SIDR Operations
Internet-Draft
Intended status: Best Current Practice
Expires: 31 March 2026

T. C. Sweetser
IEISI

27 September 2025

Operational Guidelines for RPKI Delegated Certification Authorities
draft-sweetser-bcp-rpki-ca-01

Abstract

This document provides operational guidelines for Resource Public Key Infrastructure (RPKI) delegated Certification Authorities (CAs) and registry operators managing such delegations. It addresses common operational issues including CA availability problems, publication quality issues, and lifecycle management. The guidelines aim to improve the overall health and efficiency of the RPKI ecosystem by establishing best practices for CA operations and delegation management.

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 31 March 2026.

Copyright Notice

Table of Contents

1. Introduction

The Resource Public Key Infrastructure (RPKI) [RFC6480] provides a framework for securing Internet routing through cryptographic attestation of IP address and AS number allocations. Regional Internet Registries (RIRs) and National Internet Registries (NIRs) may delegate RPKI certification authority (CA) operations to resource holders, allowing them to manage their own certificate issuance and publication.

While RPKI delegation provides operational flexibility and autonomy for resource holders, it also introduces potential failure modes that can negatively impact the broader RPKI ecosystem. Poorly operated delegated CAs can cause significant resource waste for RPKI validators, degrade overall system performance, and undermine confidence in RPKI deployment.

This document establishes operational guidelines for both delegated CA operators and registry operators managing such delegations. The guidelines address common operational issues including CA availability problems, publication inconsistencies, and lifecycle management challenges.

The recommendations in this document are based on operational experience from RPKI deployments worldwide and analysis of problematic CA behaviors observed in production systems.

2. Recommended Reading

[RFC6481] and [RFC6489] will inform readers of requirements on repository content structure, directory structure, naming of directories and managaing key rollovers.

[RFC8182] provides details of efficient alternative to rsync[RFC5781] and key operational efficiencies like caching and CDN deployment.

[CURE-NDSS24] provides statistical data and security analysis of RPKI performance. With code at https://github.com/rp-cure/rp-cure (https://github.com/rp-cure/rp-cure).

3. 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.

This document uses the following terms:

Delegated CA: A certification authority that has been delegated RPKI certificate issuance authority by a registry operator, and is operated by the resource holder rather than the registry [RFC3647][RFC5280].

Registry Operator: An organization (typically an RIR or NIR) that delegates RPKI CA authority to resource holders and maintains oversight of such delegations [RFC7020][RFC6484].

Publication Point: A repository location where RPKI objects (certificates, manifests, CRLs, ROAs, etc.) are published and made available to relying parties [RFC8181][I-D.ietf-sidrops-publication-server-bcp].

Flapping CA: A delegated CA that exhibits rapid or frequent changes in operational state, causing instability for relying party validators [RFC8211].

Dead CA: A delegated CA that is persistently non-functional or unreachable for extended periods [DEAD-CA].

Manifest: An RPKI signed object that lists the contents of a publication repository and provides integrity protection [RFC9286].

Relying Party (RP): An entity that uses RPKI data to make routing decisions, typically through RPKI validator software [RFC6480].

4. Problem Statement

Current RPKI delegation practices allow for several problematic operational scenarios that negatively impact the RPKI ecosystem. Analysis of production RPKI systems has identified recurring patterns of CA misbehavior that waste validator resources and degrade system performance.

4.1. CA Availability Issues

Delegated CAs may experience various availability problems:

Dead CAs: Some delegated CAs become completely offline for extended periods (weeks to months), yet remain referenced in the RPKI hierarchy. RPKI validators continue attempting to fetch data from these CAs, resulting in hundreds of thousands of failed synchronization attempts over time [DEAD-CA].

Flapping CAs: Certain delegated CAs exhibit intermittent availability patterns, rapidly cycling between online and offline states. This "flapping" behavior causes cache thrashing in RPKI validators and creates uncertainty about the current state of published objects.

