RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes

Romain Fontugne, Amreesh Phokeer, Cristel Pelsser, Kevin Vermeulen, Randy Bush

RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes

Romain Fontugne[1], Amreesh Phokeer[2], Cristel Pelsser[3], Kevin Vermeulen[4], and Randy Bush[1,5]

[1] IIJ Research Lab romain@iij.ad.jp
[2] Internet Society phokeer@isoc.org
[3] UCLouvain cristel.pelsser@uclouvain.be
[4] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
kevin.vermeulen@laas.fr
[5] Arrcus, Inc randy@psg.com

Abstract. As RPKI is becoming part of ISPs' daily operations and Route Origin Validation is getting widely deployed, one wonders how long it takes for the effect of RPKI changes to appear in the data plane. Does an operator that adds, fixes, or removes a Route Origin Authorization (ROA) have time to brew coffee or rather enjoy a long meal before the Internet routing infrastructure integrates the new information and the operator can assess the changes and resume work? The chain of ROA publication, from creation at Certification Authorities all the way to the routers and the effect on the data plane involves a large number of players, is not instantaneous, and is often dominated by ad hoc administrative decisions. This is the first comprehensive study to measure the entire ecosystem of ROA manipulation by all five Regional Internet Registries (RIRs), propagation on the management plane to Relying Parties (RPs) and to routers; measure the effect on BGP as seen by global control plane monitors; and finally, measure the effects on data plane latency and reachability. We found that RIRs usually publish new RPKI information within five minutes, except APNIC which averages ten minutes slower. At least one national CA is said to publish daily. We observe significant disparities in ISPs' reaction time to new RPKI information, ranging from a few minutes to one hour. The delay for ROA deletion is significantly longer than for ROA creation as RPs and BGP strive to maintain reachability. Incidentally, we found and reported significant issues in the management plane of two RIRs and a Tier1 network.

1 Introduction

The Border Gateway Protocol (BGP [1]) is the ubiquitous inter-domain routing protocol of the Internet. Unfortunately, like the rest of the early Internet, it was designed with no thought to security. One of the main efforts to secure BGP is the Resource Public Key Infrastructure [2, 3] (RPKI) which is an X.509-based system to share addressing and routing information assured by cryptographic methods. In the RPKI, Certificate Authorities (CAs), dominated by Regional

Internet Registries (RIRs), issue to ISPs resource certificates containing a list of IP prefixes allocated to them. ISPs use these certificates to create digitally signed attestations called Route Origin Authorizations (ROAs) to certify that a particular Autonomous System (AS) may advertise these prefixes. Other ISPs' routers can then use ROAs to validate incoming BGP announcements through a process called Route Origin Validation (ROV) [4] (see right side of Fig. 1).

The RPKI (a management plane) was designed to decouple and provide data redundant to the BGP control plane, allowing validation. Since operators have to apply and assess RPKI changes before updating BGP configurations, the overall routing operations are inevitably delayed by the time it takes to update and propagate RPKI data. However, the IETF specifications were lax in not specifying, or at least strongly recommending, timing parameters for the long linear RPKI management plane protocols (see *Management plane* in Fig. 1). BGP route updates propagate in less than a minute, two at worst [5, 6]. Therefore there is an expectation that propagation on the RPKI management plane is reasonably bounded; but we found that it takes on average over 25 minutes for APNIC and due to a bug we have reported as much as five hours for ARIN and LACNIC data to propagate; **orders of magnitude slower than BGP!** Ultimately RPKI updates should be applied as quickly as possible; long delays in the management plane increase the feedback loop for routing operations, increase the opportunities to let mistakes go unresolved, and increase the time needed to fix them [7, 8, 9, 10]. For example, NTT has documented three common oversights that lead to discrepancies between BGP announcements and RPKI data [11]: (1) a new prefix violating the maxLength attribute of an existing ROA, (2) announcing customer prefixes while the latter has not yet updated the corresponding ROAs (also a common for DDoS and BGP hijack mitigation [12]), (3) prefix migration from one AS to another. Since these inconsistencies between the management and control plane could lead to significant traffic loss in ROV-enabled networks [13, 14] the time it takes to fix ROAs and globally propagate them is of critical importance.

The goal of this paper is to measure the delays associated with the RPKI systems of the five RIRs and current ROV deployments by measuring the management, control, and data planes. We deploy experimental prefixes on the Internet and measure the management plane latency from ROA creation and subsequent publication by the RIRs to receipt by the routers, and then the resulting effects on the BGP control plane using RIPE RIS [15] data. We also measure some of the results on the data plane using RIPE Atlas [16] traceroutes; showing topological effects of ROAs, BGP path hunting, and latency shifts.

We make the following contributions:

A method to measure the latency induced by RPKI adoption: We design an end-to-end experiment, for each of the five RIRs, to track the delay across the different steps between the creation/deletion of a ROA by the resource holder and the time in which we see the corresponding changes on the management, control, and data planes (§ 3). We deploy two experiments, one with an AS connected

mainly to ASes performing ROV (§ 4), and another one with ASes surrounded by some, but not all, ASes performing ROV to generalize our findings (§ 5.1).

A landscape of the impact of ROV adoption on the Internet: With these experiments, we found that: (1) There was a significant time disparity across RIRs between the operator's input and the ROA publication delay (Table 2 and 3). (2) This observation allowed us to discover some startling anomalies (since corrected after our notification) at ARIN and LACNIC that delayed their ROA publication time by up to five hours (§ 4.1). (3) There is an important disparity in ISPs' reaction time between ROA creation and ROA deletion, ranging from minutes to an hour. ISPs take significantly more time to act on ROA deletion than ROA creation (§ 4.2); (4) We also reported anomalous behavior to a Tier1 network which was quickly corrected. (5) There are vast differences between the RIRs' administrative practices seriously complicating the experiment setup, and highlighting how difficult it can be for operators to streamline their RPKI management procedures at the different RIRs (§ 6).

Extending the findings with a longitudinal study: We further broaden our study with an analysis of historical RPKI and BGP data (§ 5.2) showing that the bugs reported to RIRs have been present for years and that long delays of ROA creation have been quite stable over the past four years.

Inter-RIR differences in ROA payloads: Our analysis of RPKI data also reveals ROA structural differences between the five RIRs, highlighting RIRs' different management of RPKI data and explaining some of their disparities (§ 5.3).

2 Background

RPKI prefix allocation follows the IANA allocation hierarchy, and each RIR maintains a separate trust anchor (TA) for the resources for which they are responsible. Certificates are issued to their members, which are then used to sign ROAs. Each RIR operates a public repository in which all RPKI objects (certs, ROAs, CRLs, manifest files [3, 17]) are stored.

