SIDROPS
Internet-Draft
Intended status: Informational
Expires: 3 August 2026

J. Shi
Tsinghua University
L. Liu
L. Qin
Zhongguancun Laboratory
D. Li
Tsinghua University

30 January 2026

Considerations for On-Demand Retrieval of Multiple RPKI Object Types
draft-shi-sidrops-rpki-on-demand-00

Abstract

The Resource Public Key Infrastructure (RPKI) RFC6480 [RFC6481] relies on Publication Points (PPs) to distribute cryptographically verifiable objects. Under the current design, Relying Parties (RPs) retrieve the complete set of RPKI objects from repositories, even when their operational requirements are limited to a specific subset, for example, ROAs only. This all-or-nothing retrieval model may result in increased bandwidth consumption, higher synchronization latency, and unnecessary computational overhead. This document examines these challenges and discusses directions for enabling more selective and efficient access to RPKI data.

Table of Contents

1. Introduction

As securing the Internet' s routing infrastructure becomes increasingly critical, the Resource Public Key Infrastructure (RPKI) RFC6480 [RFC6481] provides a framework for validating the authenticity and integrity of routing information. RPKI relies on cryptographically verifiable objects published at Publication Points (PPs) [RFC8182], which are retrieved and processed by Relying Parties (RPs) [RFC6480] to support route validation decisions.

While RPKI provides a robust mechanism for publishing and validating routing authorization data, its current data distribution model assumes that RPs retrieve and process the complete repository contents. As the deployment of RPKI-based security mechanisms expands, both the volume and the diversity of data stored at PPs continue to grow. Current RP implementations do not provide sufficient support for selectively retrieving specific types of RPKI objects. Consequently, operators that require only a subset of RPKI functionality must still retrieve and process the complete repository contents, leading to increased bandwidth consumption, longer

synchronization latency, and additional computational overhead. For example, an AS that only requires Route Origin Authorization (ROA) for route origin validation will, when using current RPs, also download and process unrelated RPKI objects such as ASPA and MOA.

This document analyzes these limitations in the current RPKI data retrieval model and discusses directions for enabling selective, on-demand retrieval of RPKI objects to improve scalability and operational efficiency.

1.1. 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.

2. Terminology

ROA (Route Origin Authorization) [RFC6481]: A ROA is a digitally signed object that provides a means of verifying that an IP address block holder has authorized an Autonomous System (AS) to originate routes to one or more prefixes within the address block.

ASPA (Autonomous System Provider Authorization) [I-D.ietf-sidrops-aspa-verification]: An ASPA is a digitally signed object through which the issuer (the holder of an Autonomous System identifier), can authorize one or more other ASes as its upstream providers.

MOA (Mapping Origin Authorization) [I-D.ietf-sidrops-moa-profile]: It provides a means that the address holder can authorize an IPv6 mapping prefix to originate mapping for one or more IPv4 prefixes.

RP (Relying Party) [RFC6480]: A system that downloads and validates RPKI data from a repository.

PP (Publication Point) [RFC8182]: The location where RPKI objects are published and stored, accessible via rsync or RRDP.

RRDP (RPKI Repository Delta Protocol) [RFC8182]: A pull-based protocol used by RPs to synchronize with RPKI repositories.

3. Scope and Non-Goals

This document focuses on identifying limitations in the current RPKI data synchronization model with respect to selective and on-demand retrieval of RPKI objects between Publication Points (PPs) and Relying Parties (RPs). The scope of this document is limited to analyzing the problem space, operational impacts, and high-level directions for improvement. It does not specify new protocols, data formats, or concrete mechanisms.

In scope for this document are considerations related to the retrieval of different types of RPKI objects, such as ROAs, ASPA objects, and other signed objects, during RP synchronization. The document examines how existing synchronization mechanisms, including Rsync, RRDP, and Erik, currently assume full repository synchronization and discusses why this model may not scale efficiently as the diversity and volume of RPKI objects increase.

4. Problem Statement

As the diversity and volume of RPKI objects continue to grow, there is an increasing need for Relying Parties (RPs) to retrieve only the objects relevant to their operational requirements. However, the current RPKI data synchronization model assumes that RPs synchronize and process the complete repository contents, without support for selective or on-demand retrieval. As a result, some RPs may retrieve and process unnecessary data, increasing both bandwidth consumption and operational overhead.

Based on operational measurements, synchronizing a full repository snapshot currently requires approximately 1.1 GB of bandwidth, while incremental updates require around 10 MB. Assuming that emerging object types such as ASPA and MOA generate data volumes comparable to existing ROA objects, an RP that only requires ROA information would still need to download approximately 3.3 GB for a full synchronization and 30 MB for incremental updates. In addition to bandwidth costs, if RPs process all retrieved data, including signature verification and object parsing, it will further increase computational load. These impacts are particularly pronounced in bandwidth- or resource-constrained environments, such as remote or mobile networks, edge deployments, or smaller Autonomous Systems.

