SIDN Labs

https://sidnlabs.nl

Tech Report

Title: A first look at common RPKI publication practices

Authors: Moritz Müller-Brus, Lisa Bruder, Caspar Schutijser, Ralph Koning

Link to arXiv: https://doi.org/10.48550/arXiv.2512.16369

A first look at common RPKI publication practices

MORITZ MÜLLER-BRUS, SIDN Labs, The Netherlands and University of Twente, The Netherlands
LISA BRUDER, SIDN Labs, The Netherlands
CASPAR SCHUTIJSER, SIDN Labs, The Netherlands
RALPH KONING, SIDN Labs, The Netherlands and University of Amsterdam, The Netherlands

The Resource Public Key Infrastructure (RPKI) is crucial for securing the routing system of the Internet. With the RPKI, owners of Internet resources can make cryptographically backed claims, for example about the legitimate origin of their IP space. Thousands of networks use this information to detect malicious or accidental route hijacks. The RPKI consists out of 100 distributed repositories. However, public reports claim that some of these repositories are unreliable. A current Internet-Draft suggests best practices on how to operate these repositories, with the goal to improve deployment quality.

Inspired by this draft, we take a first look at the operational practices of repositories of the RPKI. We mainly focus on the *distribution* of RPKI information. We find that there is a wide variety in deployment practices, of which some might risk the availability of parts of the information in the RPKI. This study creates a baseline for measuring the maturity of RPKI repositories in the future.

1 Introduction

The RPKI enables holders of Internet resources, like IP address space, to make cryptographically backed claims about their assets [12]. Currently, the RPKI mostly contains claims about IP prefixes and the networks from which those prefixes may be announced in so called Route Origin Authorisations (ROAs). Networks run Relying Party (RP) software to fetch and validate these objects and then base their Border Gateway Protocol (BGP) routing decisions on the validation results. At the moment, almost 60% of all IP prefixes are covered by a ROA [15], thereby protecting these prefixes against malicious attacks and misconfiguration. The validation of ROAs is also on the rise [14].

The RPKI relies on "a distributed repository system for storing and disseminating" its objects [12]. Most of these objects, like ROAs, are managed and distributed by one of the five Regional Internet Registries (RIRs) (AFRINIC, APNIC, ARIN, LACNIC, RIPE NCC). At the same time, the RPKI enables resources holders to operate their own Certificate Authority (CA). Then, resource owners can decide to publish the objects themselves or, if provided, rely on the publication service by their RIR. The RIR or the resource owner needs to make sure that the objects can be retrieved by RPs using the HTTP based RPKI Repository Delta Protocol (RRDP) or rsync.

Besides the repositories by the 5 RIRs, 95 repositories are currently listed in the RPKI. In recent years, operators of RP software frequently complained about repositories that were unreliable [13]. Aside from putting extra burden on RP software, unavailable repositories can also negatively impact the security of Internet resources. If a repository is unavailable for too long (e.g. several hours), objects published by it can not be updated and potentially expire, thereby effectively removing the protection by the RPKI. In certain scenarios, Internet resources covered by objects could even become unavailable for (parts of) the Internet.

This, likely, motivated the authors of *RPKI Publication Server Best Current Practices* to formulate a set of practices that operators of publication servers should consider implementing. The IETF working group "SIDR Operations" has adopted this document as an Internet-Draft [2].

Inspired by this effort, our goal is to measure the current RPKI publication practices. Thereby, we also create a baseline measurement of the implementation of the Best Current Practices (BCPs).

Authors' Contact Information: Moritz Müller-Brus, SIDN Labs, The Netherlands, Arnhem and University of Twente, The Netherlands, Enschede, moritz.muller@sidn.nl; Lisa Bruder, SIDN Labs, The Netherlands, Arnhem, lisa.bruderr@sidn.nl; Caspar Schutijser, SIDN Labs, The Netherlands, Arnhem, caspar.schutijser@sidn.nl; Ralph Koning, SIDN Labs, The Netherlands, Arnhem and University of Amsterdam, The Netherlands, Amsterdam, ralph.koning@sidn.nl.

Note that the goal of this study is not find out *why* some operators run their own publication server, nor to measure the performance of the publication servers. Our contributions are as follows:

- Provide a first look at RPKI publication practices, showing that there are only a few common practices most operators implement.
- Create a baseline measurement for tracking the impact of the recommended best practices.
- Highlight widely diverging practices between operators, showing that there is not *the* best way to publish objects in the RPKI.

2 RPKI Basics

In the RPKI, signed objects are grouped together into publication points based on the signing CA. Every group of objects is accompanied by a manifest that lists all signed objects and a Certificate Revocation List (CRL) that lists all objects that have been revoked.

The 5 RIRs serve as trust anchors in the RPKI and the private keys associated with their CA certificates are used to sign certificates for subordinate CAs as well as resource certificates [11]. Resource holders can let their RIR manage the signing of their objects (so-called hosted RPKI). Alternatively, it is possible to request a delegated CA certificate from an RIR and manage signing independently (delegated RPKI) [11].

