

Internet Engineering Task Force (IETF)  
Internet-Draft  
Updates: 8205 (if approved)  
Intended status: Standards Track  
Expires: July 19, 2021

O. Borchert  
D. Montgomery  
USA NIST

January 15, 2021

BGPsec Validation State Unverified  
draft-borchert-sidrops-bgpsec-state-unverified-04

## Abstract

In case operators decide to delay BGPsec path validation, none of the available states do properly represent this decision. This document introduces "Unverified" as a well-defined validation state which allows to properly identify a non-evaluated BGPsec routes as not verified.

## 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>

## Copyright Notice

Copyright (c) 2021 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
1.1. Terminology . . . . .	3
2. Suggested Reading . . . . .	3
3. Initializing BGPsec route . . . . .	3
3.1. Changes to RFC 8205 . . . . .	4
3. Usage Considerations . . . . .	4
4. Security Considerations . . . . .	4
5. IANA Considerations . . . . .	4
6. References . . . . .	5
6.1. Normative References . . . . .	5
8.2. Informative References . . . . .	5
Acknowledgements . . . . .	5
Authors' Addresses . . . . .	6

## 1. Introduction

BGPsec path validation [RFC8205] provides well defined validation states. Though, there are instances in which BGPsec routes are not immediately validated upon receiving them. This could be due to configuration where the operator chose to perform "Lazy Evaluation" or due to instances where router configuration could enable the operator to delay route validation during situations of unexpectedly high loads such as DDOS attacks or others. Here, the absence of a well-defined initialization state requires to use a validation state, that is otherwise well-defined and therefore "waters" down the meaning of that state.

Hence, this document updates the RFC 8205 by adding the proposed validation state "Unverified".

### 1.1. Terminology

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

## 2. Suggested Reading

It is assumed that the reader understands BGP [RFC4271] and BGPsec Protocol Specification [RFC8205]

## 3. Initializing BGPsec route

This document introduces the validation state "Unverified" to be used for BGPsec routes that are not evaluated otherwise.

To allow proper initialization the following state is introduced:

- o Unverified: Specifies the state of a BGPsec route where no evaluation has been performed.

### 3.1. Changes to RFC 8205

The BGPsec protocol specification as specified in [RFC8205] suffers the limitation described above in this document. [Section 5.1] of RFC 8205 specifies two states for BGPsec path validation:

The validation procedure results in one of two states:  
'Valid' and 'Not Valid'.

Also, [Section 5.1] makes it clear that:

BGPsec validation need only be performed at the eBGP edge.

This document updates RFC 8205 in such that:

BGPsec routes MUST be initialized using the BGPsec validation state "Unverified" until proper evaluation of the BGPsec route has been performed.

### 3. Usage Considerations

The validation state "Unverified" allows to distinguish between evaluated BGPsec routes and non-evaluated BGPsec routes. This allows the operator to create policies to treat such routes different from routes labeled with either validation state "Valid" or "Not Valid"

### 4. Security Considerations

This document introduces no new security concerns beyond what is described in [RFC8205]

### 5. IANA Considerations

This document has no IANA actions.

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8205] Lepinski, M., Ed., and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.

### 8.2. Informative References

- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.

## Acknowledgements

The authors would like to acknowledge the valuable review and suggestions from K. Sriram on this document.

Authors' Addresses

Oliver Borchert  
National Institute of Standards and Technology (NIST)  
100 Bureau Drive  
Gaithersburg, MD 20899  
United States of America

Email: [oliver.borchert@nist.gov](mailto:oliver.borchert@nist.gov)

Doug Montgomery  
National Institute of Standards and Technology (NIST)  
100 Bureau Drive  
Gaithersburg, MD 20899  
United States of America

Email: [doug@nist.gov](mailto:doug@nist.gov)

Internet Engineering Task Force (IETF)  
Internet-Draft  
Updates: 6811, 8097 (if approved)  
Intended status: Standards Track  
Expires: July 19, 2021

O. Borchert  
D. Montgomery  
USA NIST

January 15, 2021

RPKI Route Origin Validation State Unverified  
draft-borchert-sidrops-rpki-state-unverified-04

## Abstract

In case operators decide not to evaluate BGP route prefixes according to RPKI route origin validation (ROV), none of the available states as specified in RFC 6811 do properly represent this decision. This document introduces "Unverified" as well-defined validation state which allows to properly identify route prefixes as not evaluated according to RPKI route origin validation.

## 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>

## Copyright Notice

Copyright (c) 2021 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
1.1. Terminology . . . . .	3
2. Suggested Reading . . . . .	3
3. Initializing route prefixes . . . . .	3
3.1. Update to RFC 6811 . . . . .	4
3.2. Update to RFC 8097 . . . . .	4
3. Usage Considerations . . . . .	5
4. Security Considerations . . . . .	5
5. IANA Considerations . . . . .	5
6. References . . . . .	6
6.1. Normative References . . . . .	6
8.2. Informative References . . . . .	6
Acknowledgements . . . . .	7
Authors' Addresses . . . . .	7



## 1. Introduction

Prefix origin validation provides well-defined validation states. Though, there are instances in which no evaluation of a route prefix is performed, not through RPKI route origin validation [RFC6811], signaling via the extended community string as specified in [RFC8097], or operator configuration. In these circumstances RFC 6811 specifies the implementation SHOULD initialize the validation state of such route to "NotFound". Here, the absence of a well-defined validation state for a route prefix not evaluated, requires the usage of a state otherwise reserved as outcome of the evaluation of such. This "waters" down the meaning of the used state. The specification of a proper validation state that allows identifying non-evaluated routes, becomes of essence once an operator decides to write policies on the validation state "NotFound". A route prefix labeled "NotFound" cannot be considered same as an unverified route prefix.

Hence, this document updates RFC 6811 and RFC 8097 by adding the proposed validation state "Unverified".

### 1.1. Terminology

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

## 2. Suggested Reading

It is assumed that the reader understands BGP [RFC4271], the RPKI [RFC6480], Route Origin Authorizations (ROAs) [RFC6482], RPKI-based Prefix Validation [RFC6811], BGP Prefix Origin Validation State Extended Community [RFC8097], Clarifications to BGP Origin Validation Based on Resource Public Key Infrastructure (RPKI) [RFC8481]

## 3. Initializing route prefixes

This document introduces the validation state "Unverified" to be used for route prefixes that are not evaluated through either operator configuration, RPKI route origin validation, or other means such as receiving a signaled validation state via the extended community string. To allow proper initialization the following state is introduced:

- o Unverified: Specifies the state of a route prefix on which no evaluation has been performed.

### 3.1. Update to RFC 6811

RFC 6811 specifies that:

If validation is not performed on a Route, the implementation SHOULD initialize the validation state of such a route to "NotFound".

This document specifies that:

If no evaluation of a route prefix is performed in any form, the implementation MUST initialize the validation state of such a route to "Unverified".

This removes the necessity to initialize the route with any of the states "Valid", "Invalid", or "NotFound" and therefore does not "water-down" the meaning of such.

### 3.2. Update to RFC 8097

As specified in RFC 8097:

If the router is configured to support the extensions defined in this document - (RFC 8097) - ", it SHOULD attach the origin validation state extended community to BGP UPDATE messages sent to IBGP peers by mapping the computed validation state in the last octet of the extended community.

The missing part here is what to do with route prefixes not evaluated and no validation state was assigned. At this point the only solution is to omit the extended community for such routes. If the usage of the extended community would have been negotiated during the BGP OPEN MESSAGE the receiver would be able to determine that the sender did not evaluate the route in any form. But this is not the case, so a receiver does not know if the sender is RPKI capable and chose not to attach the origin validation state to the BGP UPDATE or the route did not have any validation state assigned.

Hence, this document specifies for all routes that are labeled as "Unverified" to attach the "unverified" state extended community to BGP UPDATE messages send to IBGP peers by mapping the computed validation state in the last octet of the extended community.

AS specified in the table below, this document adds the value "unverified = 3" to the list of acceptable values.

The value on the protocol

Value	Meaning
0	Lookup result = "valid"
1	Lookup result = "not found"
2	Lookup result = "invalid"
3	Lookup result = "unverified"

### 3. Usage Considerations

The well-defined validation state "Unverified" allows to distinguish between evaluated routes and non-evaluated routes. This allows the operator to create policies to treat such route prefixes different from route prefixes labeled with one of the validation states "Valid", "NotFound", or "Invalid".

### 4. Security Considerations

This document introduces no new security concerns beyond what is described in [RFC6811] and [RFC8097]

### 5. IANA Considerations

This document has no IANA actions.

## 6. References

### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6811] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and R. Austein, "BGP Prefix Origin Validation", RFC 6811, DOI 10.17487/RFC6811, January 2013, <<https://www.rfc-editor.org/info/rfc6811>>.
- [RFC8097] Mohapatra, P., Patel, K., Scudder, J., Ward, D., and R. Bush, "BGP Prefix Origin Validation State Extended Community", RFC 8097, DOI 10.17487/RFC8097, March 2017, <<https://www.rfc-editor.org/info/rfc8097>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

### 8.2. Informative References

- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<https://www.rfc-editor.org/info/rfc6480>>.
- [RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <<https://www.rfc-editor.org/info/rfc6482>>.
- [RFC8481] Bush, R., "Clarifications to BGP Origin Validation Based on Resource Public Key Infrastructure (RPKI)", RFC 8481, DOI 10.17487/RFC8481, September 2018, <<https://www.rfc-editor.org/info/rfc8481>>.

#### Acknowledgements

The authors would like to acknowledge the valuable review and suggestions from K. Sriram on this document.

#### Authors' Addresses

Oliver Borchert  
National Institute of Standards and Technology (NIST)  
100 Bureau Drive  
Gaithersburg, MD 20899  
United States of America

Email: [oliver.borchert@nist.gov](mailto:oliver.borchert@nist.gov)

Doug Montgomery  
National Institute of Standards and Technology (NIST)  
100 Bureau Drive  
Gaithersburg, MD 20899  
United States of America

Email: [doug@nist.gov](mailto:doug@nist.gov)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 4 August 2022

A. Azimov  
Yandex  
E. Uskov  
JetLend  
R. Bush  
Internet Initiative Japan  
K. Patel  
Arrcus  
J. Snijders  
Fastly  
R. Housley  
Vigil Security  
31 January 2022

A Profile for Autonomous System Provider Authorization  
draft-ietf-sidrops-aspa-profile-07

Abstract

This document defines a standard profile for Autonomous System Provider Authorization in the Resource Public Key Infrastructure. An Autonomous System Provider Authorization is a digitally signed object that provides a means of validating that a Customer Autonomous System holder has authorized members of Provider set to be its upstream providers and for the Providers to send prefixes received from the Customer Autonomous System in all directions including providers and peers.

Requirements Language

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

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 <https://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 4 August 2022.

## Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. The ASPA Content Type . . . . .	3
3. The ASPA eContent . . . . .	3
3.1. version . . . . .	4
3.2. AFI . . . . .	4
3.3. customerASID . . . . .	4
3.4. providerASSET . . . . .	4
4. ASPA Validation . . . . .	5
5. ASN.1 Module for the ASPA Content Type . . . . .	5
6. IANA Considerations . . . . .	6
7. Security Considerations . . . . .	7
8. Acknowledgments . . . . .	7
9. References . . . . .	7
9.1. Normative References . . . . .	7
9.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

The primary purpose of the Resource Public Key Infrastructure (RPKI) is to improve routing security. (See [RFC6480] for more information.) As part of this infrastructure, a mechanism is needed to validate that a AS has permission from a Customer AS (CAS) holder to send routes in all directions. The digitally signed Autonomous System Provider Authorization (ASPA) object provides this validation

mechanism.

