sidrops
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

D. Li
Tsinghua University
Y. Su
Zhongguancun Laboratory

7 July 2025

Decentralized RPKI Repository Architecture
draft-li-sidrops-drr-architecture-00

Abstract

The Resource Public Key Infrastructure (RPKI) plays a crucial role in securing inter-domain routing. However, the current RPKI Repository system suffers from fundamental limitations in reliability, scalability, and security. In particular, single-point failures at repository publication points (PPs), the growing number of bidirectional connections with relying parties (RPs), and the lack of mechanisms to mitigate misbehavior by certification authorities (CAs) pose significant risks to the integrity, authority, and resilience of the global RPKI ecosystem.

This document proposes the Decentralized RPKI Repository (dRR), a novel repository architecture that decouples RPKI object signing from data distribution. dRR introduces a distributed federation of certificate servers (CSs) and a layer of Monitors to achieve robust, scalable, and auditable RPKI data management. The architecture maintains full compatibility with the existing RPKI trust model rooted in the five RIR trust anchors, enabling incremental deployment without disrupting current relying parties (RPs) validation semantics. By doing so, dRR addresses longstanding repository challenges and enhances the overall trustworthiness and efficiency of RPKI-based routing security.

This Internet-Draft will expire on 8 January 2026.

Table of Contents

1. Introduction

The Resource Public Key Infrastructure has become a cornerstone for securing inter-domain routing, providing mechanisms to validate the association between Internet number resources (INR) and the entities authorized to originate them. While RPKI has proven effective in mitigating certain types of BGP hijacks, the current repository system exhibits critical architectural shortcomings.

In particular, the tight coupling between repository PPs and CAs results in three fundamental challenges. First, the system is prone to reliability issues: a failure or attack on any PP may prevent RPs from obtaining a complete or timely view of RPKI data. Second, the existing model lacks scalability, as the anticipated growth in the number of PPs and RPs—driven by broader deployment of ROAs and ROV—exacerbates synchronization load and connection overhead. Third, the architecture provides limited protections against misbehavior by CAs, leaving INR holders and RPs without effective means to detect or mitigate unilateral tampering of signed RPKI objects. These issues are discussed in detail in [rpki-repo-ps].

To address these concerns, this document introduces the Decentralized RPKI Repository (dRR), which extends and enhances the existing RPKI repository architecture. dRR is founded on the principle of decoupling RPKI object signing from data distribution. It replaces traditional PP-based storage with a distributed and decentralized federation of certificate servers (CSs), each operating as an equal peer that collaboratively hosts RPKI data. INR holders can proactively upload their RPKI objects to any trusted CSs within this federation, thereby retaining greater operational control.

Additionally, dRR incorporates Monitors as middleware components that aggregate updates from the CS federation and proactively distribute RPKI data to RPs. This design reduces the number of direct connections between RPs and repository nodes and improves both the scalability and timeliness of RPKI data distribution.

Through this architecture, dRR aims to enhance the robustness, scalability, and security of the global RPKI system while preserving compatibility with existing hierarchical trust anchors and RPKI data validation processes.

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 and Definitions

This document assumes familiarity with the concepts described in "A Profile for Resource Certificate Repository Structure" [RFC6481], "The RPKI Repository Delta Protocol (RRDP)" [RFC8182], and "An Infrastructure to Support Secure Internet Routing" [RFC6480].

This document defines the following terminology:

Decentralized RPKI Repository (dRR): The new RPKI repository architecture proposed in this document. dRR consists of multiple Certificate Servers (CSs) that collaboratively store and distribute RPKI data, along with Monitors that act as middleware between the CS federation and RPs.

Certificate Server (CS): A core component of the dRR system. A CS is a node responsible for hosting RPKI objects. All CS nodes within dRR collectively form a federation.

Monitor: Represents a key middleware component in dRR that sits between CS nodes and RPs. A Monitor retrieves RPKI data from CS nodes and proactively pushes updates to RPs.