5. Directions and Design Considerations

Type-based on-demand retrieval refers to the ability for Relying Parties (RPs) to selectively retrieve specific categories of RPKI objects, such as ROAs, ASPAs, or other types of signed objects, based on their operational requirements. This section discusses high-level directions and design considerations for enabling such selective retrieval within existing RPKI data synchronization mechanisms. This capability may be particularly beneficial in bandwidth constrained environments, specialized network deployments, or scenarios where rapid enablement of specific validation mechanisms is prioritized over comprehensive RPKI coverage.

5.1. Direction for Rsync

In Rsync servers [RSYNC], each RPKI object is mapped to a unique URI and can be retrieved individually. This model inherently allows RPs to fetch specific objects without requiring full repository synchronization. However, current RP implementations typically retrieve the complete set of published objects rather than leveraging this capability for selective retrieval.

From a design perspective, Rsync-based synchronization could support type-based on-demand retrieval by allowing RPs to identify and retrieve only the object types required for their intended validation functions.

5.2. Direction for RRDP

In RRDP servers, all RPKI objects are published in aggregated snapshot.xml and delta.xml files, including ROAs, associated certificates, and other signed objects. The current RRDP protocol is designed around full repository synchronization and does not provide mechanisms for selectively retrieving specific object types. As a result, RPs must download and process the complete dataset regardless of their actual requirements.

To better accommodate type-based on-demand retrieval, RRDP would need the ability to distinguish between different categories of RPKI objects during both initial synchronization and incremental updates, and to support selective transmission accordingly. This may require reconsideration of how snapshot and delta data are structured or organized, so that object type information can be efficiently exposed and used for selective retrieval.

5.3. Direction for Erik

The Erik Synchronization Protocol [I-D.spaghetti-sidrops-rpki-erik-protocol] is a proposed mechanism for efficient RPKI data replication. Erik employs Merkle trees to detect differences between local and remote datasets, allowing RPs to efficiently identify and retrieve objects that are missing or out of sync.

Similar to Rsync, Erik is primarily designed to support efficient synchronization of complete repository datasets. The current specification does not explicitly consider selective retrieval based on RPKI object types. However, Erik's use of Merkle tree based difference detection and object-level granularity provides a strong technical foundation for supporting more selective retrieval models. With these considerations, Erik could naturally support type-based on-demand retrieval and offer additional flexibility for deployments where only a subset of RPKI objects is required.

6. Security Considerations

This document is informational in nature and focuses on identifying challenges and discussing potential directions for improving RPKI data retrieval. It does not define new protocols, protocol extensions, or modifications to existing RPKI validation procedures or security mechanisms. Therefore, this document does not introduce new security considerations.

7. IANA Considerations

This document has no IANA requirements.

8. Appendix

The data referenced in Section 4 is based on experiments using Routinator in a production environment, measuring bandwidth consumption during Publication Point (PP) data synchronization. The measurements were conducted in two phases:

* *Bootstrapping Phase*: This phase tests the bandwidth requirements when an RP synchronizes complete PP data without any local cache, measuring the full snapshot download stage.

* *Update Phase*: This phase involves incremental updates performed every 10 minutes to measure the bandwidth consumption when RPs download delta files during routine synchronization operations.

9. References

9.1. 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>.

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>.

[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>.

[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>.

[I-D.ietf-sidrops-aspa-verification] Azimov, A., Bogomazov, E., Bush, R., Patel, K., Snijders, J., and K. Sriram, "BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-verification-24, 19 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-verification-24>.

[I-D.ietf-sidrops-moa-profile] Xie, C., Dong, G., Li, X., Huston, G., and D. Ma, "A Profile for Mapping Origin Authorizations (MOAs)", Work in Progress, Internet-Draft, draft-ietf-sidrops-moa-profile-03, 11 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-moa-profile-03>.

[I-D.spaghetti-sidrops-rpki-erik-protocol] Snijders, J., Bruijnzeels, T., Harrison, T., and W. Ohgai, "The Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI)", Work in

Progress, Internet-Draft, draft-spaghetti-sidrops-rpki-erik-protocol-04, 15 October 2025, <https://datatracker.ietf.org/doc/html/draft-spaghetti-sidrops-rpki-erik-protocol-04>.

[RSYNC] "The rsync web pages", n.d., <https://rsync.samba.org>.

Authors' Addresses

Jiayi Shi
Tsinghua University
Beijing
China
Email: sjy23@mails.tsinghua.edu.cn

Libin Liu
Zhongguancun Laboratory
Beijing
China
Email: liulb@zgclab.edu.cn

Lancheng Qin
Zhongguancun Laboratory
Beijing
China
Email: qinlc@mail.zgclab.edu.cn

Dan Li
Tsinghua University
Beijing
China
Email: tolidan@tsinghua.edu.cn