Fig. 1 depicts the steps performed when a resource holder queries an RIR to update RPKI information for its prefixes. Then the changes are fetched by operators performing Route Origin Validation (ROV-enabled ASes, green in Fig. 1) that use this new information to update their routers. Each step described below is common to all RIRs and ROV-enabled ASes, but each may perform these steps at different time intervals and frequency.

ROV-enabled ASes check route validity based on the information contained in ROAs. To get ROA information, routers need to connect to Relying Party (RP) software which is in charge of fetching ROAs, cryptographically validating their content, and feeding routers with Validated ROA Payloads (VRPs). Based on the VRPs, routers can then classify BGP announcements either as Valid, NotFound, or Invalid. ROV-enabled ASes typically drop the "Invalid" announcements.

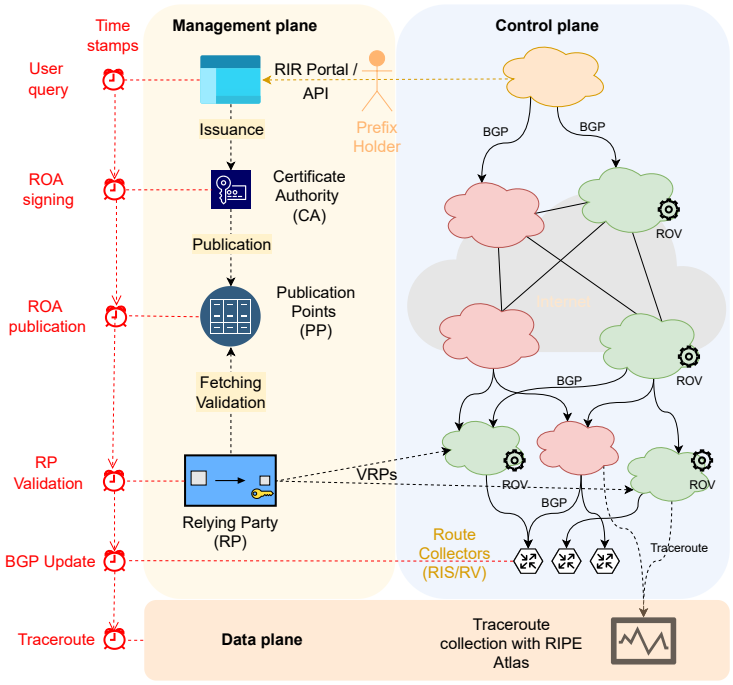


Fig. 1: Data-flow from creation of a ROA by the prefix holder to the corresponding BGP updates recorded at the route collectors (RIS / RouteViews). The red labels on the left show the points at which time measurements were taken.

Each step in the provisioning process introduces delay. The aim of this study is to track and quantify these delays across RIRs and some ISPs. For this we collect timestamps at the following points:

1. ***User Query***: The most common way for resource holders to create ROAs is to query the RIR that provided the IP prefixes. The queries are either via the RIR's web portal or the RIR's REST API if available.
2. ***ROA Signing***: RIRs collect user queries, verify that they are legitimate, and pass them to certification authority software which computes ROAs and corresponding metadata information (i.e., manifest and CRL files) and creates new signed files.
3. ***ROA Publication***: Then RIRs place new ROAs and metadata files into public repositories, called Publication Points (PPs), so that Relying Party (RP) software can fetch them when desired. This seems a simple step, but RIRs must ensure that RPKI objects are consistent at all times, hence metadata files and their corresponding ROAs must be atomically published.
4. ***Relying Party (RP) Validation***: RPs are deployed by ROV-enabled ASes and their role is to periodically fetch and validate all the objects from the global RPKI repositories. After validation, they produce a list of Validated

Table 1: Summary for the two experiments presented in Section 4 and 5.1.

Section	RIR	Origin AS	Upstreams	Period
§4	All five RIRs	3970	ROV-enabled	Nov. 2011 to Oct. 2022
§5.1	RIPE	17660, 55722, 23676	mix	May 2022 to Oct. 2022

ROA Payloads (VRPs) which routers use to verify incoming BGP announcements. Larger ISPs often deploy multiple RPs to avoid single points of failure, and to tune the timing with which each RP visits the Publication Points.

5. ***BGP Update***: ROV-enabled routers accept and advertise the *Valid* and *NotFound* announcements only and drop the *Invalid* ones based on the VRPs from the RPs. These changes propagate globally in BGP and the effects may be seen in BGP collection systems (e.g., RIS or RouteViews [15, 18]).
6. ***Traceroute***: These routing changes are reflected in the data plane and can be observed with measurement platforms such as RIPE Atlas [16].

3 RPKI beacons

To measure the propagation time of RPKI data from RIRs' Certification Authorities to BGP speaking routers we automated RPKI ROA beaconing at each of the five RIRs. Each beacon is a prefix for which we switch its RPKI status daily by creating and deleting ROAs. We announce these experimental prefixes in BGP from a few locations on the global Internet and measure the beacons' effects in the management, control, and data planes.

3.1 Beacon Methodology

We perform two experiments from diverse ASes to measure the propagation time of RPKI data (Table 1): For the first experiment (§4), we obtained from each RIR a pair of IPv4 /24 prefixes and a pair of IPv6 /48 prefixes[6]. One prefix from each pair of prefixes is used as a control, while the other is the test prefix. The control prefixes are expected to be always reachable, with an always valid RPKI status. If they are not reachable then we know that the experiment is not valid for that period. For the test prefixes, the BGP announcements do not change, but we periodically add and remove a ROA to alternatively validate and invalidate the origin AS of the test prefixes' BGP data. We track changes reflected at the management (RPKI), control (BGP), and data plane (traceroute). The prefixes are announced from AS3970 , which is directly connected to AS3130 (not implementing ROV), which in turn is connected to two ROV-enabled upstream providers, NTT (AS2914) and Sprint (AS1239), and peering with a ROV-enabled route server at a large IXP, and directly with a few non-ROV IXP peers. The results for this experiment are described in Section 4.

[6] The list of all prefixes is given in appendix, Table 6

For the second experiment (§5.1), we used three /24 prefixes (RIPE-A, RIPE-B, and RIPE-C) from the RIPE NCC and announced them from three diverse networks, including an IXP and a national ISP with 149 peer ASes. The three prefixes are used as test prefixes, meaning that we daily alternate the ROA status for all of them. The results of this experiment are described in Section 5.1.

3.2 ROA toggling

In order to measure only the impact of ROV and avoid delays caused by other filtering mechanisms, we configured all filtering with the upstream providers (e.g., through the creation of Internet Routing Registry (IRR) route objects). We verified that our providers' filters accepted our prefixes and then left these mechanisms untouched.