Object Issuance Information (OII): Metadata associated with a newly issued RPKI object. An OII is signed by the CA that issued the object and includes the fingerprint of the object along with its intended storage locations within the CS nodes. The OII allows the CS federation to verify the authenticity of the newly issued RPKI object and to record it in the federation's global ledger.

Object Revocation Information (ORI): Metadata associated with an RPKI object that is to be revoked. An ORI is signed by the INR holder of the object, along with any affected subordinate INR holders. It includes the fingerprint of the object to be revoked and serves to prove that all impacted parties have authorized the revocation. The ORI enables the CS federation to verify the legitimacy of the revocation and to record it in the federation's global ledger.

Verifiable Update Information (VUI): Cryptographic metadata generated by a Monitor when proactively distributing updates to an RP. A VUI enables the RP to verify both the authenticity and the completeness of the received RPKI update data.

These terms are used throughout this document.

3. Design Principles and Objectives

The design of dRR is guided by two primary principles:

* Preserve the existing hierarchical trust model: dRR does not modify the hierarchical certificate issuance framework rooted in the five RIR trust anchors, which MUST remain the foundational basis of the global RPKI trust model. The roles and trust relationships of CAs and INR holders remain unchanged.

* Separate distribution from signing: While certificate issuance continues to be handled by existing RPKI authorities, the storage and operational management of RPKI objects are explicitly delegated to INR holders and a decentralized set of infrastructure components. INR holders SHALL maintain direct control over where and how their RPKI data is hosted. RPs MUST continue to perform cryptographic validation of all received data, preserving end-to-end trust guarantees.

Building on these principles, the dRR architecture serves exclusively as an alternative repository layer. It provides mechanisms for the upload, hosting, and storage of resource certificates (RCs), ROAs, and potentially newer signed objects such as autonomous system provider authorization (ASPA) and signed prefix list (SPLs). The design paradigm establishes a decentralized RPKI data management platform, effectively redefining the boundaries of responsibility between RPKI authorities and INR holders and mitigating the risks associated with unilateral operational control. It also supports both incremental and full data synchronization for RPs, thereby fulfilling all core repository functions without altering existing validation semantics.

The architecture of dRR is explicitly designed to meet the following objectives:

* Compatibility: dRR MUST NOT alter the hierarchical RPKI certificate issuance architecture. It SHALL remain fully compatible with the current RPKI Repository system and SHALL support incremental deployment alongside existing PPs without disrupting established validation or synchronization workflows.

* Reliability: dRR SHALL ensure high availability of RPKI data through distributed storage. A single RPKI object MAY be redundantly stored across multiple physically isolated repository nodes. Consequently, the integrity and accessibility of RPKI data as consumed by RPs SHALL NOT be compromised even if individual nodes experience failures.

* Scalability: The dRR architecture SHALL prevent uncontrolled growth in the number of repository nodes required to host data. It SHOULD enable efficient synchronization mechanisms that ensure RPs receive timely updates of RPKI data, regardless of the scale of deployment.

* Security and Auditability: By decoupling RPKI object signing from data storage and empowering INR holders to control their own RPKI objects, dRR SHALL actively mitigate risks stemming from malicious or negligent behavior by RPKI authorities. The system SHALL also maintain a complete and consistent historical record of RPKI data and operations, thereby supporting robust auditing capabilities for verifying the correctness and continuity of repository activities.

By adhering to these principles and objectives, dRR aims to enhance the robustness, transparency, scalability, and security of the RPKI repository layer while ensuring full interoperability with the existing RPKI infrastructure.

4. Architecture Overview

4.1. Key Components

The dRR architecture replaces the conventional RPKI Repository system with two principal components: a *distributed Certificate Server federation* (CS federation) and a layer of *middleware Monitors*.

4.1.1. Certificate Server Federation

The CS federation consists of multiple independent certificate server nodes that collectively host and store RPKI data. Each CS node participates equally in this federation and operates under a shared consensus protocol to maintain a consistent view of hosted RPKI objects. INR holders MAY upload their RC certificates, ROAs, or other signed objects issued by RPKI authorities to any subset of trusted CS nodes they choose.