Slow CAs: Some delegated CAs respond to requests but with excessive latency, causing validator timeouts and retry storms. These CAs may appear functional but create significant operational burden.

Partial Failures: Delegated CAs may have some publication endpoints functioning while others fail, leading to inconsistent data availability and validator confusion.

4.2. Publication Quality Issues

Beyond availability, delegated CAs may exhibit poor publication practices:

Stale Manifests: CAs that rarely update their manifests cause validators to repeatedly attempt synchronization of unchanged data, wasting network bandwidth and processing resources.

Inconsistent Publication: CAs may publish incomplete or inconsistent object sets, with manifests referencing objects that are not available or publishing objects not listed in manifests.

Clock Skew: CAs with incorrect system time may publish objects with validity periods that cause premature expiration or rejection by validators.

Malformed Objects: CAs may publish syntactically invalid or incorrectly signed objects, causing validator errors and potentially undermining security.

4.3. Operational Anti-patterns

Several operational practices by delegated CAs create ecosystem-wide problems:

Resource Churn: Frequent unnecessary certificate reissuance creates processing overhead for validators and may indicate poor operational practices.

Publication Storms: Bulk updates that overwhelm validators or publication infrastructure, often due to poor change management.

Inconsistent Policies: Conflicting or rapidly changing ROA/ASPA publications that create uncertainty for relying parties.

5. Operational Requirements for Delegated CAs

5.1. Availability and Reliability Standards

Delegated CA operators MUST implement robust infrastructure to ensure reliable service delivery

5.1.1. Availability Requirements

* CA operators MUST maintain greater than 99.5% availability for all publication points over any 30-day period.
* CA operators MUST implement redundant publication infrastructure with automatic failover capabilities.
* CA operators MUST provide multiple publication endpoints with geographic diversity to minimize single points of failure.
* CA operators MUST implement comprehensive monitoring with automated alerting for all critical infrastructure components.

5.1.2. Infrastructure Requirements

* CA operators SHOULD implement load balancing across publication points to distribute validator traffic.
* CA operators SHOULD use Content Distribution Networks (CDNs) or anycast addressing for global distribution of RPKI objects.
* CA operators SHOULD maintain hot standby systems capable of taking over operations within one hour of primary system failure.
* CA operators SHOULD implement DDoS protection and traffic shaping to handle validator traffic spikes.

5.2. Publication Discipline

Delegated CA operators MUST follow disciplined publication practices to minimize validator burden:

5.2.1. Manifest Management

* CA operators MUST publish new manifests at regular intervals, RECOMMENDED to be between 4-8 hours for active CAs.
* CA operators MUST ensure manifest validity periods provide sufficient operational windows, RECOMMENDED minimum of 24 hours.
* CA operators MUST NOT allow manifest gaps (periods without valid manifests) longer than 24 hours except during planned maintenance.
* CA operators SHOULD implement manifest publication schedules that are predictable and avoid unnecessary updates.

5.2.2. Publication Practices

* CA operators MUST implement atomic publication updates to prevent temporary inconsistencies between manifests and published objects.
* CA operators MUST maintain consistent object naming and URI structure throughout the CA’s operational lifetime.
* CA operators SHOULD NOT publish manifest updates more frequently than every 30 minutes without operational justification.
* CA operators SHOULD batch certificate operations to minimize publication frequency while maintaining security.

5.3. Data Integrity and Consistency

Delegated CA operators MUST ensure the integrity and consistency of all published RPKI objects.

5.3.1. Validation Requirements

* CA operators MUST validate all RPKI objects before publication, including syntax, cryptographic signatures, and logical consistency.

* CA operators MUST implement automated testing of published repositories using standard RPKI validator software.
* CA operators MUST maintain audit logs of all certificate operations for a minimum of two years.
* CA operators MUST ensure accurate time synchronization (NTP) across all infrastructure components with stratum 2 or better accuracy.