The ASPA uses the template for RPKI digitally signed objects [RFC6488], which defines a Cryptographic Message Syntax (CMS) [RFC5652] wrapper for the ASPA content as well as a generic validation procedure for RPKI signed objects. As ASPAs need to be validated with RPKI certificates issued by the current infrastructure, we assume the mandatory-to-implement algorithms in [RFC6485], or its successor.

To complete the specification of the ASPA (see Section 4 of [RFC6488]), this document defines:

1. The object identifier (OID) that identifies the ASPA signed object. This OID appears in the eContentType field of the encapContentInfo object as well as the content-type signed attribute within the signerInfo structure).
2. The ASN.1 syntax for the ASPA content, which is the payload signed by the CAS. The ASPA content is encoded using the ASN.1 [X680] Distinguished Encoding Rules (DER) [X690].
3. The steps required to validate an ASPA beyond the validation steps specified in [RFC6488]).

## 2. The ASPA Content Type

The content-type for an ASPA is defined as id-cct-ASPA, which has the numerical value of 1.2.840.113549.1.9.16.1.TBD. This OID MUST appear both within the eContentType in the encapContentInfo structure as well as the content-type signed attribute within the signerInfo structure (see [RFC6488]).

## 3. The ASPA eContent

The content of an ASPA identifies the Customer AS (CAS) as well as the Set of Provider ASes (SPAS) that are authorized to further propagate announcements received from the customer. If customer has multiple providers they MUST be registered in a single ASPA object. This rule is important to avoid possible race conditions during updates. An ASPA is formally defined as:



```
ct-ASPA CONTENT-TYPE ::=
    { ASProviderAttestation IDENTIFIED BY id-ct-ASPA }

id-ct-ASPA OBJECT IDENTIFIER ::= { id-ct TBD }

ASProviderAttestation ::= SEQUENCE {
    version [0] ASPAVersion DEFAULT v0,
    aFI AddressFamilyIdentifier,
    customerASID ASID,
    providerASSET SEQUENCE (SIZE(1..MAX)) OF ASID }

ASPAVersion ::= INTEGER { v0(0) }

AddressFamilyIdentifier ::= OCTET STRING (SIZE (2))

ASID ::= INTEGER
```

Note that this content appears as the eContent within the encapContentInfo as specified in [RFC6488].

### 3.1. version

The version number of the ASProviderAttestation MUST be v0.

### 3.2. AFI

The AFI field contains Address Family Identifier for which the relation between customer and provider ASes is authorized. Presently defined values for the Address Family Identifier field are specified in the IANA's Address Family Numbers registry [IANA-AF].

### 3.3. customerASID

The customerASID field contains the AS number of the Autonomous System that authorizes an upstream providers (listed in the providerASSET) to propagate prefixes in the specified address family other ASes.

### 3.4. providerASSET

The providerASSET contains the sequence (set) of AS numbers that are authorized to further propagate announcements in the specified address family received from the customer.

#### 4. ASPA Validation

Before a relying party can use an ASPA to validate a routing announcement, the relying party MUST first validate the ASPA object itself. To validate an ASPA, the relying party MUST perform all the validation checks specified in [RFC6488] as well as the following additional ASPA-specific validation step.

- \* The autonomous system identifier delegation extension [RFC3779] is present in the end-entity (EE) certificate (contained within the ASPA), and the customer AS number in the ASPA is contained within the set of AS numbers specified by the EE certificate's autonomous system identifier delegation extension.

#### 5. ASN.1 Module for the ASPA Content Type

```

RPKI-ASPA-2020
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(0) id-mod-rpki-aspa-2020(TBD2) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS

CONTENT-TYPE
FROM CryptographicMessageSyntax-2010 -- RFC 6268
  { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1)
    pkcs-9(9) smime(16) modules(0) id-mod-cms-2009(58) } ;

ContentSet CONTENT-TYPE ::= { ct-ASPA, ... }

--
-- ASPA Content Type
--

id-smime OBJECT IDENTIFIER ::= { iso(1) member-body(2)
  us(840) rsadsi(113549) pkcs(1) pkcs-9(9) 16 }

id-ct OBJECT IDENTIFIER ::= { id-smime 1 }

id-ct-ASPA OBJECT IDENTIFIER ::= { id-ct TBD }

ct-ASPA CONTENT-TYPE ::=
  { TYPE ASPProviderAttestation IDENTIFIED BY id-ct-ASPA }

ASPProviderAttestation ::= SEQUENCE {
  version [0] ASPAVersion DEFAULT v0,
  aFI AddressFamilyIdentifier,
  customerASID ASID,
  providerASSET SEQUENCE (SIZE(1..MAX)) OF ASID OPTIONAL }

ASPAVersion ::= INTEGER { v0(0) }

AddressFamilyIdentifier ::= OCTET STRING (SIZE (2))

ASID ::= INTEGER

END

```

## 6. IANA Considerations

Please add the id-mod-rpki-aspa-2018 to the SMI Security for S/MIME Module Identifier (1.2.840.113549.1.9.16.0) registry (<https://www.iana.org/assignments/smi-numbers/smi-numbers.xml#security-smime-0>) as follows:

Decimal	Description	Specification
TBD2	id-mod-rpki-aspa-2020	[ThisRFC]

Please add the ASPA to the SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1) registry (<https://www.iana.org/assignments/smi-numbers/smi-numbers.xml#security-smime-1>) as follows:

Decimal	Description	Specification
TBD	id-ct-ASPA	[ThisRFC]

Please add Autonomous System Provider Authorization to the RPKI Signed Object registry (<https://www.iana.org/assignments/rpki/rpki.xhtml#signed-objects>) as follows:

Name	OID	Specification	Spec
Autonomous System Provider Authorization	1.2.840.113549.1.9.16.1.TBD	[ThisRFC]	

Please add an item for the Autonomous System Provider Authorization file extension to the "RPKI Repository Name Scheme" registry created by [RFC6481] as follows:

Filename	RPKI Object	Reference
.asa	Autonomous System Provider Authorization	[draft-ietf-sidrops-aspa-profile]

## 7. Security Considerations

While it's not restricted, but it's highly recommended maintaining for selected Customer AS a single ASPA object that covers all its providers. Such policy should prevent race conditions during ASPA updates that might affect prefix propagation. The software that provides hosting for ASPA records SHOULD support enforcement of this rule. In the case of the transition process between different CA registries, the ASPA records SHOULD be kept identical in all registries.

## 8. Acknowledgments

## 9. References

### 9.1. Normative References

- [IANA-AF] IANA, "Address Family Numbers",  
<[https://www.iana.org/assignments/address-family-numbers/  
address-family-numbers.xhtml](https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml)>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", BCP 14, RFC 2119,  
DOI 10.17487/RFC2119, March 1997,  
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP  
Addresses and AS Identifiers", RFC 3779,  
DOI 10.17487/RFC3779, June 2004,  
<<https://www.rfc-editor.org/info/rfc3779>>.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70,  
RFC 5652, DOI 10.17487/RFC5652, September 2009,  
<<https://www.rfc-editor.org/info/rfc5652>>.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for  
Resource Certificate Repository Structure", RFC 6481,  
DOI 10.17487/RFC6481, February 2012,  
<<https://www.rfc-editor.org/info/rfc6481>>.
- [RFC6485] Huston, G., "The Profile for Algorithms and Key Sizes for  
Use in the Resource Public Key Infrastructure (RPKI)",  
RFC 6485, DOI 10.17487/RFC6485, February 2012,  
<<https://www.rfc-editor.org/info/rfc6485>>.
- [RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object  
Template for the Resource Public Key Infrastructure  
(RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012,  
<<https://www.rfc-editor.org/info/rfc6488>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC  
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,  
May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [X680] ITU-T, "Information technology -- Abstract Syntax Notation  
One (ASN.1): Specification of basic notation",  
ITU-T Recommendation X.680, 2015.
- [X690] ITU-T, "Information Technology -- ASN.1 encoding rules:  
Specification of Basic Encoding Rules (BER), Canonical  
Encoding Rules (CER) and Distinguished Encoding Rules  
(DER)", ITU-T Recommendation X.690, 2015.

## 9.2. Informative References

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<https://www.rfc-editor.org/info/rfc6480>>.

Authors' Addresses

Alexander Azimov  
Yandex

Email: [a.e.azimov@gmail.com](mailto:a.e.azimov@gmail.com)

Eugene Uskov  
JetLend

Email: [eu@jetlend.ru](mailto:eu@jetlend.ru)

Randy Bush  
Internet Initiative Japan

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

Keyur Patel  
Arrcus, Inc.

Email: [keyur@arrcus.com](mailto:keyur@arrcus.com)

Job Snijders  
Fastly  
Amsterdam

Email: [job@fastly.com](mailto:job@fastly.com)

Russ Housley  
Vigil Security, LLC  
918 Spring Knoll Drive  
Herndon, VA 20170  
United States of America

Email: [housley@vigilsec.com](mailto:housley@vigilsec.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 26 February 2022

A. Azimov  
Yandex  
E. Bogomazov  
Qrator Labs  
R. Bush  
Internet Initiative Japan & Arrcus  
K. Patel  
Arrcus, Inc.  
J. Snijders  
Fastly  
25 August 2021

Verification of AS\_PATH Using the Resource Certificate Public Key  
Infrastructure and Autonomous System Provider Authorization  
draft-ietf-sidrops-aspa-verification-08

Abstract

This document defines the semantics of an Autonomous System Provider Authorization object in the Resource Public Key Infrastructure to verify the AS\_PATH attribute of routes advertised in the Border Gateway Protocol.

Requirements Language

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

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 <https://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 26 February 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Introduction . . . . .	2
2. Anomaly Propagation . . . . .	3
3. Autonomous System Provider Authorization . . . . .	4
4. Customer-Provider Verification Procedure . . . . .	4
5. AS_PATH Verification . . . . .	5
5.1. Upstream Paths . . . . .	5
5.2. Downstream Paths . . . . .	7
5.3. Paths from Route Server . . . . .	9
5.4. Mitigation . . . . .	10
6. Disavowal of Provider Authorizaion . . . . .	11
7. Mutual Transit (Complex Relations) . . . . .	11
8. Comparison to Peerlock . . . . .	11
9. Security Considerations . . . . .	12
10. Acknowledgments . . . . .	12
11. References . . . . .	12
11.1. Normative References . . . . .	13
11.2. Informative References . . . . .	13
Authors' Addresses . . . . .	15

## 1. Introduction

The Border Gateway Protocol (BGP) was designed without mechanisms to validate BGP attributes. Two consequences are BGP Hijacks and BGP Route Leaks [RFC7908]. BGP extensions are able to partially solve these problems. For example, ROA-based Origin Validation [RFC6483] can be used to detect and filter accidental mis-originations, and [I-D.ietf-idr-bgp-open-policy] or [I-D.ietf-grow-route-leak-detection-mitigation] can be used to detect accidental route leaks. While these upgrades to BGP are quite useful, they still rely on transitive BGP attributes, i.e. AS\_PATH, that can be manipulated by attackers.