This decentralized model decouples data storage and management responsibilities from certificate signing authorities. As a result:

* INR holders maintain operational control and management over their RPKI data.

* RPKI objects MAY be simultaneously stored across multiple CS nodes, improving redundancy and availability.

* No single CA can unilaterally manage or tamper with the stored data.

4.1.2. Middleware Monitors

Monitors act as intermediaries between the CS federation and RPs. Each Monitor SHALL:

* Receive OII and ORI from the CS federation.

* Retrieve newly published from the specified CS nodes based on OII.

* Generate VUI according OII and ORI and proactively push both VUI and the associated newly added RPKI objects to the RPs it serves.

* Maintain a full dRR-protected RPKI data snapshot.

This two-tier proactive data distribution architecture contrasts with the traditional model in which each RP MUST periodically poll multiple PPs to obtain updates. By aggregating and pushing updates, Monitors reduce synchronization delays and alleviate load on both the repository infrastructure and RPs.

4.1.3. Key Differences from the Current System

Compared to the existing RPKI repository architecture, the key differences introduced by dRR include:

* Flexible Data Storage: Under dRR, INR holders actively upload their RPKI objects to selected CS nodes. They SHALL have the freedom to choose one or more trusted nodes for storing their data. By decoupling RPKI object signing authority from data storage and management responsibilities, dRR transfers operational and management control of RPKI data directly to the legitimate INR holders. By contrast, the current model requires that data signed by a CA MUST be stored exclusively at the PP operated or designated by that CA.

* Federated Transparency:The CS nodes in dRR collectively operate under a consensus protocol. Specifically, newly uploaded objects and object revocation statements SHALL be publicly disseminated across the CS federation through OII and ORI. Only after

consensus is reached among the nodes does a new object become valid or an old certificate become invalid. This process provides a transparent and auditable view of all data storage and operational changes, which in turn enhances the overall security of the system by allowing stakeholders to independently verify repository behaviors. By comparison, existing PPs operate in isolation, each maintaining and managing its own data view without inter-node validation or transparency.

* Two-tier Push Model: dRR introduces a layered proactive distribution mechanism. CS nodes SHALL push OII and ORI and cryptographic proofs to a designated set of Mnitors. Each Monitor SHALL then retrieve newly added RPKI data from the specific CS nodes according to the OII, and SHALL generate VUI and proactively distribute both the VUI and the newly added RPKI objects to its designated set of RPs. In dRR, each CS typically serves a defined group of Monitors, and each Monitor, in turn, serves a defined group of RPs. For today' s RPKI Repository, the full set of PPs and RPs must establish a large number of bidirectional connections, which increases the load and pressure of data synchronization.

The distributed CS federation is the core component enabling decentralization in dRR. Each server node participates in a consensus protocol to build a cross-domain trust network, thereby forming a data hosting platform that operates independently of any single RPKI authority. All server nodes function as equal participants, collectively providing certificate hosting services to INR holders within the RPKI system. The middleware layer of Monitors acts as the central hub for data propagation. It receives consensus-validated RPKI data and associated operations from CS federation and, based on a proactive push model, delivers new RPKI objects to RPs.

4.2. Federation Construction

The CS federation is comprised of multiple CS nodes. The following principles govern the operation and structure of these nodes:

a) Each CS operates independently of any RPKI authority. All CS nodes SHALL be equal peers and SHALL collectively form a federation to provide secure and reliable hosting services for INR holders.

b) Any entity seeking to join the CS federation MUST undergo security validation performed by the existing CS federation members.

c) All operations within the federation— including the registration of new CS nodes, registration of INR holders, publication of RPKI objects, and object revocation— MUST achieve consensus among all participating CS federation members.

The initialization of the CS federation, along with the registration mechanisms for CS nodes and INR holders, are described below.

* Initialization of the CS Federation: The initial CS federation SHALL be established jointly by the five RIRs: RIPE NCC, APNIC, AFRINIC, LACNIC, and ARIN. During initialization, each RIR SHALL register its CS node within the federation. The registration information for each CS node MUST include: (1) A unique identifier for the CS node; (2) The node's unique public key; (3) A digital signature generated using the private key corresponding to that public key.