Objects also need to be hosted for RPs to retrieve them. Objects stored in repositories in the RPKI are distributed using rsync and RRDP servers [3]. When resource holders make use of the "hosted" RPKI service of their RIR, the objects are hosted through the RIR's RRDP and rsync servers. However, objects can also be distributed through self-hosted rsync and RRDP servers [11].

RRDP is an HTTP based protocol. A CA certificate contains a references to a notification file, which is downloaded by an RP, e.g. every 10 minutes. This file includes a reference to a snapshot file, which contains the full content of the repository. Additionally, the notification file can contain a list of delta files, listing incremental changes to the repository [3]. An RP checks if it can use the delta files to bring its cache up to date or if it needs to download the full snapshot file. If the RRDP server is not reachable, the rsync server can be used to retrieve the objects.

If neither the RRDP nor the rsync server are reachable for a prolonged time, no new certificates and signed objects can be distributed. Then, the objects signed by the CA could expire. Because current practice does not advice operators to drop routes with a status of "unknown", the impact of an expired ROA object is limited [4]. However, in specific situations where a prefix is covered by two ROAs associated with different Autonomous System Numbers (ASNs), an announcement could become invalid if one of the ROA objects expires [7, 10].

3 Methodology

The Internet-Draft "RPKI Publication Server Best Current Practices" (I-D) [2] describes best current practices for the operation of a publication, rsync and RRDP servers based on operational experience. This methodology applies to version 05 of the I-D. In this section, we briefly describe the goal of the recommended practices and the measurements that we performed, grouped by the section in the I-D they appear in. Each measurement has a short name that we use to refer to the measurement in the paper, printed in (**bold**). We refer the reader to the I-D for a more detailed description and motivation for each recommended practice.

We use RP software rpki-client v9.6 to download the current data published in the RPKI. The rpki-client cache serves as a basis for our analysis. It contains all files that have a valid signature path. Our measurement vantage point, located in a well connected network , supports both IPv4 and IPv6 and we fetch the content over whatever protocol preferred by rpki-client. For each CA certificate we encounter in the RPKI data we extract the Uniform Resource Locators (URLs) of the

RRDP and rsync servers from the Subject Information Access extension fields.[1] We use the URLs of the RRDP server as the identifier for each repository.

For the most part of this paper, we focus on the measurements run on December 6 2025. We validated that our measurement are representative for normal repository behaviour by performing two additional measurements two days before and two days after the initial measurement.

3.1 Measurements

Hostnames (I-D Section 5). The I-D recommends the use of different hostnames for the publication, RRDP and rsync server to allow for dedicated infrastructure. For each CA certificate, we compare if the Fully Qualified Domain Name (FQDN) of the URL of the RRDP and the URL of the rsync server are different at different levels of the DNS hierarchy. We compare the difference of the hostname (left most label, **Different hostnames**), the 2nd level domain name (the first label left of the Top-Level Domain (TLD) - we define the TLD using the public suffix list, **Different subdomains**), and the TLD (**Different TLDs**).

Also, we verify if the A and AAAA records of the hostnames are validly signed using Domain Name System Security (DNSSEC) (**DNSSEC (RRDP)** and **DNSSEC (rsync)**).

IP Address Space and Autonomous Systems (I-D Section 6). If a repository hosts ROAs that cover the IP address space used for its own RRDP and rsync server, there is a risk of the repository making itself unreachable, without the ability to publish remediating objects. The I-D therefore recommends that setups like this should be avoided. To analyse the dependency of a repositories' reachability on its own ROA objects (**Reachability ind. of ROA in repo**), we loop through the ROAs distributed by the repository and look for a ROA that covers the prefix(es) we determined for the service endpoints. If we find a matching ROA, we mark this prefix as being dependent on its own repository. A repository is deemed as independent of its own endpoints if none of the IP address space is covered by a ROA in the repository.

Furthermore, the BCP recommends hosting the RRDP and rsync server in different networks. We query the A and AAAA for the Uniform Resource Identifiers (URIs) of the servers and determine the matching routed prefix for each IP address. We do this using the Network Info API of RIPEstat [16]. Based on this we determine if the RRDP and rsync servers of a repository are hosted in different networks (**RRDP and rsync in diff. networks**). We mark this as fulfilled when the repository uses different prefixes for both servers and both IP address families. Having no IP addresses of a family type for one of the servers does not count as a different network.

Same origin URIs (I-D Section 7.1). The I-D remarks that as defined in RFC9674, the "uri" attributes in notification files need to refer to the same URI for all listed snapshot and delta files [18]. We download the content of each notification file and analyse the URIs in it pointing to delta and snapshot files. If they point to the same host as the notification file URI, we deem the RRDP server as compliant to this requirement (**Same origin**). In addition, we check if a HTTP redirect occurs when fetching the content.