BGPsec [RFC8205] was designed to solve the problem of AS\_PATH validation. Unfortunately, strict cryptographic validation brought expensive computational overhead for BGP routers. BGPsec also proved vulnerable to downgrade attacks that nullify the benefits of AS\_PATH signing. As a result, to abuse the AS\_PATH or any other signed transit attribute, an attacker merely needs to downgrade to 'old' BGP-4.

An alternative approach was introduced with soBGP [I-D.white-sobgp-architecture]. Instead of strong cryptographic AS\_PATH validation, it created an AS\_PATH security function based on a shared database of AS adjacencies. While such an approach has reasonable computational cost, the two side adjacencies don't provide a way to automate anomaly detection without high adoption rate - an attacker can easily create a one-way adjacency. SO-BGP transported data about adjacencies in new additional BGP messages, which was recursively complex thus significantly increasing adoption complexity and risk. In addition, the general goal to verify all AS\_PATHs was not achievable given the indirect adjacencies at internet exchange points.

Instead of checking AS\_PATH correctness, this document focuses on solving real-world operational problems - automatic detection of malicious hijacks and route leaks. To achieve this new AS\_PATH verification procedures are defined to automatically detect invalid (malformed) AS\_PATHs in announcements that are received from customers, peers, providers, RS and RS-clients. This procedure uses a shared signed database of customer-to-provider relationships using a new RPKI object - Autonomous System Provider Authorization (ASPA). This technique provides benefits for participants even during early and incremental adoption.

## 2. Anomaly Propagation

Both route leaks and hijacks have similar effects on ISP operations - they redirect traffic, resulting in increased latency, packet loss, or possible MiTM attacks. But the level of risk depends significantly on the propagation of the anomalies. For example, a hijack that is propagated only to customers may concentrate traffic in a particular ISP's customer cone; while if the anomaly is propagated through peers, upstreams, or reaches Tier-1 networks, thus distributing globally, traffic may be redirected at the level of entire countries and/or global providers.

The ability to constrain propagation of BGP anomalies to upstreams and peers, without requiring support from the source of the anomaly (which is critical if source has malicious intent), should significantly improve the security of inter-domain routing and solve the majority of problems.

### 3. Autonomous System Provider Authorization

As described in [RFC6480], the RPKI is based on a hierarchy of resource certificates that are aligned to the Internet Number Resource allocation structure. Resource certificates are X.509 certificates that conform to the PKIX profile [RFC5280], and to the extensions for IP addresses and AS identifiers [RFC3779]. A resource certificate is a binding by an issuer of IP address blocks and Autonomous System (AS) numbers to the subject of a certificate, identified by the unique association of the subject's private key with the public key contained in the resource certificate. The RPKI is structured so that each current resource certificate matches a current resource allocation or assignment.

ASPA is digitally signed object that bind, for a selected AFI, a Set of Provider AS numbers to a Customer AS number (in terms of BGP announcements not business), and are signed by the holder of the Customer AS. An ASPA attests that a Customer AS holder (CAS) has authorized Set of Provider ASes (SPAS) to propagate the Customer's IPv4/IPv6 announcements onward, e.g. to the Provider's upstream providers or peers. The ASPA record profile is described in [I-D.ietf-sidrops-aspa-profile]. For a selected Customer AS SHOULD exist only single ASPA object at any time. In this document we will use ASPA(AS1, AFI, [AS2, ...]) as notation to represent ASPA object for AS1 in the selected AFI.

### 4. Customer-Provider Verification Procedure

This section describes an abstract procedure that checks that a pair of ASNs (AS1, AS2) is included in the set of signed ASPAs. The semantics of its use is defined in next section. The procedure takes (AS1, AS2, AFI) as input parameters and returns one of three results: "Valid", "Invalid" and "Unknown".

A relying party (RP) must have access to a local cache of the complete set of cryptographically valid ASPAs when performing customer-provider verification procedure.

1. Retrieve all cryptographically valid ASPAs in a selected AFI with a customer value of AS1. The union of SPAS forms the set of "Candidate Providers."

2. If the set of Candidate Providers is empty, then the procedure exits with an outcome of "Unknown."
3. If AS2 is included in the Candidate Providers, then the procedure exits with an outcome of "Valid."
4. Otherwise, the procedure exits with an outcome of "Invalid."

Since an AS1 may have different set of providers in different AFI, it should also have different SPAS in corresponding ASPAs. In this case, the output of this procedure with input (AS1, AS2, AFI) may have different output for different AFI values.

## 5. AS\_PATH Verification

The AS\_PATH attribute identifies the autonomous systems through which an UPDATE message has passed. AS\_PATH may contain two types of components: AS\_SEQUENCES and AS\_SETs, as defined in [RFC4271].

We will use index of AS\_PATH segments, where Seg(0) stands for the segment of originating AS. We will use Seg(I).value and Seg(I).type to represent Ith segment value and type respectively.

The below procedures are applicable only for 32-bit AS number compatible BGP speakers.

### 5.1. Upstream Paths

When a route is received from a customer, a literal peer, or by a RS at an IX, each consecutive AS\_SEQUENCE pair MUST be equal (prepend policy) or belong to customer-provider or mutual transit relationship (Section 7). If there are other types of relationships, it means that the route was leaked or the AS\_PATH attribute was malformed. The goal of the procedure described below is to check the correctness of this statement.

The following Python function and algorithm describes the procedure that MUST be applied on routes with AFI received from a customer, peer or RS-client:

```
def check_upflow_path(aspath, neighbor_as, afi):
    if len(aspath) == 0:
        return Invalid

    if aspath[-1].type == AS_SEQUENCE and aspath[-1].value != neighbor_as:
        return Invalid

    semi_state = Valid

    as1 = 0
    for segment in aspath:
        if segment.type != AS_SEQUENCE:
            as1 = 0
            semi_state = Unverifiable
        elif segment.type == AS_SEQUENCE:
            if not as1:
                as1 = segment.value
            elif as1 == segment.value:
                continue
            else:
                pair_check = verify_pair(as1, segment.value, afi)
                if pair_check == Invalid:
                    return Invalid
                elif pair_check == Unknown and semi_state == Valid:
                    semi_state = pair_check
                as1 = segment.value
    return semi_state
```

1. If the AS\_PATH has zero length then procedure halts with the outcome "Invalid";
2. If the last segment in the AS\_PATH has type AS\_SEQUENCE and its value isn't equal to receiver's neighbor AS then procedure halts with the outcome "Invalid";
3. If there exists I such that Seg(I-1).type and Seg(I).type equal to AS\_SEQUENCE, Seg(I-1).value != Seg(I).value and customer-provider verification procedure (Section 4) with parameters (Seg(I-1).value, Seg(I).value, AFI) returns "Invalid" then the procedure also halts with the outcome "Invalid";
4. If the AS\_PATH has at least one AS\_SET segment then procedure halts with the outcome "Unverifiable";

5. If there exists I such that Seg(I-1).type and Seg(I).type equal to AS\_SEQUENCE, Seg(I-1).value != Seg(I).value and customer-provider verification procedure (Section 4) with parameters (Seg(I-1).value, Seg(I).value, AFI) returns "Unknown" then the procedure also halts with the outcome "Unknown";
6. Otherwise, the procedure halts with an outcome of "Valid".

## 5.2. Downstream Paths

When route is received from provider it may have both Upstream and Downstream fragments, where a Downstream follows an Upstream fragment. If the path differs from this rule, e.g. the Downstream fragment is followed by Upstream fragment it means that the route was leaked or the AS\_PATH attribute was malformed. The first unequal pair of AS\_SEQUENCE segments that has an "Invalid" outcome of the customer-provider verification procedure indicates the end of the Upstream fragment. All subsequent reverse pairs of AS\_SEQUENCE segments MUST be equal (prepend policy) or belong to a customer-provider or mutual transit relationship Section 7, thus can be also verified using ASPA objects.

The following Python function and algorithm describe the procedure that MUST be applied on routes with AFI received from a provider:

```
def check_downflow_path(aspath, neighbor_as, afi):
    if len(aspath) == 0:
        return Invalid

    if aspath[-1].type == AS_SEQUENCE and aspath[-1].value != neighbor_as:
        return Invalid
    else:
        semi_state = Valid

    as1 = 0
    upflow_fragment = True
    for segment in aspath:
        if segment.type != AS_SEQUENCE:
            as1 = 0
            semi_state = Unverifiable
        elif segment.type == AS_SEQUENCE:
            if not as1:
                as1 = segment.value
            elif as1 == segment.value:
                continue
            else:
                if upflow_fragment:
                    pair_check = verify_pair(as1, segment.value, afi)
                    if pair_check == Invalid:
                        upflow_fragment = False
                    elif pair_check == Unknown and semi_state == Valid:
                        semi_state = Unknown
                else:
                    pair_check = verify_pair(segment.value, as1, afi)
                    if pair_check == Invalid:
                        return Invalid
                    elif pair_check == Unknown and semi_state == Valid:
                        semi_state = pair_check
                as1 = segment.value

    return semi_state
```

1. If the AS\_PATH has zero length then procedure halts with the outcome "Invalid";
2. If a route is received from a provider and the last segment in the AS\_PATH has type AS\_SEQUENCE and its value isn't equal to receiver's neighbor AS, then the procedure halts with the outcome "Invalid";

3. Let's define `I_MIN` as the minimal index for which `Seg(I-1).type` and `Seg(I).type` equal to `AS_SEQUENCE`, its values aren't equal and the verification procedure for `(Seg(I-1).value, Seg(I).value, AFI)` returns "Invalid". If `I_MIN` doesn't exist put the length of `AS_PATH` in `I_MIN` variable and jump to 5.
4. If there exists `J > I_MIN` such that both `Seg(J-1).type`, `Seg(J).type` equal to `AS_SEQUENCE`, `Seg(J-1).value != Seg(J).value` and the customer-provider verification procedure (Section 4) returns "Invalid" for `(Seg(J).value, Seg(J-1).value, AFI)`, then the procedure halts with the outcome "Invalid";
5. If the `AS_PATH` has at least one `AS_SET` segment then procedure halts with the outcome "Unverifiable";
6. If there exists `J > I_MIN` such that both `Seg(J-1).type`, `Seg(J).type` equal to `AS_SEQUENCE`, `Seg(J-1).value != Seg(J).value` and the customer-provider verification procedure (Section 4) returns "Unknown" for `(Seg(J).value, Seg(J-1).value, AFI)`, then the procedure halts with the outcome "Unknown";
7. If there exists `I_MIN > J` such that both `Seg(J-1).type`, `Seg(J).type` equal to `AS_SEQUENCE`, `Seg(J-1).value != Seg(J).value` and the customer-provider verification procedure (Section 4) returns "Unknown" for `(Seg(J-1).value, Seg(J).value, AFI)`, then the procedure halts with the outcome "Unknown";
8. Otherwise, the procedure halts with an outcome of "Valid".

### 5.3. Paths from Route Server

A route received from a RS at IX has much in common with route received from a provider. A valid route from RS contains Upflow fragment and MAY contain Downflow fragment that contains IX AS. The ambiguity is created by transparent IXes that by default don't add their AS in the `AS_PATH`. In this case, a route will have only Upflow segment, though even 'transparent' IXes may support control communities that give a way to explicitly add IX AS in the path.

Routes from RS MAY be processed the same way as routes from Providers, but in the case of full IX 'transparency', it will limit the opportunity of IX members to detect and filter route leaks. This document suggests using the presence of IX AS as a token to distinguish if Upflow or Downflow path verification procedure should be applied.