* Registration of CS Nodes: Entities that meet the eligibility requirements MAY apply to operate RPKI object hosting servers and join the CS federation. This process proceeds as follows: (1) The applying entity submits its registration information to one of the five RIRs; (2) Upon successful completion of identity, security, and reliability verification (for example, evaluation of the applicant's reputation or its capability to operate on robust platforms such as CDNs), the RIR SHALL publish the entity's registration information to the CS federation; (3) The existing federation members SHALL validate the identity of the new CS node and verify the provided signatures; (4) If a majority of federation members approve, consensus SHALL be reached and the entity's registration information SHALL be permanently recorded in the federation's global ledger. The entity SHALL then formally become a member of the CS federation and MAY provide hosting services to INR holders. If consensus is not achieved, the entity SHALL NOT be admitted into the CS federation.

* Registration of INR Holders. Any INR holder deploying dRR MUST first complete identity registration within the CS federation. The process is as follows: (1) The INR holder submits its registration information to one trusted CS node; (2) The receiving CS node SHALL publish the registration information to the federation, triggering a validation process by the federation members. (3) If a majority of members validate the registration successfully, the INR holder's information SHALL be recorded in the federation's global ledger. If consensus is not achieved, registration SHALL fail.

It is important to note that any RPKI authority inherently holds an RC certificate issued by its parent authority (or, in the case of RIRs, a self-issued RC root certificate). Therefore, for the purposes of dRR, an RPKI authority is also treated as an INR holder and MUST undergo the same INR holder registration process within the CS federation.

4.3. Federation Operation

Within dRR, INR holders actively manage the distribution of their RPKI objects by uploading them to one or more trusted CS nodes. The CS federation ensures the integrity and transparency of these operations by requiring that both the publication of new objects and the revocation of existing objects be represented as OII and ORI records. These records MUST be disseminated across the federation and permanently recorded in the federation's global ledger. This immutable, time-ordered ledger provides a comprehensive audit trail of all RPKI object operations.

4.3.1. Publication of New Objects

To publish a new RPKI object, the INR holder submits the newly issued object—signed by its parent RPKI authority—to multiple trusted CS nodes for secure storage. The INR holder also submits the corresponding OII to a designated CS node, which SHALL publish this OII to the CS federation.

* If consensus on the OII is achieved among federation members, the OII SHALL be permanently recorded in the federation's global ledger. The object is then officially protected by the federation and becomes available for retrieval and synchronization by RPs.

* If consensus is not reached, the publication process SHALL fail, and the object SHALL NOT be incorporated into the federation's authoritative dataset.

An OII, as defined by this document, MUST include at a minimum: (1) The identifiers of the issuer and recipient as recorded in the federation's registration system; (2) The parent certificate of the newly issued object; (3) A unique hash fingerprint of the object; (4) A list of CS nodes that store the object; (5) A signature generated by the RPKI authority.

4.3.2. Revocation of Existing Objects

When an RPKI authority intends to revoke an RPKI object previously issued to a subordinate INR holder, it MUST first request an ORI from that INR holder. The RPKI authority then submits the ORI to one or more trusted CS nodes, which SHALL propagate the ORI to the remaining federation members.

* If consensus on the ORI is achieved among federation members, the ORI SHALL be written to the global ledger. The corresponding object SHALL then be revoked and removed from the storage of all participating CS nodes.

* If consensus is not achieved, the revocation SHALL fail, and the object SHALL remain valid in the federation.

An ORI, as defined by this document, MUST include at a minimum: (1) The unique hash fingerprint of the object to be revoked; (2) An authorization signature generated by the INR holders; (3) (Optional) Additional endorsements from affected subordinate INR holders when applicable.

By ensuring that all publication and revocation operations undergo federation-wide consensus and are permanently logged, dRR provides INR holders and RPs with a verifiable and tamper-evident history of RPKI object management. This decentralized model significantly strengthens accountability and mitigates the risks associated with unilateral control by any single RPKI authority.