Endpoint Protection (I-D Section 7.2). This BCP recommends the use of access control to protect RRDP endpoints. Here, we send an HTTP request to each URL pointing towards a notification file and record the returned HTTP status using the following methods: GET, OPTIONS, HEAD, POST, PUT, PATCH, DELETE (**Endpoint protection**). The RFCs defining the RRDP protocol do not specify which methods and parameters are mandatory. We assume that a web server implements

[1] e.g. https://rrdp.ripe.net/notification.xml and rsync://rpki.ripe.net/repository/DEFAULT/ for the CA cert. of the RIPE NCC

endpoint protection if it only answers with status code 200 for the methods GET and HEAD and with a status code between 400 and 599 for the methods PATCH and DELETE.[2]

Bandwidth and Data Usage (I-D Section 7.3). This practice recommends to support compression to save bandwidth. Our test consists of downloading each notification file. If the server serves the notification file in a compressed form (i.e., when the *Content-Encoding* header is set to something like *gzip* or *deflate*), we mark the RRDP server as compliant with this requirement (**Compression**).

Content Availability (I-D Section 7.4). The I-D's authors recommend to use a Content Delivery Network (CDN) to achieve high availability. We use two methods to determine whether a RRDP Server relies on a CDN to distribute its content (**CDN**).

First, we test if the IP addresses of the RRDP Server are announced using BGP anycast using a list IP prefixes compiled by LaCeS [5]. We use a list of IP prefixes compiled by LaCeS [5]. Second, we test whether the IP addresses of the RRDP server are announced by Autonomous System (AS) numbers that are categorized as *CDN* by bgp.tools [1]. To find out which AS (or ASes) announces a particular IP address, we use a data set provided by iptoasn.com [9].

For the most part of the paper, we classify an endpoint as relying on a CDN if either of the two methods return a positive result.

Limit Notification File Size (I-D Section 7.5). This BCP recommends practices to reduce the amount of data a RP needs to retrieve. For each repository, we collect the size of the notification, snapshot and delta file. We take the sum of the size of each delta files listed in the corresponding notification file (**Snapshot > Deltas**). To save resources of the RRDP server, we attempt to look up the file size indicated in the response header and stop the download afterwards whenever possible.

Also, we measure how long the RRDP server keeps individual delta files (**Delta older than 4h**) by downloading each notification file every 30 seconds for 24 hours. We record when we have observed each delta file for the first and for the last time. We use this information also to measure the update frequency of the publication server (**Delta frequency larger than 1min**). Note that we can only identify update frequencies of 30 seconds or longer due to our measurement frequency.

Manifest and CRL Update Times (I-D Section 7.6). The I-D discusses up- and downsides of longer and shorter validity periods for manifests and CRLs. It does not give a clear recommendation. To understand which validity times are common, we measure the average for each repository. We collect the update times from manifests and CRLs (**Manifest/CRL validity**). For both, we extract the "next update" and "this update" fields and report the average difference between the two.

3.2 Not measured practices and ethical considerations

We focus on practices that we can measure externally, without the need of ingesting content into the RPKI, thus excluding the publication server. Other aspects can only be observed in case of problems at the CA (e.g. I-D Section 4.4 Data Loss). Also, we do not measure Notification File Timing (I-D Section 7.7.1) and L4 Load-Balancing (I-D Section 7.7.2) as those require frequent, distributed measurements with low chance of actually finding mis-behaviour. Finally, we do not measure the implementation of the rsync server. Some recommendations are not visible for an external observer (e.g. parts of Load Balancing and Testing, I-D Section 8.3) or require highly frequent measurements, possibly draining the resources of the server (e.g. Consistent Content, I-D Section 8.1).

In our measurements, we do not put an unreasonable burden on the measured infrastructure. One measurement consists of 2 runs of the rpki-client. Overall, we perform 3 measurements spread over 6 days. The collected data is public and does not contain personal identifiable information.

[2]This is in accordance with Nmap (https://nmap.org/nsedoc/scripts/http-methods.html).

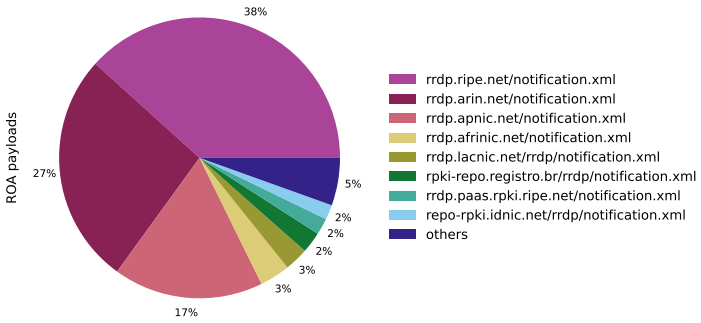


Fig. 1. Distribution of ROA payloads rounded to percent

4 Results

4.1 Overview of repositories

In this subsection, we provide an overview of the observed repositories and the implemented BCPs. In the later subsections, we discuss in more detail who implements the different BCPs and how.

We observed 100 unique repositories (identified by URLs pointing to RRDP notification files). For 10 of these 100 repositories, we were unable to fetch the notification file for different reasons. We do not take these 10 repositories into account for our analysis. Out of the 90 remaining URLs, 26 point to notification files hosted by AWS. These 26 URLs are used by individual repositories, but they all implement the same practices. If necessary, we point out the effect of the repositories by AWS on our results.

ROA payloads. Figure 1 shows the distribution of ROA payloads. The RIRs take up the first 5 positions and with that 91.12% of ROA payloads distributed through the RPKI. All repositories in the top 8 of biggest repositories are hosted by RIRs and National Internet Registries (NIRs). When grouping Amazon Web Services (AWS)' repositories together, they come in at position 5 with 2.88% of the total ROA payloads.