The following Python function and algorithm describe the procedure that SHOULD be applied on routes with AFI received from a RS:

```
def check_ix_path(aspath, neighbor_as, afi):  
    if len(aspath) == 0:  
        return Invalid  
  
    if aspath[-1].value != neighbor_as:  
        return check_upflow_path(aspath, aspath[-1].value, afi)  
    else:  
        return check_downflow_path(aspath, neighbor_as, afi)
```

1. If the AS\_PATH has zero length then procedure halts with the outcome "Invalid";
2. If a route is received from a RS and the last segment in the AS\_PATH isn't equal to receiver's neighbor AS, the result equals to the outcome of upflow verification procedure applied to AS\_PATH with neighbor\_as replaced with the value of the last AS\_PATH segment Section 5.1;
3. If a route is received from a RS and the last segment in the AS\_PATH is equal to receiver's neighbor AS, the result equals to the outcome of downflow verification procedure applied to AS\_PATH Section 5.2;

#### 5.4. Mitigation

If the output of the AS\_PATH verification procedure is "Invalid" the route MUST be rejected.

If the output of the AS\_PATH verification procedure is 'Unverifiable' it means that AS\_PATH can't be fully checked. Such routes should be treated with caution and SHOULD be processed the same way as "Invalid" routes. This policy goes with full correspondence to [I-D.kumari-deprecate-as-set-confed-set].

The above AS\_PATH verification procedure is able to check routes received from customer, peers, providers, RS, and RS-clients. The ASPA mechanism combined with BGP Roles [I-D.ietf-idr-bgp-open-policy] and ROA-based Origin Validation [RFC6483] can provide a fully automated solution to detect and filter hijacks and route leaks, including malicious ones.



## 6. Disavowal of Provider Authorizaion

An ASPA is a positive attestation that an AS holder has authorized its providers to redistribute received routes to the provider's providers and peers. This does not preclude the provider ASes from redistribution to its other customers. By creating an ASPA with providers set of [0], the customer indicates that no provider should further announce its routes. Specifically, AS 0 is reserved to identify provider-free networks, Internet exchange meshes, etc.

An ASPA(AS, AFI, [0]) is a statement by the customer AS that its routes should not be received by any relying party AS from any of its customers or peers.

By convention, an ASPA(AS, AFI, [0]) should be the only ASPA issued by a given AS holder in the selected AFI; although this is not a strict requirement. An AS 0 may coexist with other provider ASes in the same ASPA (or other ASPA records in the same AFI); though in such cases, the presence or absence of the provider AS 0 in ASPA does not alter the AS\_PATH verification procedure.

## 7. Mutual Transit (Complex Relations)

There are peering relationships which can not be described as strictly simple peer-peer or customer-provider; e.g. when both parties are intentionally sending prefixes received from each other to their peers and/or upstreams.

In this case, two corresponding records ASPA(AS1, AFI, [AS2, ...]), ASPA(AS2, AFI, [AS1, ...]) must be created by AS1 and AS2 respectively.

## 8. Comparison to Peerlock

ASPA has much in common with [Peerlock]. Peerlock is a BGP Flexsealing [Flexsealing] protection mechanism commonly deployed by global-scale Internet carriers to protect other large-scale carriers.

Peerlock, unfortunately, depends on a laborious manual process in which operators coordinate the distribution of unstructured Provider Authorizations through out-of-band means in a many-to-many fashion. On the other hand, ASPA's use of PKIX [RFC5280] allows for automated, scalable, and ubiquitous deployment, making the protection mechanism available to a wider range of Internet Number Resource holders.

ASPA mechanics implemented in code instead of Peerlock AS\_PATH regular expressions also provides a way to detect anomalies coming from transit providers and internet exchange route servers.

ASPA is intended to be a complete solution and replacement for existing Peerlock deployments.

## 9. Security Considerations

The proposed mechanism is compatible only with BGP implementations that can process 32-bit ASNs in the AS\_PATH. This limitation should not have a real effect on operations - such legacy BGP routers are rare and it's highly unlikely that they support integration with the RPKI.

ASPA issuers should be aware of the verification implication in issuing an ASPA - an ASPA implicitly invalidates all routes passed to upstream providers other than the provider ASs listed in the ASPA record. It is the Customer AS's duty to maintain a correct set of providers in ASPA record(s).

While it's not restricted, but it's highly recommended maintaining for selected Customer AS a single ASPA object that covers all its providers. Such policy should prevent race conditions during ASPA updates that might affect prefix propagation. The software that provides hosting for ASPA records SHOULD support enforcement of this rule. In the case of the transition process between different CA registries, the ASPA records SHOULD be kept identical in all registries.

While the ASPA is able to detect both mistakes and malicious activity for routes received from customers, RS-clients, or peers, it provides only detection of mistakes for routes that are received from upstream providers and RS(s).

Since an upstream provider becomes a trusted point, it will be able to send hijacked prefixes of its customers or send hijacked prefixes with malformed AS\_PATHs back. While it may happen in theory, it's doesn't seem to be a real scenario: normally customer and provider have a signed agreement and such policy violation should have legal consequences or customer can just drop relation with such a provider and remove the corresponding ASPA record.

## 10. Acknowledgments

The authors wish to thank authors of [RFC6483] since its text was used as an example while writing this document. The also authors wish to thank Iljitsch van Beijnum for giving a hint about Downstream paths.

## 11. References

## 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 11.2. Informative References

- [Flexsealing]  
McDaniel, T., Smith, J., and M. Schuchard, "Flexsealing BGP Against Route Leaks: Peerlock Active Measurement and Analysis", November 2020, <<https://arxiv.org/pdf/2006.06576.pdf>>.
- [I-D.ietf-grow-route-leak-detection-mitigation]  
Sriram, K. and A. Azimov, "Methods for Detection and Mitigation of BGP Route Leaks", Work in Progress, Internet-Draft, draft-ietf-grow-route-leak-detection-mitigation-00, 19 April 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-grow-route-leak-detection-mitigation-00.txt>>.
- [I-D.ietf-idr-bgp-open-policy]  
Azimov, A., Bogomazov, E., Bush, R., Patel, K., and K. Sriram, "Route Leak Prevention using Roles in Update and Open messages", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-open-policy-05, 15 February 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-idr-bgp-open-policy-05.txt>>.
- [I-D.ietf-sidrops-aspa-profile]  
Azimov, A., Uskov, E., Bush, R., Patel, K., Snijders, J., and R. Housley, "A Profile for Autonomous System Provider Authorization", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-profile-00, 17 May 2019, <<http://www.ietf.org/internet-drafts/draft-ietf-sidrops-aspa-profile-00.txt>>.

- [I-D.kumari-deprecate-as-set-confed-set]  
Kumari, W. and K. Sriram, "Deprecation of AS\_SET and AS\_CONFED\_SET in BGP", Work in Progress, Internet-Draft, draft-kumari-deprecate-as-set-confed-set-12, 2 July 2018, <<http://www.ietf.org/internet-drafts/draft-kumari-deprecate-as-set-confed-set-12.txt>>.
- [I-D.white-sobgp-architecture]  
White, R., "Architecture and Deployment Considerations for Secure Origin BGP (soBGP)", Work in Progress, Internet-Draft, draft-white-sobgp-architecture-02, 16 June 2006, <<http://www.ietf.org/internet-drafts/draft-white-sobgp-architecture-02.txt>>.
- [Peerlock] Snijders, J., "Peerlock", June 2016, <[https://www.nanog.org/sites/default/files/Snijders\\_Everyday\\_Practical\\_Bgp.pdf](https://www.nanog.org/sites/default/files/Snijders_Everyday_Practical_Bgp.pdf)>.
- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <<https://www.rfc-editor.org/info/rfc3779>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [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, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<https://www.rfc-editor.org/info/rfc6480>>.
- [RFC6483] Huston, G. and G. Michaelson, "Validation of Route Origination Using the Resource Certificate Public Key Infrastructure (PKI) and Route Origin Authorizations (ROAs)", RFC 6483, DOI 10.17487/RFC6483, February 2012, <<https://www.rfc-editor.org/info/rfc6483>>.
- [RFC7908] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC 7908, DOI 10.17487/RFC7908, June 2016, <<https://www.rfc-editor.org/info/rfc7908>>.

[RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <<https://www.rfc-editor.org/info/rfc8205>>.

Authors' Addresses

Alexander Azimov  
Yandex

Email: [a.e.azimov@gmail.com](mailto:a.e.azimov@gmail.com)

Eugene Bogomazov  
Qrator Labs

Email: [eb@qrator.net](mailto:eb@qrator.net)

Randy Bush  
Internet Initiative Japan & Arrcus

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

Keyur Patel  
Arrcus, Inc.

Email: [keyur@arrcus.com](mailto:keyur@arrcus.com)

Job Snijders  
Fastly  
Amsterdam

Email: [job@fastly.com](mailto:job@fastly.com)

SIDROPS  
Internet-Draft  
Intended status: Informational  
Expires: April 9, 2020

D. Ma  
ZDNS  
S. Kent  
Independent  
October 7, 2019

Requirements for Resource Public Key Infrastructure (RPKI) Relying  
Parties  
draft-ietf-sidrops-rp-06

Abstract

This document provides a single reference point for requirements for Relying Party (RP) software for use in the Resource Public Key Infrastructure (RPKI) in the context of securing Internet routing. It cites requirements that appear in several RPKI RFCs, making it easier for implementers to become aware of these requirements that are segmented with orthogonal functionalities.

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 <https://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 April 9, 2020.

Copyright Notice

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

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Fetching and Caching RPKI Repository Objects . . . . .	3
2.1. TAL Acquisition and Processing . . . . .	4
2.2. Locating RPKI Objects Using Authority and Subject Information Extensions . . . . .	4
2.3. Dealing with Key Rollover . . . . .	4
2.4. Dealing with Algorithm Transition . . . . .	4
2.5. Strategies for Efficient Cache Maintenance . . . . .	5
3. Certificate and CRL Processing . . . . .	5
3.1. Verifying Resource Certificate and Syntax . . . . .	5
3.2. Certificate Path Validation . . . . .	5
3.3. CRL Processing . . . . .	6
4. Processing RPKI Repository Signed Objects . . . . .	6
4.1. Basic Signed Object Syntax Checks . . . . .	6
4.2. Syntax and Validation for Each Type of Signed Object . .	6
4.2.1. Manifest . . . . .	6
4.2.2. ROA . . . . .	7
4.2.3. Ghostbusters . . . . .	7
4.2.4. Verifying BGPsec Router Certificate . . . . .	7
4.3. How to Make Use of Manifest Data . . . . .	7
4.4. What to Do with Ghostbusters Information . . . . .	8
5. Distributing Validated Cache . . . . .	8
6. Local Control . . . . .	8
7. Security Considerations . . . . .	9
8. IANA Considerations . . . . .	9
9. Acknowledgements . . . . .	9
10. References . . . . .	9
10.1. Normative References . . . . .	9
10.2. Informative References . . . . .	11
Authors' Addresses . . . . .	12

## 1. Introduction

The RPKI Relying Party (RP) software is used by network operators and others to acquire and verify Internet Number Resource (INR) data stored in the RPKI repository system. RPKI data, when verified, allow an RP to verify assertions about which Autonomous Systems (ASes) are authorized to originate routes for IP address prefixes. RPKI data also establishes binding between public keys and BGP routers, and indicates the AS numbers that each router is authorized to represent.