To toggle the RPKI status of the test prefixes, each was invalidated by Pre-registering a ROA with the origin AS set to the invalid AS 666. For the first experiment our AS was primarily connected to upstream networks and IXP route servers that implement ROV, thus at the initial step, our test prefixes are dropped by ROV-mechanisms and globally unreachable, as opposed to the second experiment.

The ROA toggling consists of daily repeating the following steps for each test prefix:

1. ROA creation. At a random time between 00:00 and 06:00 UTC, we request a new ROA covering the <prefix, AS> to authorize the route to the test prefix.
2. Convergence phase 1. From 06:00 to 12:00 UTC we give sufficient time for networks to obtain the new ROA, process it, and update their routing.
3. ROA deletion. At a random time between 12:00 and 18:00 UTC, we delete the ROA created at the first step, hence letting our test prefix fall back to Invalid.
4. Convergence phase 2. From 18:00 to 00:00 UTC we again wait for all networks to converge to the new state.

In order to keep the RPKI beacons running over a long time, we automated all queries to RIRs. ARIN, RIPE, and recently LACNIC, provide APIs to ease such interactions with their services. We made all queries to these three RIRs via their APIs. AFRINIC and APNIC have no APIs for RPKI management; we could only create and delete ROAs via their web portals. To automate AFRINIC and APNIC processes we implemented Selenium [19] scripts that log in to these portals and submit web forms for ROA creation and deletion.

3.3 Data collection

In order to measure the time for the above RPKI operations to propagate over the management, control, and data planes we collect temporal information from ROAs' payload, BGP data, and run traceroutes.

User query Delays are measured relative to the user query time, that is the time we request the RIRs to change RPKI (steps 1 and 3 in Section 3.2). This is logged by an NTP-synchronized host that automates the queries for ROA creation and deletion. We log the precise time of the confirmation from the RIR portal or API that the query was received without error.

RIR We infer RIRs' signing and publication delays from the RPKIviews archive [20]. This archive consists of RPKI data snapshots taken every 20 minutes. Each snapshot contains the raw ROA files of all RPKI repositories as well as the output of a relying party software, rpki-client [21]. From this dataset, we compute the signing, publication, and RP delay (Fig.1).

The signing delay is computed using the signing timestamp found in the ROA, more specifically in the Cryptographic Message Syntax (CMS) [22] wrapper of the signed object. As opposed to the "NotBefore" timestamp found in the ROA payload, which is used to determine at what time a ROA becomes "valid", the signing timestamp conveys the time at which the Certification Authority created the ROA. Unfortunately, the reliability of both timestamps are disputable as our results show that some RIRs set the signing and/or NotBefore timestamps arbitrarily (Section 4.1 and 5.3)!

The publication delay estimates the delay for an RIR to make newly created ROAs available to RPs. We infer the typical publication delay from RPKIviews snapshots. Since RPKIviews takes snapshots every 20 minutes and assuming that the publication of ROA is uniformly distributed over time, new ROAs appear in RPKIviews on average 10 minutes after their actual public availability. For ease of discussion, when reporting RPKIviews median delay publication time in Section 4, we subtract 10 minutes from the measured RPKIviews median delay. We analyze only these corrected median values, not individual delays.

Relying Party (RP) Computing Relying Party delays on the Internet is particularly challenging. The delay of RPs depends on three factors: the frequency at which they poll for new data from publication points, the downloading time, and the ROA processing time (i.e., mostly reading and decrypting files). Network operators may increase their RPs' polling frequency to fetch new data more quickly, but to reduce the burden on publication points, the recommendations are to poll for new data no more frequently than 10 minutes using RRDP (or as low as 1 minute if there is caching infrastructure and the If-Modified-Since header value is set) and not more than every 30 minutes if using rsync [23]. Furthermore, past studies showed that 2 and 10 minutes are the most common RP polling frequencies [24] which correspond to respectively RIPE v3 validator and Routinator default values (rpki-client has no default value). As RIPE v3 validator has since been deprecated, we assume that 10 minutes is now a common value used by operators and attempt to estimate RP delay for RPs polling new data every 10 minutes.

Similarly to the publication delay, we leverage RPKIviews data to infer the typical delay experienced by an RP polling data every 10 minutes. Because the 20-minute frequency of RPKIviews translates into a 10-minute median polling

delay and a 10-minute polling frequency gives a 5 minute median polling delay, when reporting results in Section 4 we correct the RP delay by subtracting 5 minutes from the median delay observed with RPKIviews' RP.

BGP The ROA toggle described above affects the global reachability of our announced prefixes. They become unreachable when corresponding ROAs are deleted and reachable again when ROAs are re-created. We monitor these shifts in BGP using the RIPE Routing Information Service (RIS) data [15]. We particularly look into the BGP update messages sent from routers peering with RIS, and we record for each peer and each test prefix the time of the first announcement after creating a ROA and the time of the first withdrawal after deleting a ROA (BGP update in Fig.1). These represent the first routing changes caused by each of our RPKI beacon events that we expect to be visible at the collector, and are accurate within seconds.

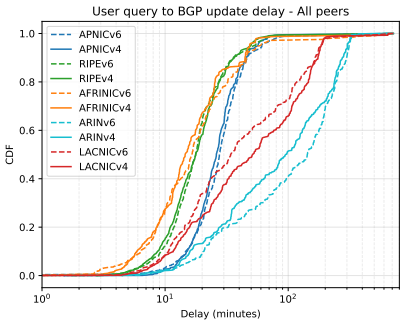
Section 4 presents delays for the RIS collectors RRC00 and RRC01. RRC00 has the advantage of being a multi-hop collector, meaning that it receives data from ASes that are located in very diverse locations. RRC01 collects data only from ASes peering at the LINX IXP which includes both upstream providers for the first experiment. Hence RRC01 allows us to investigate BGP signals from networks that make our prefixes globally reachable. In our preliminary analysis we have looked at an arbitrary set of RIS collectors (RRC03, RRC06, RRC12), but given the large amount of data, and that we see little difference across collectors, we present results only for RRC00 and RRC01. Past research has also shown a high level of redundancy between different collectors [25] which limit the benefits of using numerous collectors [26].