In comparison, 81.93% of ROA objects (potentially containing more than one payload) are distributed through the servers of the 5 RIRs. With 5.08% of ROAs, the biggest repository not hosted by an RIR is the of NIR registro.br. Figure 6 visualising the distribution of ROA objects can be found in the appendix. This shows that even though more than 80 self-hosted repositories exist, the vast majority of Internet address owners still rely on the publication servers and repositories of their parents, following the recommendation in I-D Section 4.1.

Overview of implemented BCPs. Figure 2 shows the share of repositories that implement a certain BCP. The only BCP that all repositories implement is the *same origin* policy and recommendations regarding the update of delta files. In contrast, only one of the repositories use different TLDs for their RRDP and rsync servers and only 24% host their RRDP and rsync server in different networks.

Figure 2 also shows how many ROA payloads are served by repositories implementing each BCP. While some BCPs are only implemented by a minority of servers, these servers provide still more than 80% of all ROA payloads in the RPKI. Examples include "Different Hostnames", or the recommendation "Reachability independent of ROA in repository". Overall, we find that repositories responsible for a large number of ROA payloads do not necessarily implement more BCPs: These numbers only weakly positively correlate (Pearson correlation $R = 0.270$).

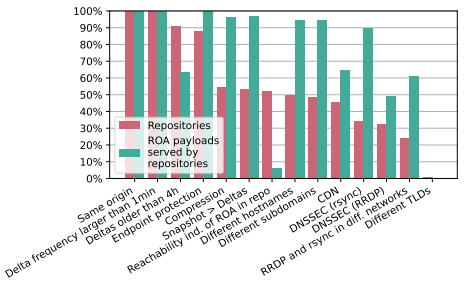


Fig. 2. Share of repositories implementing a BCP (red bars). ROA payloads served by repositories implementing each BCP (green bars).

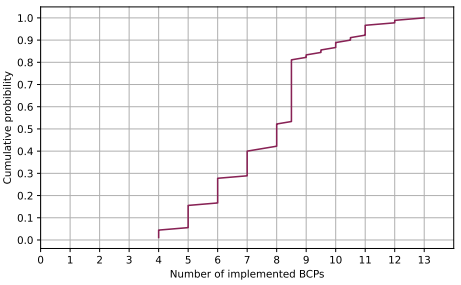


Fig. 3. CDF of the number BCPs each repository implements

Individual repositories. Figure 3 shows that 50% of the repositories implement at least 8 out of the 13 BCPs. The repositories operated by AWS influence this metric (jump at 8.5 BCPs in Figure 3).[3] When treating the repositories by AWS as one single repository, the median number of implemented BCPs drops to 7.

The first part of Table 1 shows how the measured BCPs are implemented by the RIRs. The second half shows the same, but for NIRs. For each repository, the table reports the number of ROAs, and whether the practices as outlined in Section 3 are followed. We especially focus on these repositories since, overall, they host the majority of ROAs. A list of the 20 largest repositories with regards to ROA count, excluding those hosted by the RIRs and NIRs, is part of the appendix (Table 2).

4.2 Hostnames

45 repositories (50%) use different hostnames for the RRDP and rsync server. One repository uses different TLDs.

8 out of 9 RIR hosted repositories use different hostnames for their RRDP and rsync servers. Only the *rpki.rand.apnic.net* repository does not comply with this recommendation. Within the group of NIR repositories, only 1 repository makes use of different hostnames. One other noteworthy repository is *magellan.ipxo.com* as it is the only repository that uses a subdomain of its RRDP server hostname for its rsync server.

22 (24%) repositories sign their A and AAAA records for both, their RRDP server and their rsync server. These servers are responsible for 52% (RRDP) and 92% (rsync) of all ROA payloads, including 5 out of 9 servers operated by the RIRs (see Table 1). The large difference between RRDP and rsync is caused by the RIPE repositories. RIPE NCC hosts the zone of the rsync server itself, while Akamai hosts the zone of the RRDP server. The RIPE NCC confirmed us that in their specific setup, Akamai does not support DNSSEC.

4.3 IP address space and autonomous systems

IPv4 and IPv6. Repositories use a mean of 3.24 IPv4 addresses and a mean of 3.14 IPv6 addresses to host their repository (rsync and RRDP combined). The median for both address families is 1. There is 1 repository that makes use of unreachable IP addresses for its endpoints.

On average more IP addresses are used for the RRDP server compared to the rsync server, 2.57 and 1.08 respectively. When only looking at IPv6 addresses, this difference is even bigger: on average, 3.01 IPv6 addresses are used for RRDP servers and only 0.5 for rsync servers.

[3]The servers of AWS *partially* implement **RRDP and rsync in diff. networks**.