5.3.2. Operational Consistency

* CA operators SHOULD implement configuration management systems to ensure consistent deployments across redundant infrastructure.
* CA operators SHOULD maintain staging environments for testing changes before production deployment.
* CA operators SHOULD implement automated backup and recovery procedures with regular testing of restore capabilities.
* CA operators SHOULD coordinate planned maintenance activities with registry operators and the broader community when possible.

6. Addressing Problematic CA Behaviors

6.1. Flapping CA Detection and Mitigation

Flapping behavior represents one of the most disruptive operational problems for RPKI validators. CA operators and registry operators MUST implement measures to detect and prevent flapping behaviors.

6.1.1. Flapping Patterns

Common flapping patterns include:

* Rapid cycling between online and offline states with periods shorter than typical validator refresh intervals (< 1 hour)
* Publication inconsistencies where repositories alternate between different content states
* Intermittent network connectivity causing sporadic timeouts
* Infrastructure instability causing frequent service interruptions

CA Operator Anti-Flapping Measures:

* CA operators MUST implement circuit breaker patterns for failing dependencies to prevent cascading failures.
* CA operators MUST use comprehensive health checks before bringing systems online after maintenance or failures.
* CA operators MUST implement graceful degradation procedures rather than hard failures when possible.
* CA operators SHOULD implement exponential backoff for automated recovery processes to prevent rapid cycling.

Registry Operator Flapping Detection:

* Registry operators SHOULD implement automated flapping detection algorithms monitoring CA state changes over time.
* Registry operators SHOULD apply progressive operational restrictions for CAs exhibiting flapping behavior.
* Registry operators SHOULD require stabilization periods before allowing re-delegation after revocation due to flapping.

6.2. Dead CA Management

Persistently non-functional CAs create significant waste in the RPKI ecosystem and MUST be managed through clear lifecycle policies.

6.2.1. Detection Criteria

A CA SHOULD be considered persistently non-functional if: - Valid manifest and CRL cannot be retrieved for more than 60 days - Multiple publication points are consistently unreachable - Published objects consistently fail validation - No response to operational communications for extended periods

6.2.2. Registry Operator Actions

* Registry operators MUST implement automated monitoring to detect persistently non-functional CAs.
* Registry operators MUST make reasonable efforts to contact CA operators before taking revocation actions.
* Registry operators SHOULD revoke delegations for CAs that remain non-functional for more than 90 days after initial contact attempts.
* Registry operators MUST provide clear communication about revocation policies and procedures to CA operators.

6.3. Performance Issues

CA operators MUST ensure their infrastructure provides adequate performance for the global validator ecosystem.

6.3.1. Performance Standards

* Publication points MUST respond to HTTP requests within 10 seconds under normal conditions.
* Publication points SHOULD support concurrent connections from multiple validators without degradation.
* Object retrieval SHOULD complete within 30 seconds for typical repository sizes.

* CA operators SHOULD implement caching mechanisms to reduce server load and improve response times.

6.3.2. Monitoring and Remediation

* CA operators MUST monitor response times and error rates continuously.

* CA operators SHOULD implement automated alerting for performance degradation.

* CA operators MUST have procedures for rapid response to performance issues.

7. Monitoring and Alerting Framework

7.1. CA Operator Self-Monitoring

Delegated CA operators MUST implement comprehensive monitoring of their infrastructure and operations.

7.1.1. Required Monitoring Metrics

* Publication point availability and response time from multiple geographic locations
* Manifest freshness and validity period remaining
* Object count and repository size trends over time
* Certificate expiration schedules and renewal status
* Error rates and failure patterns across all services
* Infrastructure resource utilization (CPU, memory, disk, network)

7.1.2. Alerting Requirements

CA operators MUST implement alerting for:

* Publication point failures (immediate notification)
* Manifest aging beyond 4-hour threshold
* Certificate expiration warnings at 30, 7, and 1 day intervals
* Anomalous traffic patterns or potential attacks
* Infrastructure failures or resource exhaustion

Monitoring Infrastructure:

* CA operators SHOULD implement redundant monitoring systems to prevent single points of failure.
* CA operators SHOULD use external monitoring services to detect outages from validator perspectives.