Traceroute To test data plane reachability and delay of the prefixes with toggling ROAs, we performed traceroutes every 15 minutes from RIPE Atlas with probes in 6 different ASes. The probes were chosen to be inside the ASes that also share BGP routes with RIPE RIS at RRC00. We pick these ASes to have close vantage points for BGP and traceroutes, but there is no guarantee that the Atlas probe and the BGP collector share the same routes, so there could be some mismatch. However, we also tried a wider set of RIPE Atlas probes using Atlas geo-diverse selection of probes and observed similar behaviors, so our analysis focuses only on the traceroutes obtained with the 6 probes mentioned earlier. The measurements are public (Table 7) and traceroutes are configured to send three ICMP packets per hop.

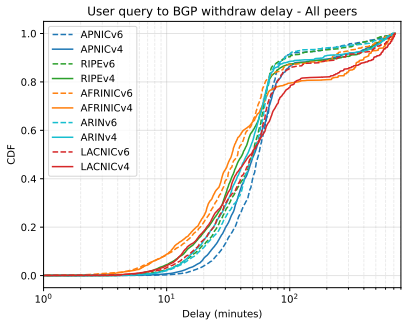
4 Eleven months in the life of RPKI beacons

We now present the results of our first experiment; over eleven months of toggling RPKI ROA beacons for prefixes from the five RIRs and announced from an AS surrounded by ROV-enabled networks (see row 1 in Table 1).

The analysis in this section follows the steps shown in Fig. 1 and is based on RIS data (RRC00 and RRC01) from November 1st 2021 until October 5th 2022,

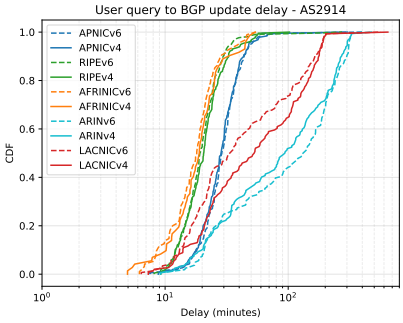


(a) ROA creation.

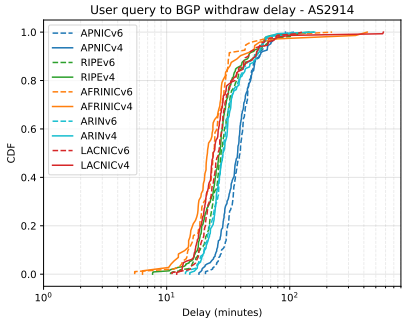


(b) ROA deletion.

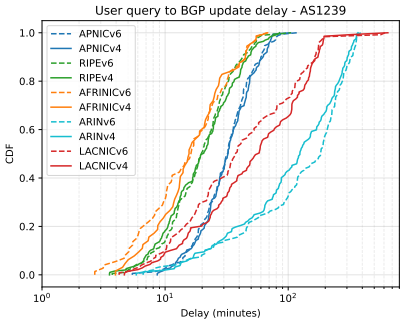
Fig. 2: Time from user query to propagation in BGP, for all RRC00 and RRC01 peers. ARIN and LACNIC had significantly longer creation delays due to a bug related to ROAs' NotBefore timestamps. APNIC delay is typically 10 minutes longer than AFRINIC and RIPE. Overall the delays for ROA deletion are higher than for ROA creation.



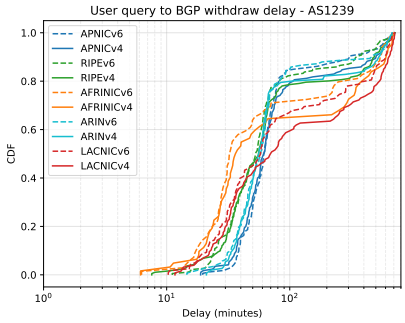
(a) ROA creation: NTT (AS2914).



(b) ROA deletion: NTT (AS2914).



(c) ROA creation: Sprint (AS1239).



(d) ROA deletion: Sprint (AS1239).

Fig. 3: Time from user query to BGP propagation (RRC00 and RRC01). Focus on the two upstream providers of our experimental AS: Sprint (AS1239) and NTT (AS2914). NTT had more consistent delays than Sprint, and Sprint had sometimes very long delays to withdraw prefixes with deleted ROAs.

except for the LACNIC beacons that started on February 1st 2022. We rely on RPKIviews data from January 1st to October 6th 2022.

Our main findings are that creation times vary significantly across RIRs, with medians ranging from a few minutes to over an hour for new ROAs to reach the publication points. The differences lie in the way ROAs are processed by RIRs, in batches at specific times of the day, and drastic issues we discovered at two RIRs (each applied a temporary fix). Second, deletion of ROAs takes longer to reflect in BGP as routers explore alternate routes that have not yet been invalidated. The slowest element drives the deletion time. Routers with a slow pulling cache, or redundant caches, invalidate routes late and are used by neighbors to reach the invalidated resource. Further, for ROA creation, most of the delay comes from the Relying Party pulling objects at different intervals.

4.1 ROA creation delay

We investigate ROA creation delay. We explore the disparity across RIRs and between upstream ASes. Fig. 2a shows the per-RIR distribution of the delay between the query time to create a ROA for our test prefix and the time when reachability is first reported by each RIS peer in BGP.

AFRINIC and RIPE prefixes are seen most quickly in RIS. The median delay for an AFRINIC IPv4 prefix is 15 minutes (16 minutes for IPv6) and 18 minutes for RIPE prefixes for both IPv4 and IPv6. APNIC is consistently slower than AFRINIC and RIPE. The median delay for an APNIC IPv4 prefix is 26 minutes (28 minutes for IPv6). This 10-minute extra delay is due to a 20-minute batching process at APNIC (see Section 5.3). ARIN and LACNIC prefixes are susceptible to significant delays. These are due to the timezone problem described below (Publication delay and ARIN/LACNIC timezone issues). We have reported this issue to both RIRs for which ARIN deployed a workaround on 21 April 2022 and LACNIC on 12 October 2022.

For the other three RIRs delays are less than 1 hour in at least 95% of the cases. We also observe some outlying values: In less than 3% of the cases, for AFRINIC, APNIC, and RIPE, the BGP delays go over 100 minutes. These delays are rarely visible from the two upstream providers, NTT and Sprint (Fig. 3a and 3c). Both always announce the AFRINIC prefix in less than 100 minutes. We cannot find consistent behaviors for the observed long delays, these could be due to unexpectedly long BGP convergence times [27]. In addition, we noticed that about half of them are related to very small ASes owned by individuals (network operators) who are active in testing new deployments (e.g., AS15562, AS35619, AS5662) so these could be the results of experiments.

We observe a large disparity across RIRs in the time elapsed between ROA creation and the effect in BGP. The same is true across our upstream ASes. In the next sections, we track the time along the different steps in Fig. 1 to understand the elements causing these disparities. We rely on Table 2, where we show the median delay for each of the steps in Fig. 1.