Noting that the essential requirements imposed on RPs to support securing Internet routing ([RFC6480]) are scattered throughout numerous RFC documents that are protocol specific or provide best practices, as follows:

- RFC 6481 (Repository Structure)
- RFC 6482 (ROA format)
- RFC 6486 (Manifests)
- RFC 6487 (Certificate and CRL profile)
- RFC 6488 (RPKI Signed Objects)
- RFC 6489 (Key Rollover)
- RFC 6810 (RPKI to Router Protocol)
- RFC 6916 (Algorithm Agility)
- RFC 7935 (Algorithms)
- RFC 8209 (Router Certificates)
- RFC 8210 (RPKI to Router Protocol, Version 1)
- RFC 8360 (Certificate Validation Procedure)
- RFC 8630 (Trust Anchor Locator)

This makes it hard for an implementer to be confident that he/she has addressed all of these generalized requirements. Besides, software engineering calls for how to segment the RP system into components with orthogonal functionalities, so that those components could be distributed across the operational timeline of the user. Taxonomy of generalized RP requirements is going to help have 'the role of the RP' well framed.

To consolidate RP requirements in one document, with pointers to all the relevant RFCs, this document outlines a set of baseline requirements imposed on RPs and provides a single reference point for requirements for RP software for use in the RPKI, as segmented with orthogonal functionalities:

- o Fetching and Caching RPKI Repository Objects
- o Processing Certificates and CRLs
- o Processing RPKI Repository Signed Objects
- o Distributing Validated Cache of the RPKI Data

This document will be update to reflect new or changed requirements as these RFCs are updated, or new RFCs are written.

## 2. Fetching and Caching RPKI Repository Objects

RP software uses synchronization mechanisms supported by targeted repositories (e.g., [rsync], RRDP [RFC8182]) to download all RPKI changed data objects in the repository system and cache them locally. The software validates the RPKI data and uses it to generate authenticated data identifying which ASes are authorized to originate



routes for address prefixes, and which routers are authorized to sign BGP updates on behalf of ASes.

## 2.1. TAL Acquisition and Processing

In the RPKI, each RP chooses its own set of trust anchors (TAs). Consistent with the extant INR allocation hierarchy, the IANA and/or the five RIRs are obvious candidates to be default TAs for the RP.

An RP does not retrieve TAs directly. A set of Trust Anchor Locators (TALs) is used by each RP to retrieve and verify the authenticity of each TA.

TAL acquisition and processing are specified in Section 3 of [RFC8630].

## 2.2. Locating RPKI Objects Using Authority and Subject Information Extensions

The RPKI repository system is a distributed one, consisting of multiple repository instances. Each repository instance contains one or more repository publication points. An RP discovers publication points using the Subject Information Access (SIA) and the Authority Information Access (AIA) extensions from (validated) certificates.

Section 5 of [RFC6481] specifies how an RP locates all RPKI objects by using the SIA and AIA extensions. Detailed specifications of SIA and AIA extensions in a resource certificate are described in Section 4 of [RFC6487].

## 2.3. Dealing with Key Rollover

An RP takes the key rollover period into account with regard to its frequency of synchronization with RPKI repository system.

RP requirements in dealing with key rollover are described in Section 3 of [RFC6489] and Section 3 of [RFC8634].

## 2.4. Dealing with Algorithm Transition

The set of cryptographic algorithms used with the RPKI is expected to change over time. Each RP is expected to be aware of the milestones established for the algorithm transition and what actions are required at every juncture.

RP requirements for dealing with algorithm transition are specified in Section 4 of [RFC6916].

## 2.5. Strategies for Efficient Cache Maintenance

Each RP is expected to maintain a local cache of RPKI objects. The cache needs to be as up to date and consistent with repository publication point data as the RP's frequency of checking permits.

The last paragraph of Section 5 of [RFC6481] provides guidance for maintenance of a local cache.

## 3. Certificate and CRL Processing

The RPKI make use of X.509 certificates and CRLs, but it profiles these standard formats [RFC6487]. The major change to the profile established in [RFC5280] is the mandatory use of a new extension to X.509 certificate [RFC3779].

### 3.1. Verifying Resource Certificate and Syntax

Certificates in the RPKI are called resource certificates, and they are required to conform to the profile [RFC6487]. An RP is required to verify that a resource certificate adheres to the profile established by Section 4 of [RFC6487]. This means that all extensions mandated by Section 4.8 of [RFC6487] must be present and value of each extension must be within the range specified by this RFC. Moreover, any extension excluded by Section 4.8 of [RFC6487] must be omitted.

Section 7.1 of [RFC6487] gives the procedure that the RP should follow to verify resource certificate and syntax.

### 3.2. Certificate Path Validation

The INRs in issuer's certificate are required to encompass the INRs in the subject's certificate. This is one of necessary principles of certificate path validation in addition to cryptographic verification i.e., verification of the signature on each certificate using the public key of the parent certificate).

Section 7.2 of [RFC6487] gives the procedure that the RP should follow to perform certificate path validation.

Certificate Authorities that want to reduce aspects of operational fragility will migrate to the new OIDs [RFC8360], informing the RP of using an alternative RPKI validation algorithm. An RP is expected to support the amended procedure to handle with accidental over-claim.

### 3.3. CRL Processing

The CRL processing requirements imposed on CAs and RP are described in Section 5 of [RFC6487]. CRLs in the RPKI are tightly constrained; only the AuthorityKeyIdentifier and CRLNumber extensions are allowed, and they are required to be present. No other CRL extensions are allowed, and no CRLentry extensions are permitted. RPs are required to verify that these constraints have been met. Each CRL in the RPKI must be verified using the public key from the certificate of the CA that issued the CRL.

In the RPKI, RPs are expected to pay extra attention when dealing with a CRL that is not consistent with the Manifest associated with the publication point associated with the CRL.

Processing of a CRL that is not consistent with a manifest is a matter of local policy, as described in the fourth paragraph of Section 6.6 of [RFC6486].

## 4. Processing RPKI Repository Signed Objects

### 4.1. Basic Signed Object Syntax Checks

Before an RP can use a signed object from the RPKI repository, the RP is required to check the signed object syntax.

Section 3 of [RFC6488] lists all the steps that the RP is required to execute in order to validate the top level syntax of a repository signed object.

Note that these checks are necessary, but not sufficient. Additional validation checks must be performed based on the specific type of signed object.

### 4.2. Syntax and Validation for Each Type of Signed Object

#### 4.2.1. Manifest

To determine whether a manifest is valid, the RP is required to perform manifest-specific checks in addition to those specified in [RFC6488].

Specific checks for a Manifest are described in Section 4 of [RFC6486]. If any of these checks fails, indicating that the manifest is invalid, then the manifest will be discarded and treated as though no manifest were present.

#### 4.2.2. ROA

To validate a ROA, the RP is required perform all the checks specified in [RFC6488] as well as the additional ROA-specific validation steps. The IP address delegation extension [RFC3779] present in the end-entity (EE) certificate (contained within the ROA), must encompass each of the IP address prefix(es) in the ROA.

More details for ROA validation are specified in Section 4 of [RFC6482].

#### 4.2.3. Ghostbusters

The Ghostbusters Record is optional; a publication point in the RPKI can have zero or more associated Ghostbuster Records. If a CA has at least one Ghostbuster Record, RP is required to verify that this Ghostbusters Record conforms to the syntax of signed object defined in [RFC6488].

The payload of this signed object is a (severely) profiled vCard. An RP is required to verify that the payload of Ghostbusters conforms to format as profiled in [RFC6493].

#### 4.2.4. Verifying BGPsec Router Certificate

A BGPsec Router Certificate is a resource certificate, so it is required to comply with [RFC6487]. Additionally, the certificate must contain an AS Identifier Delegation extension, and must not contain an IP Address Delegation extension. The validation procedure used for BGPsec Router Certificates is identical to the validation procedure described in Section 7 of [RFC6487], but using the constraints applied come from specification of Section 7 of [RFC8209].

Note that the cryptographic algorithms used by BGPsec routers are found in [RFC8208]. Currently, the algorithms specified in [RFC8208] and [RFC7935] are different. BGPsec RPs will need to support algorithms that are used to validate BGPsec signatures as well as the algorithms that are needed to validate signatures on BGPsec certificates, RPKI CA certificates, and RPKI CRLs.

#### 4.3. How to Make Use of Manifest Data

For a given publication point, the RP ought to perform tests, as specified in Section 6.1 of [RFC6486], to determine the state of the Manifest at the publication point. A Manifest can be classified as either valid or invalid, and a valid Manifest is either current and

stale. An RP decides how to make use of a Manifest based on its state, according to local (RP) policy.

If there are valid objects in a publication point that are not present on a Manifest, [RFC6486] does not mandate specific RP behavior with respect to such objects. However, most RP software ignores such objects and the authors of this document suggest this behavior be adopted uniformly.

In the absence of a Manifest, an RP is expected to accept all valid signed objects present in the publication point. If a Manifest is stale or invalid (see [RFC6486]) and an RP has no way to acquire a more recently valid Manifest, the RP is expected to contact the repository manager via Ghostbusters record and thereafter make decision according to local (RP) policy.

#### 4.4. What to Do with Ghostbusters Information

An RP may encounter a stale Manifest or CRL, or an expired CA certificate or ROA at a publication point. An RP is expected to use the information from the Ghostbusters record to contact the maintainer of the publication point where any stale/expired objects were encountered. The intent here is to encourage the relevant CA and/or repository manager to update the slate or expired objects.

#### 5. Distributing Validated Cache

On a periodic basis, BGP speakers within an AS request updated validated origin AS data and router/ASN data from the validated cache of RPKI data. The RP may either transfer the validated data to the BGP speakers directly, or it may transfer the validated data to a cache server that is responsible for provisioning such data to BGP speakers. The specification of the protocol designed to deliver validated cache data to a BGP Speaker is provided in [RFC6810] and [RFC8210].

#### 6. Local Control

ISPs may want to establish a local view of exceptions to the RPKI data in the form of local filters and additions. For instance, a network operator might wish to make use of a local override capability to protect routes from adverse actions [RFC8211]. The mechanisms developed to provide this capability to network operators are called "Simplified Local Internet Number Resource Management with the RPKI (SLURM)". If an ISP wants to implement SLURM, its RP system can follow the instruction specified in [RFC8416].

## 7. Security Considerations

The RP links the RPKI provisioning side and the routing system, establishing the local view of global RPKI data to BGP speakers. The security of the RP is critical to BGP messages exchanging. The RP implementation is expected to offer cache backup management to facilitate recovery from outage state. The RP implementation also should support application of secure transport (e.g., IPsec [RFC4301]) that is able to protect validated cache delivery in a unsafe environment.

## 8. IANA Considerations

This document has no actions for IANA.

## 9. Acknowledgements

The authors thank David Mandelberg, Wei Wang, Tim Bruijnzeels, George Michaelson and Oleg Muravskiy for their review, feedback and editorial assistance in preparing this document.

## 10. References

### 10.1. Normative References

- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <<https://www.rfc-editor.org/info/rfc3779>>.
- [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, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <<https://www.rfc-editor.org/info/rfc6481>>.
- [RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <<https://www.rfc-editor.org/info/rfc6482>>.