	ROAs	Different hostnames	Different TLDs	Different subdomains	DNSSEC (RRDP)	DNSSEC (rsync)	Reachability ind. of ROA in repo	RRDP and rsync in diff. networks	Same origin	Endpoint protection	Compression	CDN	Snapshot > Deltas	Delta frequency > 1min	Deltas > 4h
Repositories by RIRs															
rrdp-rps.arin.net/...	2091	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗
rrdp.arin.net/...	183288	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✗
rrdp.apnic.net/...	16677	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
rpki.rand.apnic.net/rrdp/...	6	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓
rrdp.sub.apnic.net/...	196	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
rrdp.afrinic.net/...	10975	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
rrdp.lacnic.net/rrdp/...	15757	✓	✗	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✗
rrdp.paas.rpki.ripe.net/...	6583	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
rrdp.ripe.net/...	45338	✓	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Repositories by NIRs															
rpki-repo.registro.br/rrdp/...	17421	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗
repo-rpki.idnic.net/rrdp/...	8464	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗
rpki-repository.nic.ad.jp/rrdp/ap/...	5533	✗	✗	✗	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✓
rrdp.twnic.tw/rrdp/...	3929	✓	✗	✓	✓	✓	✗	✗	✓	✓	✗	✓	✓	✓	✓
rpki.cnnic.cn/rrdp/...	2047	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓

Table 1. Implemented BCPs by repositories operated by RIRs and NIRs

Endpoints' reachability dependent on ROAs in repository. 40 repositories (44%) distribute at least a subset of the ROAs covering the IP ranges used for their service endpoints through their own repository. The remaining repositories either distribute ROAs for the used IP ranges through a different repository or do not publish ROAs for them at all.

Use of different networks. 37 repositories (41%) make use of different networks for their RRDP and rsync endpoints for at least one IP address family. 6 repositories use different networks for both IPv4 and IPv6, 5 of those are repositories hosted by RIRs.

4.4 RRDP server

Same origin URIs. All repositories host the delta and snapshots files under the same hostname as the notification file and are therefore compliant with the same origin recommendation. One repository makes use of an HTTP redirect when content is being fetched. However, this repository does not host any ROAs.

Endpoint Protection. 79 repositories (88%) deploy endpoint protection. For the RRDP servers included in Table 1 and in the largest 20 remaining servers, there are only 2 servers that do not deploy endpoint protection. That means that most of the servers failing our test deploy a small amount of ROAs.

Bandwidth and Data Usage. 55 repositories (61%) support HTTP compression. 2 repositories operated by RIRs do not support compression, but one of them hosts just 6 ROAs. With the NIR-operated repositories, 1 repository does not support compression. 5 out of the 20 largest remaining servers do not support compression. As such, most of the repositories that do not support compression reside in the "long tail" of smaller repositories.

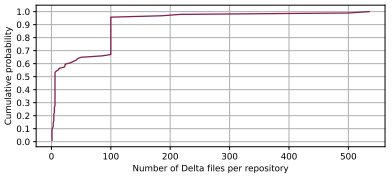


Fig. 4. Distribution of number of delta files per repository

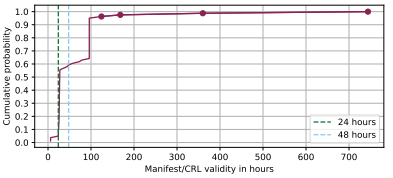


Fig. 5. Distribution of average manifest/CRL validity per repository marking the ones with > 4 days

Content Availability. 41 repositories (46%) use a CDN. Among the RIRs, ARIN does not use a CDN. There are 3 repositories listed for APNIC's, 2 of which use a CDN, and one of them does not. However, the latter hosts just 6 ROAs. Furthermore, LACNIC's server also does not use a CDN. Among the NIRs, we notice that all but one of the NIRs do *not* use a CDN to distribute their data. Among the other repositories, all repositories that have more than 1,000 ROAs use a CDN.

Notification file size. Since the content of the repository can change, we requested their contents 3 times over the course of 6 days. 81 (90%) RRDP servers never serve snapshot files exceeding 15,516 KB. The total size of delta files never exceeds 13.554 KB for 90% of the RRDP servers. See for more details Figure 7 in the appendix. The RRDP server of ARIN serve the largest snapshot file (623,152 KB), while a server of AWS serves the largest total size of delta files (144,535 KB). For the majority of the repositories, the size of the snapshot file stays stable over the course of our measurements (standard deviation $phi < 26\ KB$ for 90% of all repositories). However, the total size of the delta files varies stronger (for 17% of the repositories $phi > 100\ KB$). This is mostly the case for repositories with presumably high activity, like the ones by the RIRs.

For 44 (49%) repositories, the sum of delta files is always larger than the snapshot file (5 times larger on average). These repositories thus do not follow the recommendation. Again, the repositories by AWS stand out, with the total size of delta files exceeding the size of the snapshot files between 4 and 104 times. Instead, the RRDP servers by AWS handle a maximum number of 100 deltas files (see jump at 100 at Figure 4). Also the main repository by APNIC and the repository by the NIR registro.br have enforced a limit of 500 and 50 delta files respectively.

74 (82%) of all repositories keep delta files for at least 4 hours, thus follow the recommended practice. 6 never follow this recommendation and 10 do not show consistent behaviour. 58% of all repositories always keep their delta files for at least 1 day (the maximum time period we measured). Finally, 83 (92%) of the measured repositories always publish new delta files less frequently than once every minute.