Table 2: **ROA Creation Median Delays**. Median delay in minutes from the user query to the step indicated in each column as observed for the IPv4 prefixes from the five RIRs (IPv6 results are in parenthesis). As described in § 3, delays shown in this table are either measured from ROA attributes (*) and BGP data (‡), or inferred from RPKIviews data (†).

	Sign*	NotBefore*	Publication†	Relying Party†	BGP‡
AFRINIC	0 (0)	0 (0)	3 (2)	14 (13)	15 (16)
APNIC	10 (13)	10 (13)	14 (16)	34 (38)	26 (28)
ARIN	- (-)	- (-)	69 (97)	81 (109)	95 (143)
LACNIC	0 (0)	- (-)	54 (32)	66 (42)	51 (34)
RIPE	0 (0)	0 (0)	4 (4)	14 (13)	18 (18)
After fix:					
ARIN	- (-)	- (-)	8 (9)	21 (22)	28 (23)

Certification delay. According to the ROA signing time, AFRINIC, RIPE, and LACNIC create ROAs within a minute of receiving users' queries. APNIC's 10-minute delay appears at this very first step. APNIC's signing times for our ROAs are in 20 minutes increments (e.g. 04:30, 04:50, 05:10) suggesting that APNIC is processing users' queries in 20-minute batches, adding an average delay of 10 minutes.

We found that ARIN hard-codes both the ROAs signing time and NotBefore value to midnight UTC hence we are not able to compute signing delays for ARIN. LACNIC is signing ROAs immediately after the user query, but the NotBefore value in LACNIC ROAs is also hard-coded to midnight UTC.

Publication delay and ARIN/LACNIC timezone issues. Before April 2022, the publication delay for ARIN and LACNIC could last several hours due to a time zone conversion problem. As mentioned above both RIRs intend to set NotBefore values to midnight, but instead, ARIN has been setting this value to 04:00 UTC or 05:00 UTC (corresponding respectively to 00:00 in Eastern Daylight Time and Eastern Standard Time) and LACNIC has been setting this value to 03:00 UTC (corresponding to 00:00 in Uruguay Standard Time). For example, a query at 01:00 UTC to create a ROA in LACNIC would create a ROA with a NotBefore value set to 03:00 UTC. Therefore, the ROA would be invalid for the two hours following its creation. Our experiment reveals that the Publication Point wisely does not publish the "not-yet-valid" ROA to the repository hence delaying its availability to RPs. The same holds for ARIN. We reported this issue to both ARIN and LACNIC.

ARIN acknowledged the problem has been present since they started their RPKI service. An interim fix for this issue was deployed on 21 April 2022, by setting the signing and NotBefore timestamps at 12:00 UTC on the day before the user query. ARIN is planning further development to properly solve this issue.

Since the ARIN issue has been addressed, the publication delays for ARIN are in line with RIPE and AFRINIC at around 5 minutes median delay ("After fix" line in Table 2).

For LACNIC the issue is apparently only affecting ROAs created with their API, not the ones created manually on their portal. LACNIC deployed a similar fix on October 12, 2022, that sets the NotBefore timestamp at 03:00 UTC on the day before the user query. Since our LACNIC prefixes were returned on October 25, 2022, we observed the effects of this fix for less than two weeks and found that LACNIC publication delay fell in line with the other RIRs' delay (not included in Table 2 due to the small sample size). As these issues bias results for LACNIC and ARIN, the remainder of this section focuses on results from the other RIRs.

Relying Party delay. Propagation to Relying Parties (RPs) represents the most time-consuming step observed in ROA processing. Unlike other steps where data are pushed to the next component, RPs periodically pull RPKI data from Publication Points. The delay we observe between the ROA creation and the time when an RP validates the new ROA is usually less than 15 minutes (38 minutes for APNIC). This is 10 minutes more than the publication delay and consists mainly of the polling interval (5 minutes delay on average), downloading time from all Certification Authorities (4 minutes), and the ROA processing time (1 minute). The downloading time can be negatively impacted by Publication Points that are responding slowly. Single-threaded RPs, such as the one used by RPKIviews (rpki-client), are particularly affected by this as they sequentially visit all Publication Points and may be blocking on slow Publication Points.

BGP updates. We usually observe BGP updates for the newly created ROAs about 3 minutes after the estimated RP validation time. This delay includes both the router's polling from RPs and BGP propagation time, as we are not able to measure the RP to router delay alone. As RPs signal routers when to pull, this delay should be dominated by the data transfer and the router processing of VRPs. Past work on BGP propagation estimate that a new announcement on BGP takes usually less than a minute to propagate globally [5, 6], hence one can estimate the RP to routers delay should be no more than 2 minutes.

To further dissect delays observed at this step, we compute the BGP delay only for the two upstream providers. The BGP delay distributions of NTT (Fig. 3a) and Sprint (Fig. 3c) are similar to those observed for other peers (Fig. 2a), and their median values are all within a 4-minute difference. Given that ROV is still deployed very sparsely [28], these results show that (1) the delay for ASes that are not along ROV-enabled AS paths is dictated by our upstream providers, (2) ASes beyond our upstreams that perform ROV slower would invalidate new routes. In the latter case, because we are connected to Tier1 networks, and there are many paths between Tier1 networks and RIS collectors, the effect of other ROV deployments is rarely observed.

We also compared these distributions with five other networks that are implementing ROV and announcing our prefixes to RRC00 or RRC01 (AS1299,

AS6939, AS7018, AS9002, AS14907) and found no notable differences in the distributions, meaning that these networks behave similarly to our upstreams; i.e., they are at least as fast as our upstreams to pull new RPKI data. With these data we cannot distinguish if they can fetch RPKI data faster than our upstreams as their BGP announcements are bound by the time our upstreams made the prefixes globally available.We come back to this in Section 5.1 with experiments announcing prefixes from very diverse locations.

A careful inspection of the delays for our two upstreams reveals that NTT is more consistent, the 10th to 90th percentile range for the IPv4 RIPE prefix corresponds to 12 and 32 minutes (Fig. 3a) whereas these same percentiles correspond to a range twice as large for Sprint, i.e., 7 and 47 minutes (Fig. 3c). Although the first quartile delay for Sprint is always better than for NTT (e.g. 12 minutes vs 15 minutes for RIPEv4), the third quartile delay for Sprint is consistently longer by 1 to 10 minutes for the AFRINIC, APNIC, and RIPE prefixes. We believe this is the result of a longer RP polling frequency for Sprint but a shorter RP to router delay, and we confirmed with network operators that indeed NTT is polling RPKI data more frequently than Sprint and Sprint is using faster RP software.