* CA operators SHOULD maintain historical monitoring data for trend analysis and capacity planning.

7.2. Registry Operator Monitoring

Registry operators MUST implement monitoring systems to oversee all delegated CAs under their authority.

7.2.1. Monitoring Scope

Registry operators MUST monitor:

* Availability and response times for all delegated CA publication point
* Manifest publication frequency and consistency
* Object validation status and error patterns
* Validator impact metrics (fetch attempts, success rates, timing)
* Compliance with operational guidelines and policies

7.2.2. Automated Detection

Registry operators SHOULD implement automated detection of:

* Dead CAs (extended periods of non-functionality)
* Flapping CAs (rapid state changes)
* Performance degradation trends
* Policy violations or operational anomalies

Reporting and Communication:

* Registry operators SHOULD provide operational dashboards for CA operators to monitor their own performance.
* Registry operators SHOULD publish aggregate ecosystem health metrics.
* Registry operators MUST maintain communication channels for urgent operational issues.

7.3. Validator-side Monitoring

RPKI validator operators are encouraged to implement monitoring that can help identify problematic CAs.

7.3.1. Recommended Metrics

* Per-CA fetch success rates and timing
* Manifest validation patterns and frequency
* Repository size and change detection
* Error categorization and frequency analysis

* Resource consumption per CA

7.3.2. Community Reporting

* Validator operators SHOULD report persistent CA issues to registry operators.
* Validator operators MAY implement automated reporting of severe CA performance issues.
* Registry operators SHOULD provide mechanisms for community feedback on CA operations.

8. CA Lifecycle Management

8.1. Pre-Delegation Requirements

Registry operators MUST establish clear requirements for CA delegation to ensure operational readiness.

8.1.1. Technical Requirements

Before approving delegation requests, registry operators SHOULD verify:

* Stable infrastructure demonstrated for minimum 30-day period
* Implementation of monitoring and alerting systems
* Documented operational procedures and emergency contacts
* Backup and recovery capabilities with tested procedures
* Adequate technical expertise and operational resources

Documentation Requirements:

* CA operators MUST provide detailed operational procedures
* CA operators MUST maintain current emergency contact information
* CA operators MUST document backup and disaster recovery procedures
* CA operators SHOULD provide capacity planning and scaling documentation

Validation Process:

* Registry operators SHOULD implement technical validation of CA infrastructure before delegation.
* Registry operators MAY require probationary periods for new delegations with enhanced monitoring.
* Registry operators SHOULD provide operational guidance and training resources for new CA operators.

8.2. Ongoing Operational Standards

Delegated CA operators MUST maintain high operational standards throughout the lifecycle of their delegation.

8.2.1. Operational Disciplines:

* CA operators MUST implement regular infrastructure maintenance windows with advance notification to registry operators.
* CA operators MUST implement proactive certificate renewal procedures, initiating renewal at least 30 days before expiration.
* CA operators MUST follow change management procedures for all infrastructure updates.
* CA operators MUST maintain incident response plans and communication procedures.

8.2.2. Reporting Requirements

* CA operators SHOULD provide regular operational reports to registry operators.
* CA operators MUST report significant incidents or outages to registry operators within 24 hours.
* CA operators SHOULD participate in operational reviews and community engagement activities.

8.2.3. Continuous Improvement

* CA operators SHOULD regularly review and update operational procedures.
* CA operators SHOULD implement lessons learned from incidents and outages.
* CA operators SHOULD stay current with RPKI operational best practices and security recommendations.

8.3. Graceful Shutdown Procedures

CA operators planning to discontinue operations MUST follow established procedures to minimize ecosystem disruption.

8.3.1. Planning Requirements

For planned CA retirement:

* CA operators MUST provide minimum 90-day advance notice to registry operators and affected communities.
* CA operators MUST coordinate with registry operators on migration procedures for dependent resources.

