence of two hierarchies, however, a CRL revocation may force the preferred path to be destroyed. If the RP later determines that the CRL revocation should not have occurred, he is faced with an undesirable situation: the deprecated path will be discovered. In order to prevent this outcome, an RP MUST be able to configure one or more additional repository URIs in support of local trust anchor management.

#### 6.1 No state bits set

If the CRL names a certificate that has neither the ORIGINAL state bit set nor the PARA state bit set, revocation proceeds normally. All children of the revoked certificate have their state modified so that the NOCHAIN state bit is set.

#### 6.2 ORIGINAL state bit set

If the CRL names a certificate with the ORIGINAL state bit set and the PARA state bit clear, then this certificate is revoked as usual. If this original certificate also has the TARGET state bit set, then the corresponding paracertificate (if it exists) is not revoked; if this original certificate has the TARGET state bit clear, then the corresponding paracertificate is revoked as well. Note that since all the children of the original certificate have been re-parented to be children of the corresponding paracertificate, as described above, the revocation algorithm MUST NOT set the NOCHAIN state bit on these children unless the paracertificate is also revoked. Note also that if the original certificate is revoked but the paracertificate is not revoked, the paracertificate retains its PARA state bit. This is to ensure that path discovery proceeds preferentially through the paracertificate hierarchy, as described above.

#### 6.3 PARA state bit set

If the CRL names a certificate with the PARA state bit set and the ORIGINAL state bit clear, this CRL must have been issued, perforce, by the RP itself. This is because all the paracertificates are children of the RP's TA certificate. (Recall that a TA is not revoked via a CRL; it is merely removed from the repository.) The paracertificate is revoked and all children of the paracertificate have the NOCHAIN state bit set. No action is taken on the corresponding original certificate; in particular, its ORIGINAL state bit is not cleared.

Note that the serial numbers of paracertificates are synthesized according to the procedure given in Table 1, rather than being assigned by an algorithm under the control of the (original) issuer.

#### 6.4 Both ORIGINAL and PARA state bits set

This is an error. The revocation algorithm MUST alert the user and take no further action.

,

### 7 Security Considerations

The goal of the algorithm described in this document is to enable an RP to impose its own constraints on its view of the RPKI, which itself is a security function. An RP using a constraints file is trusting the assertions made in that file. Errors in the constraints file used by an RP can undermine the security offered by the RPKI, to that RP. In particular, since the paracertificate hierarchy is intended to trump the original certificate hierarchy for the purposes of path discovery, an improperly constructed paracertificate hierarchy could validate origin attestations that would otherwise be invalid, or could declare as invalid origin attestations that would otherwise be valid. As a result, an RP must carefully consider the security implications of the constraints file being used.

### 8 IANA Considerations

[Note to IANA, to be removed prior to publication: there are no IANA considerations stated in this version of the document.]

### 9 Acknowledgements

The authors would like to acknowledge the significant contributions of Charles Gardiner, who was the original author of an internal version of this document, and who contributed significantly to its evolution into the current version.

### 10 References

#### 10.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3513] Hinden, R., and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.
- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key

Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

- [RFC5396] Huston, G., and G. Michaelson, "Textual Representation of Autonomous System (AS) Numbers", RFC 5396, December 2008.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Policy Structure", RFC 6481, February 2012.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.

## 10.2 Informative References

None.

### Authors' Addresses

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

Email: kent@bbn.com

Matthew Lepinski  
Raytheon BBN Technologies  
10 Moulton St.  
Cambridge, MA 02138

Email: mlepinsk@bbn.com

Mark Reynolds  
Island Peak Software  
328 Virginia Road  
Concord, MA 01742

Email: mcr@islandpeaksoftware.com

## Appendix A: Sample Constraints File

```
;
; Sample constraints file for TBO LTA Test Corporation.
;
; TBO manages its own local (10.x.x.x) address space
; via the target blocks in this file.
;

;
; Relying party subsection. TBO uses ssh-agent as
; a software cryptographic agent.
;

PRIVATEKEYMETHOD      OBO(ssh-agent)
TOPLEVELCERTIFICATE    tbomaster.cer

;
; Flags subsection
;
; Always use the resources in this file to augment
; certificate resources.
; Always process resource conflicts in the tree, even
; if the target certificate is missing.
; Always search the entire tree.
;

CONTROL  resource_nounion      FALSE
CONTROL  intersection_always   TRUE
CONTROL  treegrowth            TRUE

;
; Tags subsection
;
; Copy the original cert's validity dates.
; Use the default policy OID.
; Use our own CRLDP.
; Use our own AIA.
;

TAG      Xvalidity_dates      C
TAG      Xcp                  D
TAG      Xcrl dp              rsync://tbo_lta_test.com/pub/CRLs
TAG      Xaia                  rsync://tbo_lta_test.com/pub/repos

;
; Block subsection
```

```
;
;
; First block: TBO corporate
;

SKI 00112233445566778899998877665544332211
  IPv4
    10.2.3/24
    10.8/16
  IPv6
    2000:2:3:4:5:6/112
  AS#
    60123
    5507

;
; Second block: TBO LTA Test Enforcement Division
;

SKI 653420AF758421CF600029FF857422AA6833299F
  IPv4
    10.2.8/24
    10.47/16
  IPv6
  AS#
    60124

;
; Third block: TBO LTA Test Acceptance Corporation
; Quality financial services since sometime
; late yesterday.
;

SKI 19:82:34:90:8b:a0:9c:ef:00:af:a0:98:23:09:82:4b:ef:ab:98:09
  IPv4
    10.3.3/24
  IPv6
  AS#
    60125

; End of TBO constraints file
```