Using only the data after ARIN's fix we confirm the delay for ARIN prefixes improved significantly (shown in Appendix Fig. 11). The interquartile range corresponds to 13 and 33 minutes for IPv4 (13 and 33 for IPv6) which comes very close to RIPE and AFRINIC results for the same time period. RIPE's interquartile is 11 to 32 minutes for IPv4 (11 to 27 for IPv6) and AFRINIC's interquartile is 10 to 25 minutes for IPv4 (9 to 29 for IPv6) between 21 April 21st and May 15th 2022.

Data plane availability. Fig. 4 shows how prefix reachability/unreachability on the data plane for IPv4 (IPv6 in Appendix, Fig. 10) is affected by ROA creation and deletion. Each row of these graphs shows a sequence of traceroutes for a different Atlas probe/prefix pair. A pack of 6 rows shows the traceroutes to the same destination, from 6 diverse RIPE Atlas probes, indexed from top (#1) to bottom (#6). The colors of the dots show whether the destination is reachable (cyan) or unreachable (black). We add to this graph the user query times for ROA creation (green dots) and deletion (red dots).

At ROA creation, the delay between the user query and data plane reachability is similar to BGP. This is represented in Fig. 4 by the time difference between a green dot and the first next cyan dot. We observe a median delay between 23 minutes (RIPE) and 50 minutes (APNIC). Given that traceroutes are run every 15 minutes, these delays include on average an additional 7.5 minutes delay from Atlas, hence we estimate the median data plane delay in our experiments to range between 15 and 43 minutes which is in line with the median delays observed in BGP (Table 2).

4.2 End to end ROA deletion delay

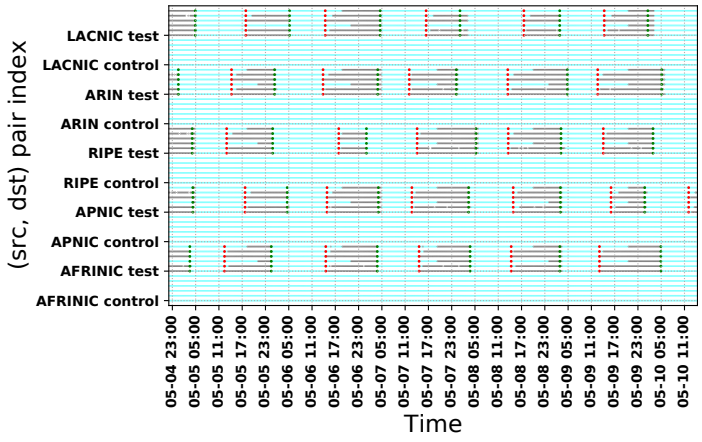


Fig. 4: Effects of ROA creation (green dots) and ROA deletion (red dots) on prefix reachability (cyan dot) and unreachability (black dot) in traceroute. Each line shows a different Atlas probe/prefix pair. Delay between ROA deletion and unreachability highly varies depending on the topology. IPv4 only, see Fig. 10 for IPv6.

We investigate the ROA revocation timing along the steps in Fig. 1. In addition to longer deletion than creation due to path exploration, we show that while APNIC demonstrates longer times for the revocation to be published and to reach Relying parties, the prefixes disappear from BGP only slightly after the prefixes with ROAs hosted by other RIRs.

Certification, Publication, and Relying Party delay. At ROA deletion, the delays from the management plane to the RP are the same as those observed at ROA creation. The timestamps that appear in Certificate Revocation List (CRL) files usually match our user query time, and RP delays are similar across all RIRs, with the exception of APNIC which still lags 10 minutes behind other RIRs (Table 3).

BGP withdraw. BGP delays are significantly higher for ROA deletion than for ROA creation (Fig. 2b). The median BGP delay for unreachability goes up to 51 minutes for IPv4 and 56 minutes for IPv6 (Table 3). We rarely observe short BGP delays (Fig. 2b). At best the BGP delay first quartile corresponds to less than 20 minutes (AFRINICv4) and at worst less than 39 minutes (APNICv6).

There are two related causes for these high delays, one is related to BGP and the other to RPs/routers interactions. At ROA creation a prefix is announced globally in BGP by one of the prefix's upstreams as soon as either one of them fetches the new ROA. But at ROA deletion neighbors must all withdraw the

Table 3: ROA Deletion. Median delay in minutes from user query to the step indicated in each column as observed for the IPv4 prefixes from the five RIRs (IPv6 results in parenthesis). These delays are either measured from CRL files (*) and BGP data (‡), or estimated from RPKIviews data (†).

	Revocation*	Relying Party†	BGP‡
AFRINIC	0 (0)	13 (14)	34 (38)
APNIC	10 (12)	31 (36)	51 (56)
ARIN	0 (0)	14 (16)	45 (51)
LACNIC	0 (0)	18 (20)	48 (49)
RIPE	0 (0)	14 (13)	41 (50)

ROA to make it globally unreachable. Similarly, for reliability via redundancy, the RPKI-to-Router Protocol [4] allows a router to receive data from multiple Relying Party caches. This makes ASes using multiple RP caches likely to react significantly more slowly to ROA deletion than to ROA creation. This is because the BGP prefix is valid if there is a matching ROA from any of the caches. So ROA deletion is not effective until the last cache withdraws. Conversely, the first cache to receive a new ROA validates the BGP prefix, so ROA creation is seen relatively quickly.

The effect of multi-RP setups is evident for 3970's two upstream networks. Both have longer delays for ROA deletion than creation. Sprint also frequently experiences very high delays (greater than 100 minutes). We privately contacted the operators, and they confirmed that this delay is likely due to a reported bug in the Routinator Relying Party implementation sometimes not withdrawing ROAs (which was recently addressed in Routinator version, 0.11.2 [29]). Sprint is deploying the fix for this issue. These long delays are propagated to certain RIS peers, especially for the AFRINIC and LACNIC prefixes (Fig. 2b). This illustrates the effect caused by BGP, as only one delayed upstream kept the prefixes globally reachable for a longer period of time. Not all RIS peers are impacted though. ASes that implement ROV, or that are surrounded by ROV-enabled networks, may drop the prefix before Sprint, which is for example the case for NTT (Fig.3b). But RIS peers that are not implementing ROV and reaching our test prefixes via Sprint are surely affected by the high Sprint delay. A mixture of both can even be observed. A good example is Deutsche Telekom AS3320 (see Fig. 5), which is highly impacted in IPv4, but not in IPv6, as the BGP paths show, it reaches the IPv6 prefixes only through NTT or through Hurricane Electric via the IXP route server, never through Sprint.