- [RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <<https://www.rfc-editor.org/info/rfc6486>>.
- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <<https://www.rfc-editor.org/info/rfc6487>>.
- [RFC6488] Lepinski, M., Chi, A., and S. Kent, "Signed Object Template for the Resource Public Key Infrastructure (RPKI)", RFC 6488, DOI 10.17487/RFC6488, February 2012, <<https://www.rfc-editor.org/info/rfc6488>>.
- [RFC6493] Bush, R., "The Resource Public Key Infrastructure (RPKI) Ghostbusters Record", RFC 6493, DOI 10.17487/RFC6493, February 2012, <<https://www.rfc-editor.org/info/rfc6493>>.
- [RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <<https://www.rfc-editor.org/info/rfc6810>>.
- [RFC7935] Huston, G. and G. Michaelson, Ed., "The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure", RFC 7935, DOI 10.17487/RFC7935, August 2016, <<https://www.rfc-editor.org/info/rfc7935>>.
- [RFC8208] Turner, S. and O. Borchert, "BGPsec Algorithms, Key Formats, and Signature Formats", RFC 8208, DOI 10.17487/RFC8208, September 2017, <<https://www.rfc-editor.org/info/rfc8208>>.
- [RFC8209] Reynolds, M., Turner, S., and S. Kent, "A Profile for BGPsec Router Certificates, Certificate Revocation Lists, and Certification Requests", RFC 8209, DOI 10.17487/RFC8209, September 2017, <<https://www.rfc-editor.org/info/rfc8209>>.
- [RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <<https://www.rfc-editor.org/info/rfc8210>>.

- [RFC8360] Huston, G., Michaelson, G., Martinez, C., Bruijnzeels, T., Newton, A., and D. Shaw, "Resource Public Key Infrastructure (RPKI) Validation Reconsidered", RFC 8360, DOI 10.17487/RFC8360, April 2018, <<https://www.rfc-editor.org/info/rfc8360>>.
- [RFC8630] Huston, G., Weiler, S., Michaelson, G., Kent, S., and T. Bruijnzeels, "Resource Public Key Infrastructure (RPKI) Trust Anchor Locator", RFC 8630, DOI 10.17487/RFC8630, August 2019, <<https://www.rfc-editor.org/info/rfc8630>>.

## 10.2. Informative References

- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<https://www.rfc-editor.org/info/rfc6480>>.
- [RFC6489] Huston, G., Michaelson, G., and S. Kent, "Certification Authority (CA) Key Rollover in the Resource Public Key Infrastructure (RPKI)", BCP 174, RFC 6489, DOI 10.17487/RFC6489, February 2012, <<https://www.rfc-editor.org/info/rfc6489>>.
- [RFC6916] Gagliano, R., Kent, S., and S. Turner, "Algorithm Agility Procedure for the Resource Public Key Infrastructure (RPKI)", BCP 182, RFC 6916, DOI 10.17487/RFC6916, April 2013, <<https://www.rfc-editor.org/info/rfc6916>>.
- [RFC8182] Bruijnzeels, T., Muravskiy, O., Weber, B., and R. Austein, "The RPKI Repository Delta Protocol (RRDP)", RFC 8182, DOI 10.17487/RFC8182, July 2017, <<https://www.rfc-editor.org/info/rfc8182>>.
- [RFC8211] Kent, S. and D. Ma, "Adverse Actions by a Certification Authority (CA) or Repository Manager in the Resource Public Key Infrastructure (RPKI)", RFC 8211, DOI 10.17487/RFC8211, September 2017, <<https://www.rfc-editor.org/info/rfc8211>>.
- [RFC8416] Ma, D., Mandelberg, D., and T. Bruijnzeels, "Simplified Local Internet Number Resource Management with the RPKI (SLURM)", RFC 8416, DOI 10.17487/RFC8416, August 2018, <<https://www.rfc-editor.org/info/rfc8416>>.



[RFC8634] Weis, B., Gagliano, R., and K. Patel, "BGPsec Router Certificate Rollover", BCP 224, RFC 8634, DOI 10.17487/RFC8634, August 2019, <<https://www.rfc-editor.org/info/rfc8634>>.

[rsync] "rsync web page", <<http://rsync.samba.org/>>.

#### Authors' Addresses

Di Ma  
ZDNS  
4 South 4th St. Zhongguancun  
Haidian, Beijing 100190  
China

Email: [madi@zdns.cn](mailto:madi@zdns.cn)

Stephen Kent  
Independent

Email: [kent@alum.mit.edu](mailto:kent@alum.mit.edu)

Network Working Group  
Internet-Draft

Updates: 6841, 8182 (if approved)

Intended status: Standards Track Internet Initiative Japan & Arrcus, Inc.

Expires: October 27, 2020

T. Bruijnzeels  
NLnet Labs

R. Bush

G. Michaelson

APNIC

April 25, 2020

Resource Public Key Infrastructure (RPKI) Repository Requirements  
draft-sidrops-bruijnzeels-deprecate-rsync-01

Abstract

This document formulates a plan of a phased transition to a state where RPKI repositories and Relying Party software performing RPKI Validation will use the RPKI Repository Delta Protocol (RRDP) [RFC8182] as the only mandatory to implement access protocol.

In short this plan consists of the following phases.

In phase 0, today's deployment, RRDP is supported by most, but not all Repositories, and most but not all RP software.

In the proposed phase 1 RRDP will become mandatory to implement for Repositories, in addition to rsync. This phase can start as soon as this document is published.

Once the proposed updates are implemented by all Repositories phase 2 will start. In this phase RRDP will become mandatory to implement for all RP software, and rsync must no longer be used.

Measurements will need to be done to help determine when it will be safe to transition to the final phase of this plan. During this phase Repositories will no longer be required to provide rsync access for RPKI validation purposes. However, they may still provide rsync access for direct access to files for other purposes, if desired, at a best effort basis.

Although this document currently includes descriptions and updates to RFCs for each of these phases, we may find that it will be beneficial to have separate documents for the plan, and each phase, so that it might be more clear to all when the updates to RFCs take effect.

## 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 <https://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 October 27, 2020.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

## Table of Contents

1. Requirements notation . . . . .	3
2. Motivation . . . . .	3
3. Plan . . . . .	4
3.1. Phase 0 - RPKI repositories support rsync, and optionally RRDP . . . . .	4
3.2. Phase 1 - RPKI repositories support both rsync and RRDP .	4
3.2.1. Current Support for RRDP in Repository Software . . .	4
3.2.2. Updates to RFC 6481 . . . . .	5
3.2.3. Measurements . . . . .	6
3.3. Phase 2 - All RP software prefers RRDP . . . . .	6
3.3.1. RRDP support in Relying Party software . . . . .	6
3.3.2. Updates to RFC 8182 . . . . .	6
3.3.3. Measurements . . . . .	7
3.4. Phase 3 - RPKI repositories support RRDP, and optionally	

rsync . . . . .	7
3.4.1. Updates to RFC 6481 . . . . .	7
4. Rsync URIs as object identifiers . . . . .	8
5. IANA Considerations . . . . .	9
6. Security Considerations . . . . .	9
7. Acknowledgements . . . . .	9
8. Normative References . . . . .	9
Authors' Addresses . . . . .	10

## 1. Requirements notation

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

## 2. Motivation

The Resource Public Key Infrastructure (RPKI) [RFC6480] as originally defined uses rsync as its distribution protocol, as outlined in [RFC6481]. Later, the RPKI Repository Delta Protocol (RRDP) [RFC8182] was designed to provide an alternative. In order to facilitate incremental deployment RRDP has been deployed as an additional optional protocol, while rsync was still mandatory to implement.

A number of issues observed with rsync motivated the design of RRDP, e.g.:

- o rsync is CPU and memory heavy, and easy to DoS
- o rsync library support is lacking
- o rsync makes it somewhat difficult to publish sets of object atomically

RRDP was designed to leverage HTTPS CDN infrastructure to provide RPKI Repository content in a resilient way, while reducing the load on the Repository server. It supports that updates are published as atomic deltas, which can help prevent most of the issues described in section 6 of [RFC6486].

For a longer discussion please see section 1 of [RFC8182].

In conclusion: we believe that RRDP is the better solution. Therefore, this document outlines a transition plan where RRDP

becomes mandatory to implement, and rsync becomes optional and eventually deprecated.

### 3. Plan

Changing the RPKI infrastructure to rely on RRDP instead of rsync is a delicate operation. There is current deployment of Certification Authorities, Repository Servers and Relying Party software which relies on rsync, and which may not yet support RRDP.

Therefore we need to have a plan that ultimately updates the relevant RFCs, but which uses a phased approach combined with measurements to limit the operational impact of doing this to (almost) zero.

The general outline of the plan is as follows. We will describe each step in more detail below.

Phase	Description
0	RPKI repositories support rsync, and optionally RRDP
1	RPKI repositories support both rsync and RRDP
2	All RP software prefers RRDP
3	RPKI repositories support RRDP, and optionally rsync

#### 3.1. Phase 0 - RPKI repositories support rsync, and optionally RRDP

This is the situation at the time of writing this document. Relying Parties can prefer RRDP over rsync today, but they need to support rsync until all RPKI repositories support RRDP. Therefore all repositories should support RRDP at their earliest convenience.

#### 3.2. Phase 1 - RPKI repositories support both rsync and RRDP

During this phase we will make RRDP mandatory to support for Repository Servers, and measure whether the deployed Repository Servers have been upgraded to do so, in as far as they don't support RRDP already.

##### 3.2.1. Current Support for RRDP in Repository Software

The currently known support for RRDP for repositories is as follows:

Repository Implementation	Support for RRDP
afrinic	yes
apnic	yes
arin	yes
lacnic	planned
ripe ncc	yes
Dragon Research Labs	yes (1,2)
krill	yes (1)

(1) in use at various National Internet Registries, as well as other resource holders under RIRs. (2) not all organizations using this software have upgraded to using RRDP.

### 3.2.2. Updates to RFC 6481

During this phase the updates are applied to section 3 of [RFC6481].

#### OLD:

- o The publication repository SHOULD be hosted on a highly available service and high-capacity publication platform.
- o The publication repository MUST be available using rsync [RFC5781] [RSYNC]. Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

#### NEW:

- o The publication repository MUST be available using the RPKI Repository Delta Protocol [RFC8182]. The RRDP server SHOULD be hosted on a highly available platform.
- o The publication repository MUST be available using rsync [RFC5781] [RSYNC]. The rsync server SHOULD be hosted on a highly available platform.
- o Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

### 3.2.3. Measurements

We can find out whether all RPKI repositories support RRDP by running (possibly) modified Relying Party software that keeps track of this.

When it is found that Repositories do not yet support RRDP, outreach should be done to them individually. Since the number of Repositories is fairly low, and it is in their interest to run RRDP because it addresses availability concerns, we have confidence that we will find these Repositories willing to make changes.

### 3.3. Phase 2 - All RP software prefers RRDP

Once all Repositories support RRDP we can proceed to make RRDP mandatory to implement for Relying Party software.

#### 3.3.1. RRDP support in Relying Party software

The currently known support for RRDP in Relying Party software is as follows:

Relying Party Implementation	RRDP	version	since
FORT	yes	?	?
OctoRPKI	yes	?	?
rcynic	yes	?	?
RIPE NCC RPKI Validator 2.x	yes	?	?
RIPE NCC RPKI Validator 3.x	yes	?	?
Routinator	yes	0.6.0	Sep 2019
rpki-client	no	?	?
RPSTIR	yes	?	?