4.4. Monitor Operation

A Monitor in the dRR architecture acts as a middleware entity responsible for bridging the CS federation and RPs. Its primary function is to facilitate the efficient dissemination of updated RPKI data to RPs.

Whenever the CS federation reaches consensus on a new set of OII and ORI, the participating CS nodes SHALL proactively push these confirmed records to their connected Monitors. Each Monitor SHALL then:

* Retrieve the newly added or updated RPKI objects from the relevant CS nodes based on the received OIIs.

* Use the OIIs and ORIs to generate VUI, which cryptographically binds the latest updates to a verifiable proof.

* Proactively push the VUI, along with the newly retrieved RPKI objects, to the set of RPs it serves.

* Maintain a real-time full dRR-protected RPKI data snapshot to serve newly added RPs.

This process ensures that RPs receive not only the updated data but also the cryptographic means to verify both the authenticity and the completeness of each update.

5. dRR Operation Procedures

The primary functions of an RPKI Repository are to store, manage, and synchronize certificates and signed objects, including the publication and revocation of such data. The following subsections describe how these processes are handled in the dRR system.

5.1. Publication and Synchronization

Figure 1 illustrates the end-to-end process by which a new RPKI certificate or signed object is published and subsequently synchronized to RPs within the dRR system.

The sequence proceeds as follows:

Step 1. Request Object: The INR holder submits a request to the relevant RPKI authority to obtain resources along with the corresponding RC or signed objects (such as ROAs, ASPAs, or SPLs).

Step 2. Issue Object and OII: The RPKI authority issues the requested RC or signed object (collectively referred to as objects) and returns these, along with the associated OII, to the INR holder.

Step 3. Upload the Object and OII: The INR holder uploads the object to multiple trusted CS nodes and submits the corresponding OII to one of the CS nodes hosting the object.

Step 4. Object Hosting: The selected CS nodes store the uploaded RPKI object on behalf of the INR holder.

Step 5. Consensus in the Federation: One of the hosting CS nodes submits the OII to the CS federation. Federation members execute a consensus protocol to collectively validate the OII. If consensus is achieved, the OII is permanently recorded in the federation’s global ledger, completing the object publication.

Step 6. Push OII to Monitors: The CS nodes push the consensus-approved OII to their connected Monitors.

Step 7. Retrieving the Object by Monitors: Each Monitor retrieves the newly published object from the relevant CS nodes according to the details in the OII.

Step 8. Pushing Data to RPs: The Monitor generates a VUI record based on the new OII and proactively distributes bboth the VUI and the newly retrieved RPKI data to its designated RPs. This enables RPs to verify the authenticity and completeness of the received updates.

This coordinated process ensures that newly issued RPKI object is reliably stored, transparently validated through federation consensus, and efficiently delivered to RPs with cryptographic assurances of integrity and completeness.