* CA operators SHOULD gradually reduce certificate validity periods to facilitate migration.
* CA operators MUST maintain service levels during the shutdown transition period.

8.3.2. Migration Support

* Registry operators SHOULD provide migration assistance and alternative delegation options.
* CA operators SHOULD provide technical assistance for resource migration where possible.
* Registry operators MUST ensure continuity of service for affected resources during transition.

8.3.3. Final Procedures

* CA operators MUST revoke all issued certificates in an orderly manner.
* CA operators MUST coordinate final revocation and cleanup procedures with registry operators.
* Registry operators MUST update delegation records and hierarchy information.

9. Registry Operator Responsibilities

9.1. Monitoring and Enforcement

Registry operators bear responsibility for oversight of delegated CAs and enforcement of operational standards.

9.1.1. Monitoring Infrastructure

Registry operators MUST:

* Implement automated monitoring of all delegated CAs with appropriate alerting thresholds.
* Establish clear Service Level Agreements (SLAs) and enforcement procedures.
* Provide operational dashboards and reporting for CA operators.
* Maintain emergency contact procedures for critical issues.

9.1.2. Enforcement Escalation

Registry operators SHOULD implement progressive enforcement: 1. Automated monitoring alerts and initial operator notification 2. Formal operator notification within 24-48 hours of issue detection 3. Public visibility of persistent issues within one week 4. Revocation procedures for issues persisting longer than 60-90 days

The specific timeframes MAY be adjusted based on the severity of the operational issue and its impact on the RPKI ecosystem.

9.1.3. Documentation and Communication

* Registry operators MUST maintain clear documentation of monitoring criteria and enforcement procedures.
* Registry operators MUST provide regular communication about enforcement actions and their rationale.
* Registry operators SHOULD coordinate enforcement actions with other registry operators when appropriate.

9.2. Support and Community Engagement

Registry operators SHOULD provide comprehensive support for delegated CA operators.

9.2.1. Support Services

* Registry operators SHOULD maintain comprehensive operational guidance documentation.
* Registry operators SHOULD provide training and certification programs for CA operators.
* Registry operators SHOULD maintain community forums and troubleshooting resources.
* Registry operators SHOULD conduct regular operational reviews and provide feedback to CA operators.

9.2.2. Community Coordination

* Registry operators SHOULD coordinate with other registry operators on operational standards and procedures.
* Registry operators SHOULD participate in operational working groups and standards development.
* Registry operators SHOULD share operational experience and lessons learned with the broader community.

10. Implementation Considerations

10.1. Deployment Strategies

Organizations implementing these guidelines SHOULD consider phased deployment approaches.

10.1.1. Phased Implementation

Phase 1: Establish monitoring and measurement baseline

* Implement comprehensive monitoring systems
* Establish baseline metrics for current operations
* Begin collecting data on CA performance and validator impact

Phase 2: Implement basic operational standards

* Deploy availability and reliability requirements
* Establish publication discipline practices
* Implement automated alerting and response procedures

Phase 3: Advanced operational features

* Deploy advanced monitoring and analytics
* Implement predictive failure detection
* Establish community reporting and feedback mechanisms

Coordination Requirements:

* Registry operators SHOULD coordinate implementation timelines with CA operators.
* Registry operators SHOULD provide adequate notice for new requirements and enforcement procedures.
* CA operators SHOULD plan infrastructure upgrades to meet new requirements within reasonable timeframes.

10.2. Tooling and Automation

Effective implementation of these guidelines requires appropriate tooling and automation.

Recommended Tools:

* RPKI validator software for continuous validation testing
* Monitoring systems with RPKI-specific metrics and alerting
* Configuration management systems for consistent deployments
* Automated backup and recovery systems

Open Source Resources:

* CA operators SHOULD leverage existing open source RPKI tools where appropriate.
* The community SHOULD develop and maintain reference implementations of monitoring and validation tools.
* Registry operators SHOULD contribute to open source tooling development.