Manifest/CRL validity. 82 repositories (91%) publish minimum one valid manifest or CRL. The remaining repositories do not host any manifests and CRLs or none with a valid signature chain.

Figure 5 shows the distribution of the average validity time per repository. Here, we only include repositories with minimum one valid manifest or CRL. Because the average manifest validity does not differ significantly from the average CRL validity for any of the repositories,[4] we discuss them together. The I-D describes that "one large repository" decided to adjust their reissuance cycle from once every 24 hours to once every 48 hours. To illustrate where most repositories fall with regards to that, the 24 and 48 hour timestamps are marked in Figure 5.

[4]There are 2 repositories where it does differ, however, by less than 2 milliseconds.

Most repositories have an average manifest and CRL validity time between 1 and 2 days (median 1 day and 03:15:50). When rounding the validity period to days, 1 day is the most common value with 43 repositories (48%) of the evaluated repositories, followed by 4 days with 26 repositories (27%). These 26 are all hosted by AWS. The minimum average validity is 6 hours (3 repositories), the maximum average validity is 31 days 00:05:00 (1 repository). Only 4 repositories (4%) have a average validity longer than 4 days, they are marked in Figure 5.

Rounded to days, RIPE's main repository has an average validity period of 1 day. ARIN's and AFRINIC's main repositories have an average of around 2 days. For LACNIC and APNIC we see longer average validity periods of 5 and 7 days respectively.

5 Discussion

5.1 Common ground and uncommon practices

There is some common ground on how to operate a repository, even before the best practices document is published. From Table 1, we can derive that serving the content from the same origin and enabling some basic form of endpoint protection, are two configuration knobs most operators can agree on.

In contrast, some suggested practices are rarely implemented. For example, almost none of the operators choose for different TLDs for the rsync and RRDP servers and also many domain names of RRDP endpoints are not DNSSEC-signed.

Some practices are mostly implemented by repositories that host many ROA payloads, for example relying on different hostnames and distributing RRDP servers using CDNs. While the use of a CDN might be perceived as unnecessary for many small repositories, the security of their network resources depends on the availability of their repositories. Putting effort into highly available RRDP servers should therefore not be underestimated.

5.2 Potentially problematic practices

We observed some practices that should be adjusted because they may cause unnecessary burden for RPs or put the availability of the repository at an unnecessary risk.

RRDP servers providing larger snapshot and delta files and their clients could benefit the most from HTTP compression. However, as discussed in Section 4.4, some RRDP servers do not support compression and we find that the median snapshot and delta file size that these servers serve is higher than of the servers that do support compression.

Also, the I-D does not give a clear recommendation on manifest/CRL validity but remarks that longer and shorter durations have certain up- and downsides.

One downside of short validity periods is that ROAs might become invalid while a repository is unreachable. Of the 3 repositories with an average validity time of less than 1 day (6 hours), 2 distribute less than 10 ROA payloads. One however, *rpki.ccnic.cn*, distributes over 2000 ROA payloads and the short validity time could lead to ROAs becoming invalid rather quickly if the repository becomes unavailable for example due to unexpected operational issues.

The high average validity periods of 15 and 31 days that we measured for 2 repositories[5] makes these more vulnerable to potential replay attacks with old ROA objects [10].

5.3 Alternatives to BCPs

As also noted in the I-D, there might be good reasons to diverge from the suggested practices.

For example, not using a CDN does not mean a service cannot have a good level of availability. ARIN's server, for instance, is classified as not using a CDN. However, the hostname of that server

[5]Seen for *rpki-repository.nic.ad.jp* and *akane.maru.co.jp* respectively.

(*rrdp.arin.net*) resolves to 6 IPv4 addresses and 6 IPv6 addresses. For both address families, 3 of the 6 IP addresses are announced by AS 394018, and the other 3 are announced by AS 393225, also connected at different Internet Exchange Points (IXPs). As such, in terms of geographical location and connectivity to the Internet, the 2 groups of servers serving *rrdp.arin.net* seem somewhat independent from each other.

Also, there might be good arguments for not signing the Domain Name System (DNS) records of the publication servers with DNSSEC. DNS spoofing would only have limited effect since the integrity and authenticity of the information in RPKI itself is cryptographically protected.

5.4 Measurement limitations

We faced several limitations. First, there is not a definite way of measuring compliance with **Endpoint protection** and the I-D does not provide guidelines. We picked an interpretation of this recommendation that can be measured (see Section 3.1), but acknowledge that the test could cover more scenarios. Failing our test does not mean that the configuration of the server is wrong.

Also, while we did observe that a substantial amount of repositories make use of different hostnames for their RRDP and rsync servers, those hostnames could still rely on the same underlying DNS infrastructure (**Different hostnames**).

Next, the use of different IP addresses does not necessarily tell us about the redundancy and independency of the underlying infrastructure (**RRDP and rsync in diff. networks**). The different IPs could still lead to the same physical infrastructure.

Finally, some repositories might make use of different networks for their endpoints, where some are covered by a ROA in different repositories (**Reachability ind. of ROA in repo**). We did not measure if there are any ROAs covering (part of) the IP address space in other repositories. These could be used to distribute remediating updates in case a ROA in the repository makes its own origination invalid and with that becomes temporarily unreachable.