The authors kindly request Relying Party software implementers to let us know in which version of their tool support for RRDP was introduced, and when that version was released.

#### 3.3.2. Updates to RFC 8182

From this phase onwards the updates are applied to section 3.4.1 of [RFC8182].

OLD: When a Relying Party performs RPKI validation and learns about a valid certificate with an SIA entry for the RRDP protocol, it SHOULD use this protocol as follows.

NEW: When a Relying Party performs RPKI validation and learns about a valid certificate with an SIA entry for the RRDP protocol, it MUST use this protocol. It MUST NOT depend on object retrieval for this certificate over rsync for validation, although it MAY still use rsync access for other purposes under the understanding that availability is not guaranteed.

### 3.3.3. Measurements

Although the tools may support RRDP, users will still need to install updated versions of these tools in their infrastructure. Any Repository operator can measure this transition by observing access to their RRDP and rsync repositories respectively.

But even after new versions have been available, it is expected that there will be long, low volume, tail of users who did not upgrade and still depend on rsync.

It is hard to quantify here now, what would be an acceptable moment to conclude that it's safe to move to the next phase and make rsync optional. A parallel to the so-called DNS Flag Day comes to mind.

### 3.4. Phase 3 - RPKI repositories support RRDP, and optionally rsync

The end goal of this phase is that there will be no operational dependencies on rsync for Repositories, although they MAY still choose to operate rsync at a best effort basis.

#### 3.4.1. Updates to RFC 6481

From this phase onwards these updates are applied to section 3 of [RFC6481] as it was updated during Phase 2 described above:

OLD:

- o The publication repository MUST be available using the RPKI Repository Delta Protocol [RFC8182]. The RRDP server SHOULD be hosted on a highly available platform.
- o The publication repository MUST be available using rsync [RFC5781] [RSYNC]. The rsync server SHOULD be hosted on a highly available platform.
- o Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.



## NEW:

- o The publication repository MUST be available using the RPKI Repository Delta Protocol [RFC8182]. The RRDP server SHOULD be hosted on a highly available platform.
- o The publication repository MAY be available using rsync [RFC5781] [RSYNC].
- o Support of additional retrieval mechanisms is the choice of the repository operator. The supported retrieval mechanisms MUST be consistent with the accessMethod element value(s) specified in the SIA of the associated CA or EE certificate.

## 4. Rsync URIs as object identifiers

If and when RPKI Repositories no longer need to support rsync, this begs the question whether rsync should still be used in URIs used in RPKI objects.

[RFC6481] defines a profile for the Resource Certificate Repository Structure. In this profile objects are identified through rsync URIs. E.g. a CA certificate has an Subject Information Access descriptor which uses an rsync URI to identify its manifest [RFC6486]. The manifest enumerates the relative names and hashes for all objects published under the private key of the CA certificate. The full rsync URI identifiers for each object can be resolved relative to the manifest URI.

Though it would be possible in principle to build up an RPKI tree hierarchy of objects based on key identifiers and hashes [RFC8488], most Relying Party implementations have found it very useful to use rsync URIs for this purpose. Furthermore, these identifiers make it much easier to name object in case of validation problems, which help operators to address issues.

For these reasons, RRDP still includes rsync URIs in the definition of the publish, update and withdraw elements in the snapshot and delta files that it uses. See section 3.5 of [RFC8182]. Thus, objects retrieved through RRDP can be mapped easily to files and URIs, similar to as though rsync would have been used to retrieve them.

Even though objects are no longer guaranteed to be available over rsync, we still use rsync as the mandatory scheme in the CRL Distribution Points, Authority Information Access, and Subject Information Access defined in [RFC6487]. Changing this would introduce breaking changes which make deployment very hard indeed: we

would need to invent an alternative naming scheme, which would need to be supported by all Relying Parties, before Certification Authorities can issue any certificate or RPKI signed objects using these schemes.

Furthermore, it is very convenient to have direct access to RPKI objects using rsync for troubleshooting, debugging and research purposes. Therefore Repository operators MAY still choose to make an rsync repository available for these purposes.

## 5. IANA Considerations

This document has no IANA actions.

## 6. Security Considerations

TBD

## 7. Acknowledgements

TBD

## 8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5781] Weiler, S., Ward, D., and R. Housley, "The rsync URI Scheme", RFC 5781, DOI 10.17487/RFC5781, February 2010, <<https://www.rfc-editor.org/info/rfc5781>>.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<https://www.rfc-editor.org/info/rfc6480>>.
- [RFC6481] Huston, G., Loomans, R., and G. Michaelson, "A Profile for Resource Certificate Repository Structure", RFC 6481, DOI 10.17487/RFC6481, February 2012, <<https://www.rfc-editor.org/info/rfc6481>>.
- [RFC6486] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 6486, DOI 10.17487/RFC6486, February 2012, <<https://www.rfc-editor.org/info/rfc6486>>.

- [RFC6487] Huston, G., Michaelson, G., and R. Loomans, "A Profile for X.509 PKIX Resource Certificates", RFC 6487, DOI 10.17487/RFC6487, February 2012, <<https://www.rfc-editor.org/info/rfc6487>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8182] Bruijnzeels, T., Muravskiy, O., Weber, B., and R. Austein, "The RPKI Repository Delta Protocol (RRDP)", RFC 8182, DOI 10.17487/RFC8182, July 2017, <<https://www.rfc-editor.org/info/rfc8182>>.
- [RFC8488] Muravskiy, O. and T. Bruijnzeels, "RIPE NCC's Implementation of Resource Public Key Infrastructure (RPKI) Certificate Tree Validation", RFC 8488, DOI 10.17487/RFC8488, December 2018, <<https://www.rfc-editor.org/info/rfc8488>>.

## Authors' Addresses

Tim Bruijnzeels  
NLnet Labs

Email: [tim@nlnetlabs.nl](mailto:tim@nlnetlabs.nl)  
URI: <https://www.nlnetlabs.nl/>

Randy Bush  
Internet Initiative Japan & Arrcus, Inc.

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

George Michaelson  
APNIC

Email: [ggm@apnic.net](mailto:ggm@apnic.net)  
URI: <http://www.apnic.net>

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 7, 2020

R. Austein  
Arrcus  
R. Bush  
Arrcus & Internet Initiative Japan  
G. Huston  
G. Michaelson  
Asia Pacific Network Information Centre  
November 4, 2019

Resource Transfer in the Resource Public Key Infrastructure  
draft-ymbk-sidrops-transfer-00

Abstract

Transfer within the RPKI of actual address space and/or autonomous system number resources between two Internet registries (ISPs, RIRs, NIRs, etc.) is reasonably achievable for most useful operational needs. In this paper, we describe, at a high level, how this may be accomplished.

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 <https://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 May 7, 2020.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction and Terms . . . . .	2
2. A Simplistic Case . . . . .	3
2.1. Steps in Simple Case . . . . .	4
2.2. The Torn Euro Protocol . . . . .	4
3. A More Complex Case . . . . .	5
3.1. The Indirect Buyer . . . . .	6
3.2. The Difference Between Buyer and Seller Chain . . . . .	8
4. Transfer in the Absence of a Common Ancestor . . . . .	8
5. Transfer in process: Resources Change Forced from Above . . . . .	9
6. Security Considerations . . . . .	9
7. Acknowledgements . . . . .	9
8. IANA Considerations . . . . .	9
9. References . . . . .	10
9.1. Normative References . . . . .	10
9.2. Informative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction and Terms

To paraphrase the Introduction of [RFC6480], the "Resource Public Key Infrastructure (RPKI) represents the allocation hierarchy of IP address space and Autonomous System (AS) numbers; and is a distributed repository system for storing and disseminating the data objects that comprise the RPKI, as well as other signed objects necessary for improved routing security."

An Internet Registry (IR) is the IANA, a Regional Internet Registry (RIR), a National Internet Registry (NIR), a Local Internet Registry (LIR), a Internet Service Provider (ISP), or an end site which may hold IP resources and is the subject of one or more certificates using [RFC3779] extensions in the Resource Public Key Infrastructure (RPKI), see [RFC6480].

It is increasingly necessary to transfer resources between resource-holding entities in the RPKI, to do so without violating contracts, policies, etc., and while maintaining operational reliability and administrative accuracy with minimal administrative overhead.

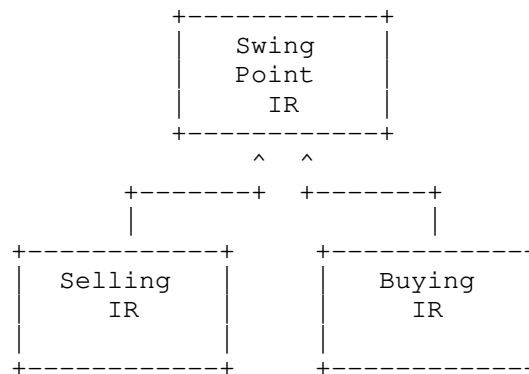


Fig 1. The Simplest Example of Seller, Buyer, and Swing Point

Seller and Buyer are used to describe the end parties to a transfer, the selling IR transferring the resource(s) to the buying IR. For the purposes of this document, the terms seller and buyer are used, although layer nine considerations may require less commercial formal roles.

Transfer is the sale and corresponding purchase of literal address space or autonomous system numbers between two parties. The seller relinquishing some amount of resource and the buyer being allocated a similar amount but not the same literal address space, is not a transfer, and is not further considered here.

A Swing Point is the IR at the lowest point in the RPKI hierarchy which the seller and buyer have as a common parent and which has agreed to be used as the agent of transfer.

While there is no automated method for the RPKI to assist the parties to a transaction in determining that all business and policy aspects of a transaction are satisfied, these layer eight and nine issues can be resolved using normal business practices and therefore not addressed in this document.

## 2. A Simplistic Case

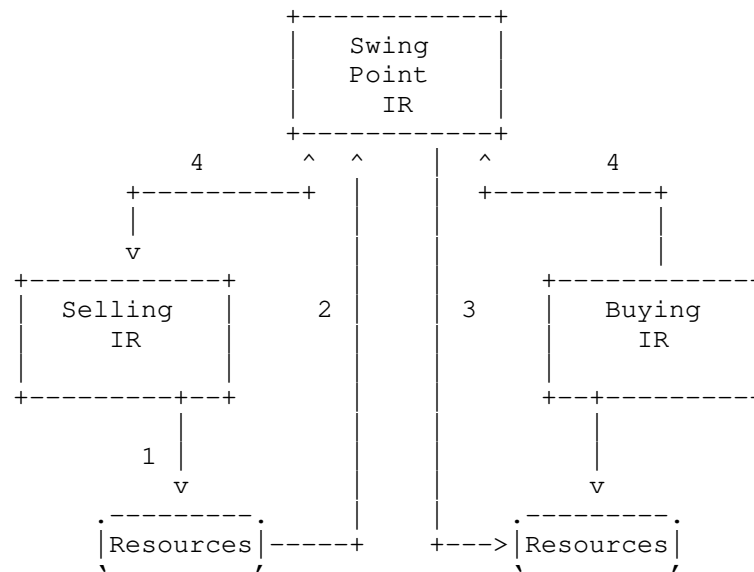


Fig 2. Steps in a Simple Transfer

### 2.1. Steps in Simple Case

As a formal business relationship between all parties to a transfer provides a level of trust which allows simple transactions, we first consider the simple case where the seller and the buyer are both directly known to the swing point, see Figure 1.