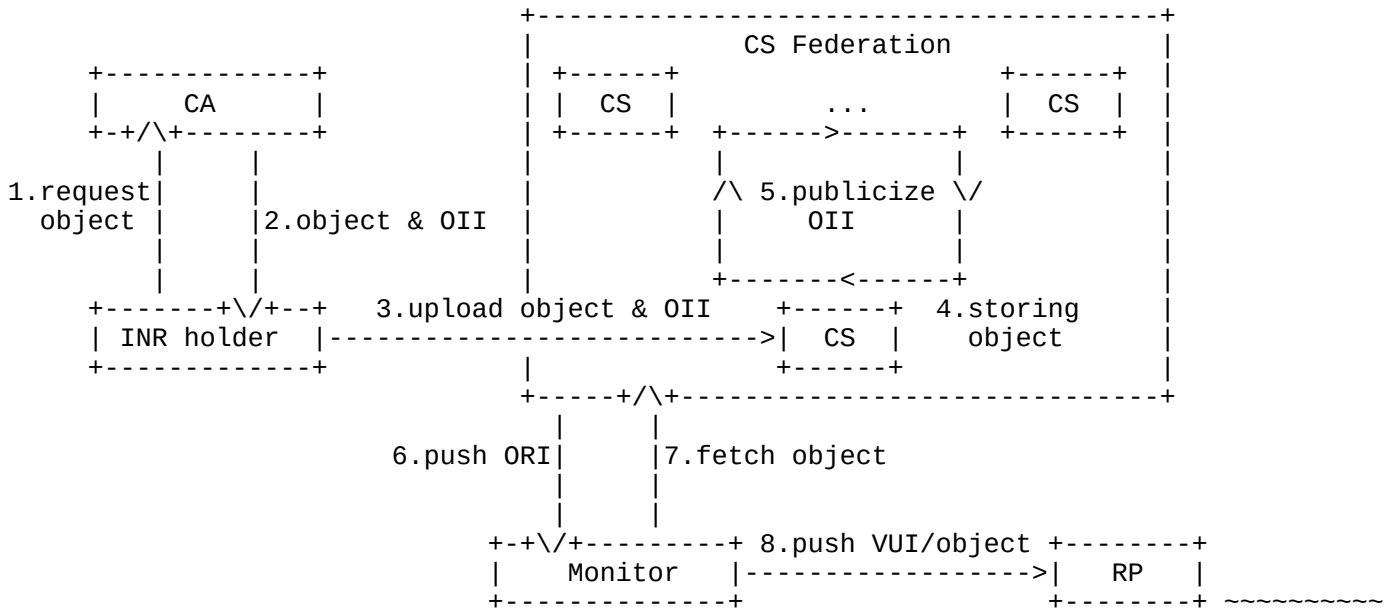

Figure 1: RPKI object publication and synchronization procedure in dRR.

5.2. Revocation and Synchronization

Figure 2 illustrates the process by which an existing RPKI object is revoked within the dRR system and how this revocation information is synchronized to RPs:

Step 1. Revocation request: The RPKI authority initiates a revocation by requesting the INR holder to revoke a specific RPKI object.

Step 2. Issue ORI: Upon agreeing to the revocation, the INR holder generates and signs an ORI, which serves as verifiable proof that the INR holder (and any affected subordinate holders) consent to the revocation.

Step 3. Upload ORI: The RPKI authority submits the signed ORI to one of the trusted CS nodes.

Step 4. Federation consensus: The receiving CS node publishes the ORI to the CS federation. Federation members execute a consensus protocol to validate and accept the ORI. If consensus is reached, the ORI is recorded in the federation’s global ledger, the object is officially revoked, and it is removed from all relevant CS storage. If consensus fails, the revocation does not proceed.

Step 5. Push to Monitors: CS nodes proactively push the consensus-approved ORI to their connected Monitors.

Step 6. Push to RPs: Monitors generate a VUI based on the new ORI and proactively push both the revocation data and the VUI to the RPs they serve. This enables RPs to verify the authenticity and completeness of the revocation update.

This process ensures that object revocations in dRR are cryptographically proven, transparently recorded across the federation, and rapidly synchronized to relying parties through proactive distribution.