6 Related Work

The performance of RPKI publication servers has been subject of study before. Hlavacek et al. [6] performed a measurement study into the resilience of RPs, identifying DNS as a weak link in the RPKI system. In another study, Hlavacek et al. [7] investigated ways in which RPs software can be attacked by malicious publication points. Our work complements Hlavacek et al.'s work because we look at RPKI repositories as opposed to RP software.

Huston [8] discusses the resilience of the RPKI, but does not contain measurements. Huston concludes that RPKI's resilience is at an acceptable level of resilience at the moment, but needs to be re-evaluated if RPKI were to get more or new roles.

Two policy documents at RIPE (published, not yet implemented [21]) and APNIC (waiting for endorsement by the Executive Council [19]) propose the "revocation of persistently non-functional RPKI CAs". We also encountered CAs of which the RRDP and rsync servers were unreachable or of which its content was not cryptographically valid, but in general we focus on the behaviour of the functional CAs and their publication servers.

Recently, Schulmann et al. [17] proposed a method to fingerprint RPKI repositories. While not the goal of our research, some of the implementation details studied in this paper could be useful for fingerprinting repositories (e.g. the chosen maximum number of delta files or the update frequency).

Currently, a new protocol called Erik is being developed in the Internet Engineering Task Force (IETF) [20]. Erik adds an additional distribution layer which could improve the availability of information in the RPKI, rendering some of the recommended practices in the I-D [2] less urgent.

7 Conclusion

Even before the "RPKI Publication Server Best Current Practices" I-D has become an official RFC, the major share of objects in the RPKI are already published by servers implementing many of the suggested best practices. At the same time, there is still plenty room for improvement, e.g. when increasing resilience against attacks and outages in the DNS. Lastly, there are operational practices that diverge significantly from the suggested best practices (e.g. the total size of the published delta files). Future work could study the motivation of these design choices. As soon as the RFC is published, measuring the uptake of the suggested best practices at publication servers should be considered.

References

[1] BGP.Tools. 2025. CDN Tag Export (CSV). https://bgp.tools/tags/cdn.csv. https://bgp.tools/tags/cdn.csv Accessed: 2025-12-11.

[2] Tim Bruijnzeels, Ties de Kock, Frank Hill, Tom Harrison, and Job Snijders. 2025. *RPKI Publication Server Best Current Practices*. Internet-Draft draft-ietf-sidrops-publication-server-bcp-05. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-sidrops-publication-server-bcp/05/ Work in Progress.

[3] Tim Bruijnzeels, Oleg Muravskiy, Bryan Weber, and Rob Austein. 2017. The RPKI Repository Delta Protocol (RRDP). RFC 8182. doi:10.17487/RFC8182

[4] Jerome Durand, Ivan Pepelnjak, and Gert Döring. 2015. BGP Operations and Security. RFC 7454. doi:10.17487/RFC7454

[5] Remi Hendriks, Matthew Luckie, Mattijs Jonker, Raffaele Sommese, and Roland van Rijswijk-Deij. 2025. LACeS: an Open, Fast, Responsible and Efficient Longitudinal Anycast Census System. arXiv:2503.20554 [cs.NI] https://arxiv.org/abs/2503.20554

[6] Tomas Hlavacek, Philipp Jeitner, Donika Mirdita, Haya Shulman, and Michael Waidner. 2022. Behind the Scenes of RPKI. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1413–1426. doi:10.1145/3548606.3560645

[7] Tomas Hlavacek, Philipp Jeitner, Donika Mirdita, Haya Shulman, and Michael Waidner. 2023. Beyond Limits: How to Disable Validators in Secure Networks. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 950–966. doi:10.1145/3603269.3604861

[8] Geoff Huston. 2025. Resilience in the RPKI. https://blog.apnic.net/2025/05/09/resilience-in-the-rpki/

[9] IPtoASN. 2025. IP-to-ASN Combined Database (ip2asn-combined.tsv.gz). https://iptoasn.com/data/ip2asn-combined.tsv.gz. https://iptoasn.com/data/ip2asn-combined.tsv.gz Accessed: 2025-12-11. IP-to-ASN combined data mapping IP address ranges to Autonomous System Numbers (ASN), country codes, and descriptions..

[10] NLnet Labs. 2025. Krill Documentation: Failure and Recovery Scenarios. https://krill.docs.nlnetlabs.nl/en/latest/failure-scenarios.html

[11] NLnet Labs. 2025. RPKI Documentation. https://rpki.readthedocs.io/

[12] Matt Lepinski and Stephen Kent. 2012. An Infrastructure to Support Secure Internet Routing. RFC 6480. doi:10.17487/RFC6480

[13] Dan Li, Li Chen, Yingying Su, and Yu Fu. 2025. *RPKI Repository Problem Statement and Analysis*. Internet-Draft draft-li-sidrops-rpki-repository-problem-statement-02. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-li-sidrops-rpki-repository-problem-statement/02/ Work in Progress.