Data plane unreachability. Results from traceroute provide additional insight into slow withdrawals. It is reflected in Fig. 4 by a large gap between a red dot (ROA deletion) and the next black dot, indicating a path still active after the deletion. Probe#1 and probe#4 have longer delays, on the order of hours,

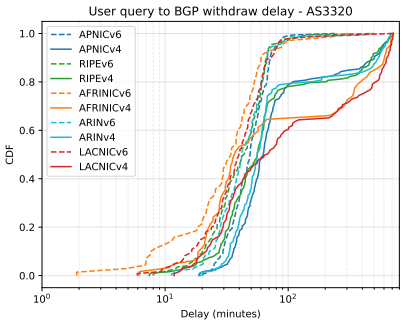


Fig. 5: ROA deletion. Time from user query to BGP withdraw for Deutsche Telekom (AS3320). IPv4 delays are impacted by Sprint late withdrawing.

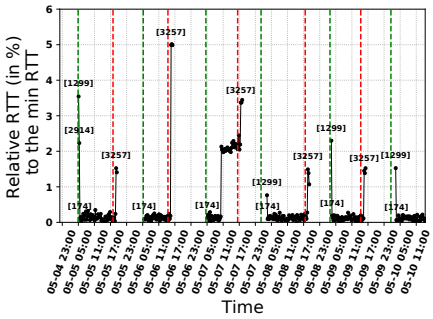
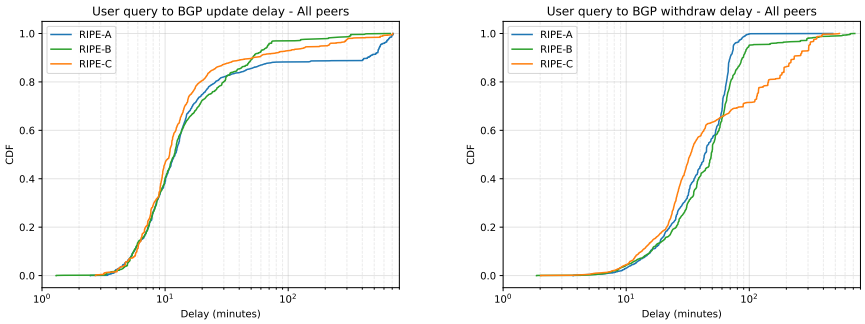


Fig. 6: Effects of ROA creation/deletion on the data plane. After each ROA creation or deletion, we observe BGP path hunting with AS path changes.

for all test prefixes. For probe#1, we observe that after ROA deletion, the AS path between the RIPE Atlas probe and the destination changes from [Source AS, AS174 (Cogent), AS1239 (Sprint), Destination AS] to [Source AS, AS6762 (Telecom Italia), AS1239 (Sprint), Destination AS] before becoming unreachable. Since Telecom Italia is not performing ROV [30], our hypothesis is that Cogent fetches RPKI data faster than Sprint and drops the prefix while Sprint is still announcing it. BGP path hunting then selects an alternate path via Telecom Italia, until finally Sprint also drops the route and the prefix becomes unreachable. For probe#4, the delay before unreachability is similar to probe#1, but we do not observe an AS path change between ROA deletion and destination being unreachable, the AS path remaining [Source AS, AS7575 (AARNET), AS6461 (Zayo), AS1239 (Sprint), Destination AS]. Again, our hypothesis is that Sprint is slow to drop this route and keeps announcing the route to Zayo, which does not perform ROV [30], so it announces the prefix until Sprint drops it.

Impact on AS path Fig. 6 shows the impact of ROA creation and deletion on the observed paths and illustrates BGP path hunting for one of the Atlas probe/prefix pairs. The Y axis represents the latency between the Atlas probe and the destination relative to the minimum RTT observed during the measurement period, and the X axis represents time. The vertical lines show the times of ROA creation/deletion. Each dot is a traceroute, and every time the AS path changes, we put a label above the dot with the new AS seen in paths taken by the traceroute packets.

BGP convergence and path hunting are each illustrated after ROA creation and deletion. After ROA creation, we observe a first path going through AS1299 (Telia), and then a preferred path (in the sense of BGP) going through AS174 (Cogent) is selected. This suggests that Telia was faster to integrate the new ROA than Cogent. After ROA deletion, we observe that BGP finds another path



(a) ROA creation seen from RRC00 and RRC01 peers.

(b) ROA deletion seen from RRC00 and RRC01 peers.

Fig. 7: Time from user query to BGP propagation for prefixes RIPE-A, RIPE-B, and RIPE-C as observed by RRC00 and RRC01 peers.

going through AS3257 (GTT), and then the destination becomes unreachable, as we see the dots stopping a short time after the red lines.

The latency shift observed at 05-07 10:05 AM is due to an intradomain routing change within Sprint with two hops instead of one, likely not related to ROA creation/deletion, as it only appears once during our measurement time and not close to any ROA event.

5 A bird's-eye view of RPKI ROA delay

The above experiment measures delays introduced by RPKI in routing using resources from all five RIRs. We discovered different handling of ROA creation at RIRs as well as the effect of different timings at ISPs when pulling RPKI data. The latter was made possible because all providers of our vantage point perform ROV. Next, we conduct a second experiment and investigate other datasets in order to generalize some of our findings.

Tier1 networks usually react to new ROAs within 20 minutes after the user's ROA creation query. They drop prefixes for deleted ROA within 40 minutes after a user's ROA deletion query; though we observe certain cases when they may take up to one hour. We discover the existence of Tier-1's that are faster than NTT and Sprint to react to ROA creation. Since the upstreams of our new prefix origins do not all perform ROV, we observe BGP collection points and traceroute vantage points with continuous connectivity to the prefixes despite the invalidation of origin ASes. Hence we show how ROV complicates the routing information propagation process and how difficult it is to predict ROV timing, especially for prefixes originated by networks that have rich and diverse connectivity. Using longitudinal datasets, we also confirm that observed delays have been stable over the past four years, and we reveal ROA structural differ-

ences between the five RIRs, highlighting RIRs' different management of RPKI information and explaining some of their disparities.

5.1 Topology dependence

The results of Section 4 are constrained by the location of our originating AS in the Internet, and in particular by the way its upstream networks handle RPKI. For example, at ROA creation, the time it takes for the prefixes to become globally reachable in BGP is bounded by the reaction time of NTT and Sprint. In this section, we show that these results are representative not only for the numerous networks relying on these two Tier1 networks, but also for networks relying on other large Internet providers.