Integration Considerations:

* Tools SHOULD integrate with existing operational infrastructure.
* Monitoring systems SHOULD provide standard APIs for integration with other operational tools.
* CA operators SHOULD implement automation carefully to avoid creating new failure modes.

11. IANA Considerations

This document has no IANA actions.

12. Security Considerations

Implementation of these operational guidelines has several security implications that MUST be carefully considered.

Operational Security

* Enhanced monitoring may expose operational details that could be useful to attackers.
* Automated systems MUST be secured against manipulation or abuse.
* Emergency procedures MUST balance rapid response with security controls.
* Key management and rotation procedures MUST be maintained even during operational issues.

Ecosystem Security

* Revocation procedures MUST NOT be exploitable for denial of service attacks.
* Monitoring systems MUST be protected against false reporting or manipulation.
* Community reporting mechanisms MUST prevent abuse while encouraging legitimate feedback.

Privacy Considerations

* Monitoring data may contain sensitive operational information.
* Public reporting of CA issues MUST balance transparency with operational privacy.
* Contact information and communication procedures MUST be protected appropriately.

Risk Management

* Organizations MUST balance operational requirements with security controls.
* Incident response procedures MUST account for potential security implications.

* Regular security reviews SHOULD be conducted for operational systems and procedures.

13. References

13.1. Normative References

[I-D.ietf-sidrops-publication-server-bcp]
Bruijnzeels, T., de Kock, T., Hill, F., Harrison, T., and J. Snijders, "RPKI Publication Server Best Current Practices", Work in Progress, Internet-Draft, draft-ietf-sidrops-publication-server-bcp-04, 15 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-publication-server-bcp-04>.

[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>.

13.2. Informative References

[CURE-NDSS24]
Mirdita, D., Schulmann, H., Vogel, N., and M. Waidner, "The CURE to Vulnerabilities in RPKI Validation", In Proceedings of the Network and Distributed System Security (NDSS) Symposium 2024, February 2024, <https://dx.doi.org/10.14722/ndss.2024.241093>.

[DEAD-CA] Snijders, J., "Non-functional RPKI Certification Authorities", 2 September 2025, <https://console.rpki-client.org/nonfunc.html>.

[RFC3647] Chokhani, S., Ford, W., Sabett, R., Merrill, C., and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 3647, DOI 10.17487/RFC3647, November 2003, <https://www.rfc-editor.org/info/rfc3647>.

[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>.

[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>.

[RFC6484] Kent, S., Kong, D., Seo, K., and R. Watro, "Certificate Policy (CP) for the Resource Public Key Infrastructure (RPKI)", BCP 173, RFC 6484, DOI 10.17487/RFC6484, February 2012, <https://www.rfc-editor.org/info/rfc6484>.

[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>.

[RFC7020] Housley, R., Curran, J., Huston, G., and D. Conrad, "The Internet Numbers Registry System", RFC 7020, DOI 10.17487/RFC7020, August 2013, <https://www.rfc-editor.org/info/rfc7020>.

[RFC8181] Weiler, S., Sonalker, A., and R. Austein, "A Publication Protocol for the Resource Public Key Infrastructure (RPKI)", RFC 8181, DOI 10.17487/RFC8181, July 2017, <https://www.rfc-editor.org/info/rfc8181>.

[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>.

[RFC9286] Austein, R., Huston, G., Kent, S., and M. Lepinski, "Manifests for the Resource Public Key Infrastructure (RPKI)", RFC 9286, DOI 10.17487/RFC9286, June 2022, <https://www.rfc-editor.org/info/rfc9286>.

Appendix A. GITHUB

This I-D and related documents, code and AI analysis are available at https://github.com/IEISI-ORG/bcp-rpki-ca-prop (https://github.com/IEISI-ORG/bcp-rpki-ca-prop) .

Author’s Address

Terence Charles Sweetser
Internet Engineering & Infrastructure Strategic Initiative
Wynnum, QLD
Australia
Phone: +61 447 069 725
Email: tcs@ieisi.org
URI: https://about.me/terry.sweetser