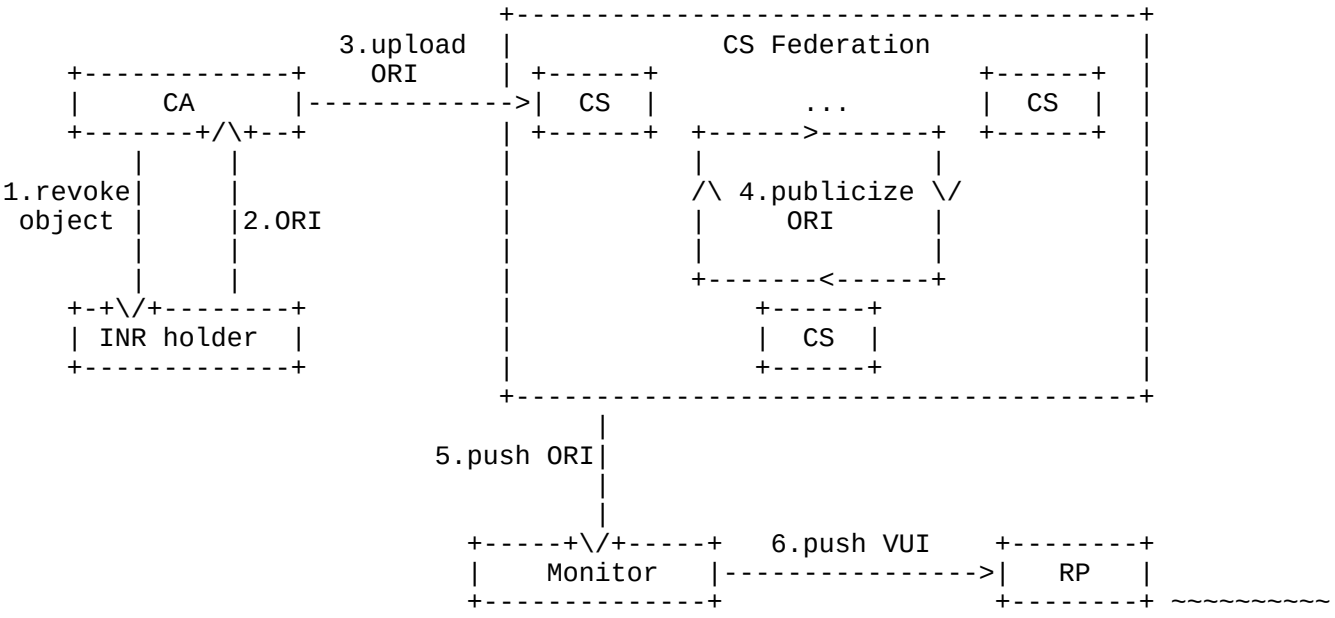

Figure 2: RPKI object revocation and synchronization procedure in dRR.

It should be noted that in practice, the CS federation may concurrently process multiple OIIs and ORIs. Similarly, a single CS node MAY aggregate and push multiple OIIs and ORIs to connected Monitors within the same operation. As a result, the VUI generated by a Monitor and delivered to RPs MAY encapsulate multiple new object publications and multiple object revocations in a single update.

6. dRR Participants and Roles

The dRR system is explicitly designed to support a diverse set of operational participants to ensure technical neutrality, operational robustness, and broad stakeholder representation.

Operators of Certificate Servers. Within the CS federation, nodes MAY be operated not only by the five RIRs—RIPE NCC, APNIC, AFRINIC, LACNIC, and ARIN—but also by other qualified entities. These MAY include large Internet Service Providers (ISPs), national Internet registries, Content Delivery Network (CDN) service providers, or other infrastructure organizations capable of reliably hosting RPKI data and serving RPs. By enabling a mix of technical and geographic participation, the CS federation mitigates risks associated with centralized control and improves the resilience of the overall repository system.

Operators of Monitors. In the dRR architecture, the Monitor role is intended to serve multiple RPs that typically exist within the same trust domain as the Monitor itself. For example, these RPs MAY include multiple Autonomous Systems (ASes) under the operational control of the same ISP, or networks within the jurisdiction of a single governmental or national organization. This trust relationship allows Monitors to efficiently aggregate updates from the CS federation and securely distribute RPKI data to affiliated RPs, thereby reducing synchronization overhead and enhancing overall system efficiency.

By explicitly supporting a multi-party operational model for both CS federation nodes and Monitors, the dRR architecture fosters participation from a range of organizational types and jurisdictions. This approach helps achieve a balanced ecosystem that reflects diverse operational interests while reducing systemic dependencies on any single class of stakeholder. The inclusion of varied participants is fundamental to ensuring the long-term neutrality, transparency, and accountability of the RPKI data distribution infrastructure. Whether and how CS nodes provide services to INR holders on a compensated basis is out of scope for this document.