[14] Weitong Li, Zhexiao Lin, Md. Ishtiaq Ashiq, Emile Aben, Romain Fontugne, Amreesh Phokeer, and Taejoong Chung. 2023. RoVista: Measuring and Analyzing the Route Origin Validation (ROV) in RPKI. In *Proceedings of the 2023 ACM on Internet Measurement Conference* (Montreal QC, Canada) *(IMC '23)*. Association for Computing Machinery, New York, NY, USA, 73–88. doi:10.1145/3618257.3624806

[15] National Institute of Standards and Technology (NIST). 2025. NIST RPKI Monitor — ROV All Invalid Originations View. https://rpki-monitor.antd.nist.gov/ROV/All/4. https://rpki-monitor.antd.nist.gov/ROV/All/4 Accessed: 2025-12-11.

[16] RIPE Network Coordination Centre (RIPE NCC). 2025. RIPEstat Data API — Network Info Endpoint. https://stat.ripe.net/docs/data-api/api-endpoints/network-info. https://stat.ripe.net/docs/data-api/api-endpoints/network-info Accessed: 2025-12-11.

[17] Haya Schulmann and Niklas Vogel. 2025. Poster: We must talk about RPKI Repositories. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (Taipei, Taiwan) *(CCS '25)*. Association for Computing Machinery, New York, NY, USA, 4755–4757. doi:10.1145/3719027.3760715

[18] Job Snijders. 2024. Same-Origin Policy for the RPKI Repository Delta Protocol (RRDP). RFC 9674. doi:10.17487/RFC9674

[19] Job Snijders. 2025. prop-166: Revocation of Persistently Non-functional RPKI Certification Authorities. https://www.apnic.net/community/policy/proposals/prop-166/

[20] Job Snijders, Tim Bruijnzeels, Tom Harrison, and Wataru Ohgai. 2025. *The Erik Synchronization Protocol for use with the Resource Public Key Infrastructure (RPKI).* Internet-Draft draft-ietf-sidrops-rpki-erik-protocol-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-sidrops-rpki-erik-protocol/01/ Work in Progress.
[21] Job Snijders and Nick Hilliard. 2025. RIPE-847: Revocation of Persistently Non-functional Delegated RPKI CAs. https://www.ripe.net/publications/docs/ripe-847/

Appendices

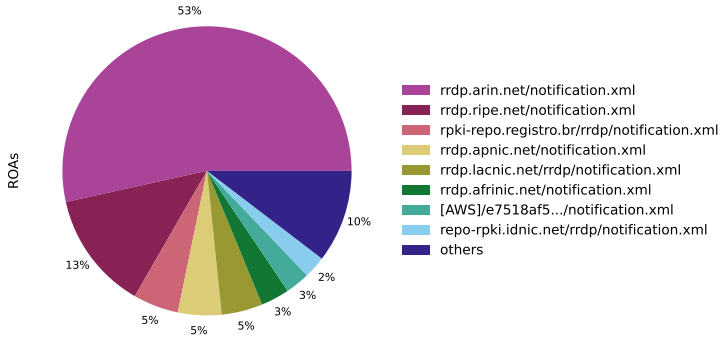


Fig. 6. Distribution of ROAs rounded to percent

	ROAs	Different hostnames	Different TLDs	Different subdomains	DNSSEC (RRDP)	DNSSEC (rsync)	Reachability ind. of ROA in repo	RRDP and rsync in diff. networks	Same origin	Endpoint protection	Compression	CDN	Snapshot > Deltas	Delta frequency > 1min	Deltas > 4h
[AWS]/e7518af5.../...	9352	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
[AWS]/20aa329b.../...	5327	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
[AWS]/f703696e.../...	4592	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
[AWS]/08c2f264.../...	1136	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
[AWS]/dba8f01c.../...	868	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
cloudie.rpki.app/rrdp/...	396	✓	✗	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓	✓
[AWS]/54602fb0.../...	339	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
[AWS]/517f3ed7.../...	310	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
rpki.admin.freerangecloud.com/rrdp/...	221	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓	✗	✓	✓	✓
magellan.ipxo.com/rrdp/...	210	✓	✗	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓
[AWS]/967a255c.../...	183	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
[AWS]/e72d8db0.../...	163	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
rpki.roa.net/rrdp/...	150	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓
krill.47272.net/rrdp/...	129	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓
rpki-01.pdxnet.uk/rrdp/...	119	✗	✗	✗	✓	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓
[AWS]/fe3737fb.../...	108	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓
ca.nat.moe/rrdp/...	99	✗	✗	✗	✓	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓
[AWS]/b8a1dd25.../...	89	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✓	✓
rrdp.rp.ki/...	76	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓
repo.rpki.space/rrdp/...	75	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓

Table 2. Implemented BCPs by 20 largest repositories by ROA count operated by other instances

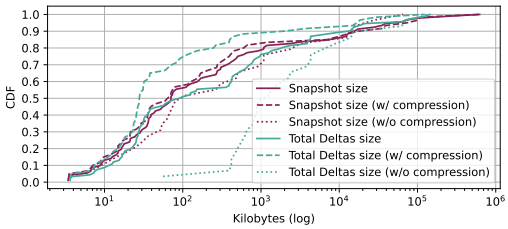


Fig. 7. Maximum snapshot and delta file size per repository over the course of 6 days served by all RRDP servers, and servers that support and do not support compression.