## Appendix B: Optional Sorting Algorithm for Ancestor Processing

Sorting is performed in an effort to eliminate any order dependencies in ancestor processing, as described in section 4.2.3 of this

document. The sorting algorithm does this by rearranging the processing of certificates such that if A is an ancestor of B, B is processed before A. The sorting algorithm is an OPTIONAL part of ancestor processing. Sorting proceeds as follows. The collection created at the beginning of ancestor processing is traversed and any certificate in the collection that is visited as a result of path discovery is temporarily marked. After the traversal, all unmarked certificates are moved to the beginning of the collection. The remaining marked certificates are unmarked, and a traversal again performed through this sub-collection of previously marked certificates. The sorting algorithm proceeds iteratively until all certificates have been sorted or until a predetermined fixed number of iterations has been performed. (Eight is suggested as a munificent value for the upper bound, since the number of sorting steps need not be any greater than the maximum depth of the tree.) Finally, the ancestor processing algorithm is applied in turn to each certificate in the remaining sorted collection. If the sorting algorithm fails to converge, that is if the maximum number of iterations has been reached and unsorted certificates remain, the implementation SHOULD warn the user.



Network Working Group  
Internet-Draft  
Updates: RFC 6490 (if approved)  
Intended status: Standards Track  
Expires: January 10, 2013

R. Gagliano  
Cisco Systems  
C. Martinez  
LACNIC  
July 9, 2012

Multiple Repository Publication Points support in the Resource Public  
Key Infrastructure (RPKI)  
draft-rogaglia-sidr-multiple-publication-points-00

## Abstract

The Resource Public Key Infrastructure (RPKI) depends on Relying Parties (RP) ability to access its Trust Anchors' certificate specified in the different "Trust Anchor Locator (TAL)" files and the Repository Objects located at the Certificate Authorities (CA) repositories hosted in its respective publication point. This document updates [RFC6490] by allowing multiple URI associated to a single public key in a TAL file and introduces the concept of multiple repository publication point operators for every CA in the RPKI. This document provides also recommendation for the RP behavior when analyzing signed objects that include multiple publications points.

## Status of this Memo

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

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

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

This Internet-Draft will expire on January 10, 2013.

## Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal

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

## Table of Contents

1. Requirements notation . . . . .	3
2. Introduction . . . . .	4
3. Multiple Operators support in TAL files . . . . .	5
3.1. Update to RFC 6490 Section 2.1 . . . . .	5
3.2. Rules for Relying Parties (RP) . . . . .	5
4. Multiple Operators support in Certificates . . . . .	7
4.1. Rules for Relying Parties (RP) . . . . .	7
5. IANA Considerations . . . . .	8
6. Security Considerations . . . . .	9
7. Acknowledgements . . . . .	10
8. Normative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Requirements notation

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].

## 2. Introduction

When thinking on how to scale the RPKI repository system described in [RFC6481] CA operators have a number of options such as:

- o Give the content to a Content Delivery Network (CDN) to have the content distributed (as long as the CDN supports the access method for the CA, which at this time is rsync).
- o Copy the content to different Repository Publication Points around the globe (i.e. using [I-D.ietf-sidr-publication]) and load balance the content using different Domain Name System (DNS) techniques.

When using any of these scaling technique to a unique CA Repository Publication Point URI, there is a dependency in the resolution of a single Fully Qualified Domain Name (FQDN). Also, when a single operator manages a RPKI Repository Publication Point, it is possible to introduce circular dependencies when the Route Origin Authorization (ROA) signed objects for the Repository Publication Point IP addresses are hosted in servers that uses those same addresses. The idea of having multiple Repository Publication Points operators for a RPKI CA mitigates these risks and is complementary to any other scaling solution (as the ones described above).