The transfer is done in the following steps (see Figure 2):

1. The seller creates a certificate describing the subset of the seller's resources which are to be transferred.
2. The seller tells the swing point that it wishes to transfer the resources described by the certificate to the buyer.
3. The swing point issues a new expanded certificate to the buyer describing the buyer's old holdings plus the new resources.
4. When the seller and the buyer are comfortable that both the technical aspects (customers swung, routing done, etc.) and the business aspects of the transfer have been accomplished, they inform the swing point which then issues a new certificate to the seller, having removed the transferred resources.

### 2.2. The Torn Euro Protocol

Due to issues of cancellation, reneging, and fraud, step 4 above, where the seller and the buyer tell the swing point that the deal is done, needs to be formal in some fashion. For this purpose, we

envision a yet to be described torn Euro protocol, where the buyer and the seller each hold one half of a virtual torn Euro note, and the swing point believes the transaction to be complete when it has received both halves and they match.

This protocol has yet to be described.

### 3. A More Complex Case

What happens when the seller is not a direct customer of the swing point as in Figure ?.

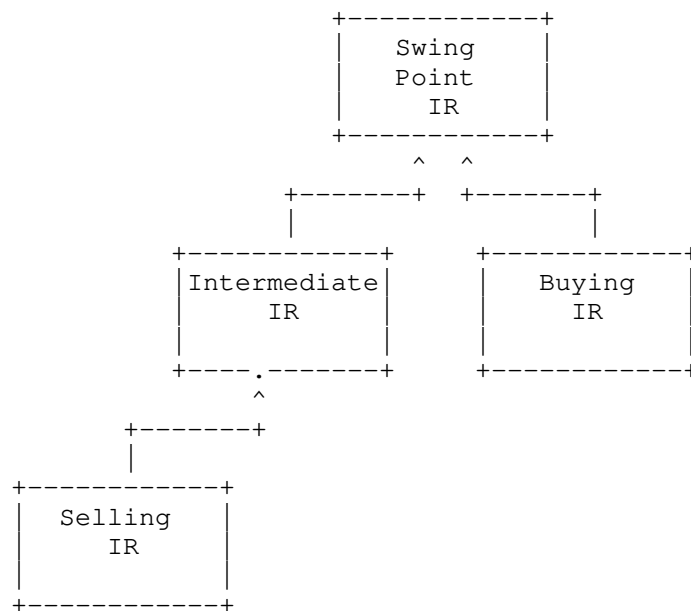


Figure 3. The Case of The Indirect Seller

The swing point needs to be assured that it is contractually able to move the resource given its relationship to the Other IR. As RFC 3779 extensions do not codify business issues such as PI/PA, and rights to resell, this has to be handled out of band; there is no way to automate it. This is part of today's IR address space management process and will continue to be handled manually.

Therefore the process is the same as for the simple case, except that, before issuing the expanded certificate to the buyer in step 3, the swing point must assure itself that policy and contractual issues have been addressed. The swing point might be well-advised to contact the intermediate IR and gain its consent, possibly with the assistance of the seller. The bottom line is that the swing point



does own/control the resource being transferred, and therefore has the prerogative to act based on its perception of the liabilities it is incurring.

This freedom allowing the seller to be indirectly related to the swing point can accommodate more levels of indirection. It is the swing point's obligation to perform diligence on the iterative financial, contractual, and policy obligations of the relationships down to the seller. Unfortunately, the RPKI can not automate this.

All certificates below the swing point down to and including the seller will need to be reissued with the appropriate reduction of resources. All certificated below the swing point down to and including the buyer will need to be [re-]issued to include the addition of the resource(s) being transferred to the buyer.

### 3.1. The Indirect Buyer

The case where the buyer is not directly known to the swing point is more difficult. Among other issues, the buyer may not be an existing resource holder at all, i.e. there may be no path down from the IANA root to the buyer. In this case, the buyer must explore the graph and choose an IR with which to contract a relationship. This can be both a business issue and a policy issue, e.g. can a buyer in Asia choose a parent which is, directly or indirectly, an ARIN customer?

The case where the buyer contracts directly to become a customer of the swing point has been explored above. What if the buyer becomes a grandchild of the swing point, as in Figure 4?

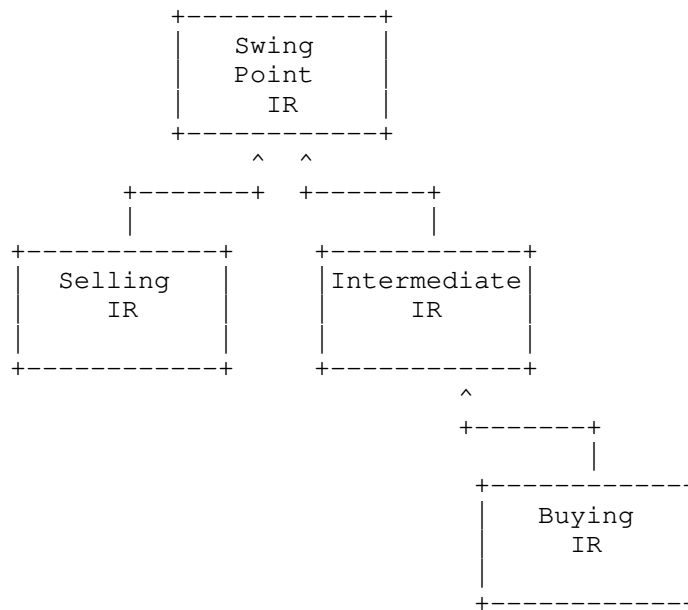


Figure 4. The Case of The Indirect Buyer

Somewhat analogously to the case of the indirect seller, the swing point has to iteratively verify that the IRs between it and the buyer are all willing to contractually and technically accept the resource(s) to be allocated to the buyer. But, in the case of the indirect buyer, the iterative conditions are much stronger. In the indirect seller case, the swing point has contractual control of the chain between it and the seller. In the case of the indirect buyer, all intermediate IRs between the swing point and the buyer must give business and technical consent to the transfer. The swing point can not directly or transitively force its child to issue a resource certificate to the buyer.

Things may not be as bad as they appear at first blush. The buyer is actually contracting with its parent, and part of that contract will presumably be that the parent agrees to issue the resource certificate to the buyer when it receives the resource from it's parent. And this presumably applies to the buyer's parent's relationship to a grandparent and so forth. On the other hand, the swing point has no mechanical way to test the willingness of the IRs on the buyer's indirect chain. But the swing point will need to know when the buyer is happy that it has received the resources.

### 3.2. The Difference Between Buyer and Seller Chain

Essentially, the difference between an indirect buyer chain and an indirect seller chain is that the swing point has the logical, though maybe not contractual, prerogative to pull address space from the seller's chain, but does not have the power to push it down the buyer's chain. All IRs on the buyer's chain must agree to certify downward toward the buyer.

### 4. Transfer in the Absence of a Common Ancestor

For political reasons, the current RPKI structure has no single root trust anchor. There are a number of trust anchors, e.g. the five RIRs which do not descend from the IANA. This creates considerable complexity and some risk for resource transfers between entities without a common ancestor.

To work around this problem, each RIR certifies a subsidiary Certification Authority for each other RIR to which it transfers resources, see Figure 5, and issues the transferred resources to that subsidiary CA.

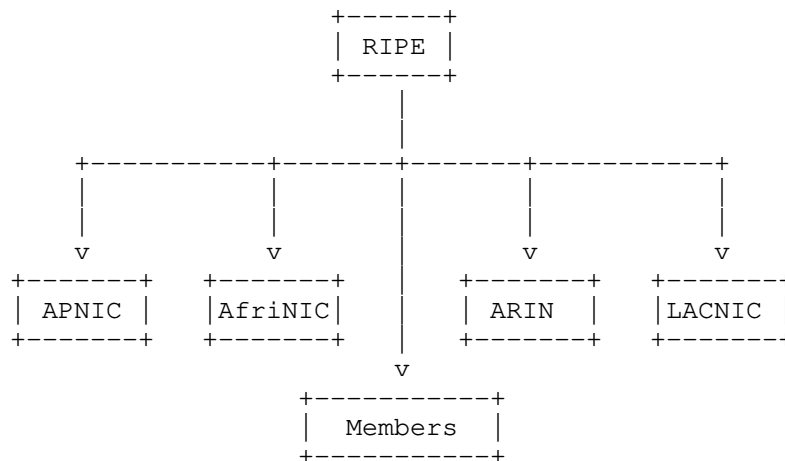


Figure 5: The RIRs each certify proxy CAs for all of the other RIRs.

But, to use the example of Figure 5, the APNIC CA to which RIPE issues resources is, in fact, run by APNIC under APNIC's Business Certificate PKI (see [RFC6492] Section 3) and uses an APNIC-provided publication point.

Thus APNIC has under its control, among other things, four CAs, one with resources from each of the other CAs. And similarly for each of the other RIRs.

So the swing point for a transfer from an APNIC member to a RIPE member is the APNIC CA. And an APNIC member holding resources originated by APNIC as well as resources transferred in from another RIR, e.g. RIPE, actually holds two resource certificates.

This could probably be made more complicated and brittle, but it would require serious effort.

#### 5. Transfer in process: Resources Change Forced from Above

Even though both seller and buyer have agreed to a transfer, the seller might try to not relinquish the resource, hoping to sell it more than once. Therefore it may become necessary to force closure for a non-compliant seller. In this case, a resource holding would be changed, shrunk, by force from above.

A 'normal' (i.e. what the RPKI design anticipated) resource shrinkage is initiated by the leaf resource holder and propagates upward toward the root of the tree. At no point in this process does a holder claim more than their parent believes they have.

When a resource is forcibly removed from 'above', the shrinkage propagates downward. Until the ultimate holder relinquishes the resource, at some point in the path down the tree a child holds more resources than its parent believes it does. As the protocol model is bottom initiated polling, see [RFC6492], the time window of exposure of this over-claiming can be relatively large.

#### 6. Security Considerations

Ghu only knows.

#### 7. Acknowledgements

Thanks to Steve Kent for comments.

#### 8. IANA Considerations

Nothing is required of the IANA; though it would make some things a lot simpler if the IANA was the root TA/CA of the entire tree.

## 9. References

### 9.1. Normative References

- [RFC6492] Huston, G., Loomans, R., Ellacott, B., and R. Austein, "A Protocol for Provisioning Resource Certificates", RFC 6492, DOI 10.17487/RFC6492, February 2012, <<http://www.rfc-editor.org/info/rfc6492>>.

### 9.2. Informative References

- [RFC3779] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, DOI 10.17487/RFC3779, June 2004, <<http://www.rfc-editor.org/info/rfc3779>>.
- [RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <<http://www.rfc-editor.org/info/rfc6480>>.

## Authors' Addresses

Rob Austein  
Arrcus, Inc

Email: [sra@hacitrn.net](mailto:sra@hacitrn.net)

Randy Bush  
Arrcus & Internet Initiative Japan  
5147 Crystal Springs  
Bainbridge Island, WA 98110  
US

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

Geoff Huston  
Asia Pacific Network Information Centre  
6 Cordelia St  
South Brisbane, QLD 4101  
AU

Email: [gih@apnic.net](mailto:gih@apnic.net)

George Michaelson  
Asia Pacific Network Information Centre  
6 Cordelia St  
South Brisbane, QLD 4101  
AU

Email: ggm@apnic.net