7. Benefits and Improvements

By introducing a federation of CS nodes and a layer of Monitors, dRR May address several longstanding challenges of the current RPKI Repository system and achieves tangible operational benefits:

* *Enhancing Reliability and Mitigating P1:* In dRR, each RPKI object can be simultaneously hosted across multiple CS nodes. This means that the failure or compromise of any single node does not jeopardize the availability or integrity of RPKI data as retrieved by RPs. Additionally, the federated structure inherently filters for participants with the capacity to provide robust, stable hosting, further improving the system’s resilience against localized outages.

* *Improving Scalability and Mitigating P2:* dRR enhances scalability through two key mechanisms. First, the federation's controlled admission mechanism ensures that only vetted entities can operate CS nodes, effectively limiting the overall number of repository nodes while still supporting a globally distributed architecture. Second, dRR's two-tier data distribution model means that CS nodes only need to establish connections with their designated Monitors, and RPs in turn only maintain connections with their respective Monitors. This structured push model significantly reduces the volume of direct bidirectional connections between RPs and repository nodes, alleviating synchronization pressure and supporting large-scale deployment without compromising timeliness or efficiency.

* *Strengthening Security & Accountability and Mitigating P3:* By decoupling RPKI object signing from data storage, dRR shifts operational control of published RPKI objects directly to INR holders. This substantially reduces the risk of unilateral deletions or modifications by upstream CAs and strengthens accountability. The transparent consensus mechanism within the federation also provides an auditable record of all publication and revocation operations.

8. Deployment Considerations

The incremental deployment of dRR is designed to align with the existing hierarchical structure of the RPKI system, following a top-down approach. Specifically, an RPKI object MAY be protected by dRR only if its parent RC is already under dRR protection. This section outlines deployment behaviors in partial adoption scenarios.

(1) Certificate Management Model.

When an RPKI authority adopts dRR, its subordinate INR holders MAY independently decide whether to deploy dRR. As a result:

* An RPKI authority SHALL continue operating its traditional repository PP for as long as any of its subordinate INR holders have not deployed dRR, to ensure continued hosting of their RPKI objects.

* Only when all subordinate INR holders have fully transitioned to dRR MAY the RPKI authority discontinue operation of its PP.

* For subordinate INR holders that have not deployed dRR, their RPKI data management remains unchanged, and their objects SHALL continue to be stored in the authority's PP.

* For INR holders that have deployed dRR, their RPKI data SHALL NOT be stored in the parent RPKI authority's PP, and the corresponding PP manifest SHALL NOT include RCs or signed objects protected by dRR.

(2)Incremental Data Synchronization.

During partial deployment phases, RPs SHOULD synchronize data from both dRR and the existing RPKI repository system to maintain a complete view of the RPKI state. This involves:

* Updates via dRR: The RP applies updates proactively pushed by Monitors, using the received VUI and associated data. The RP SHALL add newly published objects indicated by OIIs and remove revoked objects as specified by ORIs.

* Updates via Traditional Repositories: The RP SHALL continue to periodically synchronize RPKI data not protected by dRR by querying all relevant repository PPs using protocols such as Rsync or RRDP.

(3)Full Data Synchronization.

Similarly, during partial deployment, RPs MUST acquire full RPKI data snapshot from both dRR Monitor and the current RPKI repository to ensure full synchronization of all RPKI objects.

dRR explicitly supports such phased deployment models. During incremental adoption, dRR continues to provide full security assurances for certificates under its protection, ensuring a consistent trust model throughout the transition process.

9. Security Considerations

TBD

10. IANA Considerations

This document has no IANA requirements.

11. References

11.1. Normative References

[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/rfc/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/rfc/rfc8182>.

[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/rfc/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/rfc/rfc6481>.

[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/rfc/rfc2119>.

11.2. Informative References

[rpki-repo-ps] "RPKI Repository Problem Statement", Internet-Draft draft-li-sidrops-rpki-repository-problem-statement-01 , 2025, <https://datatracker.ietf.org/doc/draft-li-sidrops-rpki-repository-problem-statement/>.

Authors' Addresses

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

Yingying Su
Zhongguancun Laboratory
Beijing
China
Email: suyy@mail.zgclab.edu.cn