The first thing that is needed is to add multiple URIs support for each Trust Anchor. [RFC6490] requires that each TAL file includes a unique URI. This document remove this requirement by allowing one or more URI for each public key in a TAL file.

A CA can add support for multiple Repository Publication Points operators by adding more than one respective object for the Authority Information Access (AIA), the Subject Information Access (SIA) and the CRL Distribution Points (CRLDP) and which is supported by [RFC5280] and [RFC6487] .

The addition of multiple Repository Publication Points operators for CAs in the RPKI introduces complexity for the RP. This documents provide some recommendations for RP implementors.

### 3. Multiple Operators support in TAL files

The idea of multiples operators support for a Trust Anchor certificate expressed on its TAL file is similar to the support for several Root Server operators in a DNS hints file.

An example of such a TAL file with 3 operators would be:

```
rsync://rpki.operator1.org/rpki/hedgehog/root.cer
rsync://rpki.operator2.net/rpki/hedgehog/root.cer
rsync://rpki.operator3.biz/rpki/hedgehog/root.cer
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAovWQL2lh6knDx
GUG5hbtCXvvh4AOzjhDkSHlj22gn/loiM9IeDATIwP44vhQ6L/xvuk7W6
Kfa5ygmqQ+xOZOWTPcrUbqaQyPNxokuivzyvqVZVDecOEqs78q58mSp9
nbtxmLRW7B67SJCBSzfa5XpVyXYEgYAJkk3fpmefU+AcxtxvvHB5OVPIa
BfPcs80ICMgHQX+fphvute9XLxjfJKJWkhZqZ0v7pZm2uhkcPx1PMGcrG
ee0WSDC3fr3erLueagpiLsFjwwpX6F+Ms8vqz45H+DKmYKvPSstZjCCq9
aJ0qANT9OtnfSDOS+aLRPjZryCNyvvBHxZXqj5YCGKtwIDAQAB
```

As we can see in this example, a RP would have different URI where to fetch the self-signed certificate for the trust anchor. In each location, the same result should be expected as all the URI share the same public key.

In order to increase in diversity, It is RECOMMENDED that different FQDN could be resolved to IP addresses included in ROA objects from different CAs and hosted in diverse Repository Publication Points.

#### 3.1. Update to RFC 6490 Section 2.1

The following text will replace the last paragraph on Section 2.1 of RFC 6490:

The TAL is an ordered sequence of:

- 1) One or more rsync URI [RFC5781],
- 2) A <CRLF> or <LF> line break after each URI, and
- 3) A subjectPublicKeyInfo [RFC5280] in DER format [X.509], encoded in Base64 (see Section 4 of [RFC4648]).A

#### 3.2. Rules for Relying Parties (RP)

A RP can use different rules to select the URI from where fetch the Trust Anchor certificate. Some examples are:

- o Using the order provided in the TAL file
- o Selecting the URI randomly from the available list
- o Creating a prioritized list of URIs based on RP specific parameters such as connection establishment delay

If the connection to the preferred URI fails or the fetched certificate public key does not match the TAL public key, the RP SHOULD fetch the TA certificate from the next URI of preference.

#### 4. Multiple Operators support in Certificates

The support for multiple operators in the RPKI Certificate Authority (CA) and End Entity (EE) certificates is supported as the RFC 5082 allows multiple repository publication point operators as the SIA, AIA and CRLDP are implemented as sequences. Consequently, no changes are needed on the existing RPKI standard and this section could be considered informative.

In the case of the SIA extension, for each operator, the accessMethods for both the CA repository publication point and for the correspondent manifest needs to be added.

##### 4.1. Rules for Relying Parties (RP)

A RP can use different rules to select the URI to fetch the different repository objects and when performing the validation.

When a RP needs to fetch one or more object from a list of possible URIs, it can chose the URI by adopting a locally defined rule that could be:

- o Using the order provided in the correspondent certificate
- o Selecting the URI randomly from the available list
- o Creating a prioritized list of URIs based on RP specific parameters such as connection establishment delay

If the connection to the preferred URI fails , the RP SHOULD fetch the repository objects from the next URI of preference.

## 5. IANA Considerations

No IANA requirements



## 6. Security Considerations

TBA

## 7. Acknowledgements

TBA.

## 8. Normative References

- [I-D.ietf-sidr-publication]  
"A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", <<http://www.ietf.org/id/draft-ietf-sidr-publication-02.txt>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, February 2012.
- [RFC6484] Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource Public Key Infrastructure (RPKI)", BCP 173, RFC 6484, February 2012.
- [RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure (RPKI)", RFC 6485, February 2012.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, February 2012.
- [RFC6490] Huston, G., Weiler, S., Michaelson, G., and S. Kent, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 6490, February 2012.
- [RFC6492] Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", RFC 6492, February 2012.

