ï»¿

Internet Engineering Task Force (IETF)
Request for Comments: 8897
Category: Informational
ISSN: 2070-1721

D. Ma
ZDNS
S. Kent
Independent

September 2020

Requirements for Resource Public Key Infrastructure (RPKI) Relying Parties

Abstract

This document provides a single reference point for requirements for Relying Party (RP) software for use in the Resource Public Key Infrastructure (RPKI). It cites requirements that appear in several RPKI RFCs, making it easier for implementers to become aware of these requirements. Over time, this RFC will be updated to reflect changes to the requirements and guidance specified in the RFCs discussed herein.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8897.

Table of Contents

1. Introduction

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, allows an RP to verify assertions about which Autonomous Systems (ASes) are authorized to originate routes for IP address prefixes. RPKI data also establishes a binding between public keys and BGP routers and indicates the AS numbers that each router is authorized to represent.

The essential requirements imposed on RP software to support secure Internet routing [RFC6480] are scattered throughout numerous protocol-specific RFCs and Best Current Practice RFCs. The following RFCs define these requirements:

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)

The distribution of RPKI RP requirements across these 13 documents makes it hard for an implementer to be confident that he/she has addressed all of these requirements. Additionally, good software engineering practice may call for segmenting the RP system into components with orthogonal functionalities so that those components may be distributed. A taxonomy of the collected RP software requirements can help clarify the role of the RP.

To consolidate RP software 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. The requirements are organized into four groups:

* Fetching and Caching RPKI Repository Objects

* Processing Certificates and Certificate Revocation Lists (CRLs)

* Processing RPKI Repository Signed Objects

* Distributing Validated Cache of the RPKI Data

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

2. Fetching and Caching RPKI Repository Objects

RP software uses synchronization mechanisms supported by targeted repositories (e.g., [rsync] or RRDP [RFC8182]) to download RPKI signed objects from the repository system in order to update a local cache. These mechanisms download only those objects that have been added or replaced with new versions since the time when the RP most recently checked the repository. RP 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 specified ASes.

2.1. TAL Configuration and Processing

In the RPKI, each RP chooses a set of trust anchors (TAs). Consistent with the extant INR allocation hierarchy, the IANA and/or the five Regional Internet Registries (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 RP software to retrieve and verify the authenticity of each TA.

TAL configuration 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. RP software 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 RP software 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 Sections 4.8.8 and 4.8.7 of [RFC6487], respectively.

2.3. Dealing with Key Rollover

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

RP software requirements for 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 software 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 brought up to date and made consistent with the repository publication point data as frequently as allowed by repository publication points and by locally selected RP processing constraints.

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

3. Certificate and CRL Processing

The RPKI makes use of X.509 certificates and CRLs, but it profiles the standard formats described in [RFC6487]. The major change to the profile established in [RFC5280] is the mandatory use of a new extension in RPKI certificates, defined in [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 described in [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 the value of each extension must be within the range specified by [RFC6487]. Moreover, any extension excluded by Section 4.8 of [RFC6487] must be omitted.

Section 7.1 of [RFC6487] specifies the procedure that RP software follows when verifying extensions described in [RFC3779].

3.2. Certificate Path Validation

Initially, the INRs in the issuer's certificate are required to encompass the INRs in the subject's certificate. This is one of the 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] specifies the procedure that RP software should follow to perform certificate path validation.

Certification Authorities (CAs) that want to reduce aspects of operational fragility will migrate to the new OIDs [RFC8360], informing RP software to use an alternative RPKI validation algorithm. An RP is expected to support the amended procedure to handle accidental overclaiming, which is described in Section 4 of [RFC8360].

3.3. CRL Processing

The CRL processing requirements imposed on CAs and RPs are described in Section 5 of [RFC6487]. CRLs in the RPKI are tightly constrained; only the AuthorityKeyIdentifier (Section 4.8.3 of [RFC6487]) and CRLNumber (Section 5.2.3 of [RFC5280]) extensions are allowed, and they are required to be present. No other CRL extensions are allowed, and no CRLEntry extensions are permitted. RP software is 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 fifth 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, RP software is required to check the signed-object syntax.

Section 3 of [RFC6488] lists all the steps that RP software 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, as described in Section 4.2.

4.2. Syntax and Validation for Each Type of Signed Object

4.2.1. Manifest

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

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

4.2.2. ROA

To validate a Route Origin Authorization (ROA), RP software is required to perform all the checks specified in [RFC6488] as well as 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 Ghostbusters Records. If a CA has at least one Ghostbusters Record, RP software is required to verify that this Ghostbusters Record conforms to the syntax of signed objects defined in [RFC6488].

The payload of this signed object is a (severely) profiled vCard. RP software 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 (Section 4.8.11 of [RFC6487]) and must not contain an IP Address Delegation extension (Section 4.8.10 of [RFC6487]). The validation procedure used for BGPsec Router Certificates is analogous to the validation procedure described in Section 7 of [RFC6487], but it uses the constraints defined in Section 3 of [RFC8209].

Note that the cryptographic algorithms used by BGPsec routers are found in [RFC8608]. Currently, the algorithms specified in [RFC8608] and [RFC7935] are different. BGPsec RP software 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, RP software 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 or 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.

In the absence of a manifest, an RP is expected to accept all valid signed objects present in the publication point (see Section 6.2 of [RFC6486]). If a manifest is stale or invalid and an RP has no way to acquire a more recent valid manifest, the RP is expected to contact the repository manager via Ghostbusters Records and thereafter make decisions according to local (RP) policy (see Sections 6.3 and 6.4 of [RFC6486]).

4.4. What To Do with Ghostbusters Information

RP software 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 Records 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 stale 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 (local) 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 specifications of the protocol designed to deliver validated cache data to a BGP Speaker are 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

This document does not introduce any new security considerations; it is a resource for implementers. The RP links the RPKI provisioning side and the routing system, establishing a verified, local view of global RPKI data to BGP speakers. The security of the RP is critical for exchanging BGP messages. Each RP implementation is expected to offer cache backup management to facilitate recovery from outages. RP software should also support secure transport (e.g., IPsec [RFC4301]) that can protect validated cache delivery in an unsafe environment. This document highlights many validation actions applied to RPKI signed objects, an essential element of secure operation of RPKI security.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. 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>.

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC8608] Turner, S. and O. Borchert, "BGPsec Algorithms, Key Formats, and Signature Formats", RFC 8608, DOI 10.17487/RFC8608, June 2019,

<https://www.rfc-editor.org/info/rfc8608>.

[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>.

[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>.

9.2. 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>.

[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>.

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

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.

Authors' Addresses

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

Email: madi@zdns.cn

Stephen Kent
Independent

Email: kent@alum.mit.edu