Authors' Addresses

Roque Gagliano  
Cisco Systems  
Avenue des Uttins 5  
Rolle, 1180  
Switzerland

Email: [rogaglia@cisco.com](mailto:rogaglia@cisco.com)

Carlos Marcelo  
LACNIC  
Rambla Republica de Mexico 6125  
Montevideo, 11400  
Uruguay

Email: [carlos@lacnic.net](mailto:carlos@lacnic.net)



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: December 31, 2012

R. Bush  
Internet Initiative Japan  
July 2012

Responsible Grandparenting in the RPKI  
draft-ymbk-rpki-grandparenting-02

Abstract

There are circumstances in RPKI operations where a resource holder's parent may not be able to, or may not choose to, facilitate full and proper registration of the holder's data. As in real life, the holder may form a relationship with their grandparent who is willing to aid the grandchild. This document describes simple procedures for doing so.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 31, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

## Table of Contents

1. Introduction . . . . .	2
2. Suggested Reading . . . . .	2
3. What to Do . . . . .	2
4. Security Considerations . . . . .	3
5. IANA Considerations . . . . .	3
6. References . . . . .	3
Author's Address . . . . .	4

## 1. Introduction

There are circumstances in RPKI operations where a resource holder's parent may not be able to, or may not choose to, facilitate full and proper registration of the holder's data. As in real life, the holder may form a relationship with their grandparent who is willing to aid the grandchild. This document describes simple procedures for doing so.

An example might be when provider A allowed a child, C, to move to other provider(s) and keep their address space, either temporarily or permanently, and C's child, G, wished to stay with provider A.

Or a child, C, in the process of going out of business might place their grandchildren in precarious circumstances until they can re-home. The grandparent, without disturbing the child's data, could simply issue ROAs for the grandchildren, or issue certificates for those willing to manage their own rpki data.

Certification Authorities with a large number of children, e.g. very large ISPs or RIRs, might offer documented grandparenting processes and/or agreements. This might reassure grandchildren with worries about irresponsible parents.

Other examples occur in administrative hierarchies, such as large organizations or military and other government hierarchies, when A's child C wishes to manage their own data but does not wish the technical or administrative burden of managing their children's, Gs', data.

## 2. Suggested Reading

It is assumed that the reader understands the RPKI, see [RFC6480], ROAs, see [RFC6482], BGPSEC Router Certificates, see [I-D.ietf-sidr-bgpsec-pki-profiles], and the operational guidance for origin validation, [I-D.ietf-sidr-origin-ops].

## 3. What to Do

A hypothetical example might be that A has the rights to 10.0.0.0/8, has delegated 10.42.0.0/16 to their child C, who delegated 10.42.2.0/23 to their child G. C has changed providers and kept, with A's consent, 10.42.0.0/16, but G wishes to stay with A and keep 10.42.2.0/23.

Perhaps there are also AS resources involved, and G wishes to issue Router Certificates for their AS(s).

Managing RPKI data in such relationships is simple, but should be done carefully.

First, using whatever administrative and/or contractual procedures are appropriate in the local hierarchy, the grandparent, A, should ensure their relationship to the grandchild, G, and that G has the right to the resources which they wish to have registered. These are local matters between A and G.

Although A has the rights over their child's, C's, resources, it would be prudent and polite to ensure that C agrees to A forming a relationship to G. Again, these are local matters between A, C, and G. Often, no one outside of one of these bi-lateral relationships actually knows the agreement between the parties.

Then, it is trivial within the RPKI for A to certify G's data, even though it is a subset of the resources A delegated to C. A may certify G's resources, or issue one or more EE certificates and ROAs for G's resources. Which is done is a local matter between A and G.

#### 4. Security Considerations

This operational practice presents no technical security threats beyond those of the relevant RPKI specifications.

There are threats of social engineering by G, lying to A about their relationship to and rights gained from C.

There are also threats of social engineering by C, attempting to prevent A from giving rights to G which G legitimately deserves.

#### 5. IANA Considerations

This document has no IANA Considerations.

#### 6. References

[I-D.ietf-sidr-bgpsec-pki-profiles]

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

[I-D.ietf-sidr-origin-ops]

Bush, R., "RPKI-Based Origin Validation Operation", Internet-Draft draft-ietf-sidr-origin-ops-17, June 2012.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.



[RFC6482] Lepinski, M., Kent, S. and D. Kong, "A Profile for Route  
Origin Authorizations (ROAs)", RFC 6482, February 2012.

Author's Address

Randy Bush  
Internet Initiative Japan  
5147 Crystal Springs  
Bainbridge Island, Washington 98110  
US

Email: [randy@psg.com](mailto:randy@psg.com)