More locations. For this stage we obtained three /24 IPv4 prefixes (RIPE-A, RIPE-B, and RIPE-C) from RIPE NCC and three topologically diverse operators generously agreed to announce these prefixes from their networks. These three networks differ significantly from our experimental AS in the first setup; the locations are on a different continent and have different upstream providers including networks that do not implement ROV. Therefore, when running RPKI beacons for these prefixes their reachability is unaffected along paths that have no network implementing ROV. Only RIS peers that implement ROV or that are surrounded by ROV lose reachability to these prefixes.

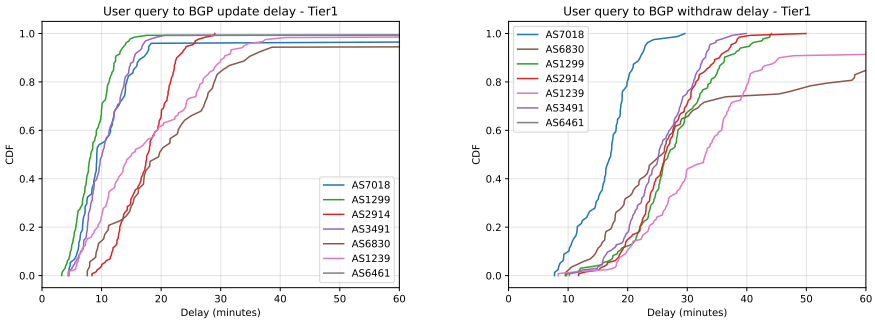
We measured these prefixes from May 6th to October 5th 2022, and again observe that BGP delay for ROA deletion is significantly longer than it is for ROA creation (Fig. 7).

The median BGP delay for ROA creation is shorter than during the first experiment, the median ranging between 11 and 12 minutes (Fig. 7a) compared to the median of 18 minutes observed previously for our IPv4 RIPE prefix, suggesting that ROV-enabled networks between these origin ASes and RIS peers are faster than NTT and Sprint in the previous experiment (see Section 5.1).

On data plane reachability, we observe the expected behavior that here some probes never lose reachability, because they find a route via a provider that does not enforce ROV, as opposed to the probes in the first experiment (Fig. 4).

ROV by Tier1. We leverage this experiment to measure the BGP delays of Tier1's that peer with RIS and implement ROV. Since these three prefixes are announced in places that are not entirely surrounded by networks performing ROV we assume the prefixes remain continuously reachable by a large fraction of the Internet. Hence, well-connected networks, i.e., Tier1s, are likely to adopt new paths for these prefixes based on their ROV mechanisms, not owing to a change in BGP reachability.

Starting with a list of Tier1 networks (CAIDA's peering clique of ASes [31]), we select six networks that are peering with RIS (RRC00 and RRC01) and that are known to implement ROV [28]. Fig. 8 shows the measured BGP delay for these six networks. Comparing the delays at ROA creation with the RIPE IPv4



(a) ROA creation: ROV-enabled Tier1. (b) ROA deletion: ROV-enabled Tier1.

Fig. 8: User query to BGP delay for prefixes RIPE-A, RIPE-B, and RIPE-C as observed by Tier1 networks implementing ROV.

results of the previous experiment (Fig. 3a) confirms the stable delays for NTT and the higher variability of Sprint (AS1239) for these additional prefixes.

We also notice that NTT (AS2914) is consistently 5 to 10 minutes slower than AT&T (AS7018), Telia (AS1299), and PCCW (AS3491). This suggests that these networks fetch RPKI data more frequently than NTT, which we confirmed with operators of two of these networks. This difference may also explain the 7 to 8 minute difference between the median delay for the RIPE prefix in the previous experiment (Fig. 2a) and the three prefixes used here (Fig. 7a).

The delay at ROA deletion is higher than at ROA creation for all monitored networks as seen in Fig. 8b. As we expect these networks to drop the prefixes as soon as they get in sync with RPKI, the slower deletion of ROAs is the result of RP redundancy, i.e. the ROA deletion is not effective until the last cache withdraws the ROA (Section 4.2). The anomaly in Sprint ROA deletion was confirmed to be due to the Routinator bug.

5.2 Delay analysis from historical data

Are these results consistent over time? We investigate historical ROA and BGP data and compute the delay between the *BGP withdrawal* time t_1, of an RPKI-invalid prefix and the NotBefore time of the ROA t_0 that invalidates the <prefix, origin> pair. In this experiment, we track the occurence of BGP Withdraw (W) messages instead of BGP Announcements (A), as we cannot affirmatively say whether a ROA creation triggered an update (A). BGP updates (A) can happen both when the routes are tagged as "RPKI-valid" or "RPKI-notfound". However, withdrawals (W) following the creation of ROAs are more likely due to the routes being tagged as "RPKI-Invalid" and being dropped by ROV-enabled ASes.

Furthermore, as opposed to our active measurements, we do not have access to the user ROA query time and hence we rely on the NotBefore time as a proxy. The NotBefore time indicates when a ROA becomes valid and therefore

Table 4: Data processed for historical analysis (IPv6 in parentheses)

Date	# RIB entries	# VRPs	# invalids	# withdrawals
2018-05-22	786361	52421	17435 (1079)	1344 (-)
2019-05-01	853149	83221	20854 (1467)	2765 (86)
2020-05-15	923715	149075	21689 (2624)	2827 (351)
2021-05-02	1010201	247858	28203 (2764)	3837 (1751)
2022-05-13	1078454	342199	34604 (4688)	5918 (4191)

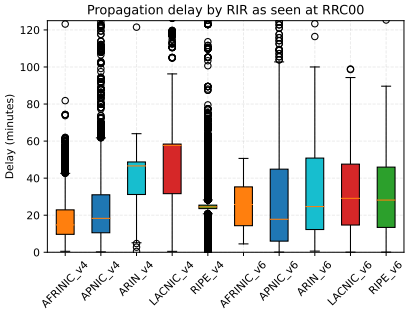
actionable for ROV. A quick analysis of the current RPKI repository shows that 77% of ROAs have a signing time equal to their NotBefore time, except for ARIN and LACNIC where the NotBefore time is not reliable as explained in section 4.1. We observe that for AFRINIC, RIPE and APNIC, there is almost no time difference between the signing time and NotBefore time, except for a few exceptional cases with AFRINIC ($< 10\%$) where the NotBefore time is set before signing time. This provides confidence that the NotBefore time is usually a good estimator of the signing time for AFRINIC, RIPE and APNIC but not for ARIN and LACNIC. We also confirmed from our active measurements that the NotBefore time for RIPE and AFRINIC is usually within a minute of our query time and on average 10 minutes later for APNIC.

Below is the process to calculate BGP delay using historical data:

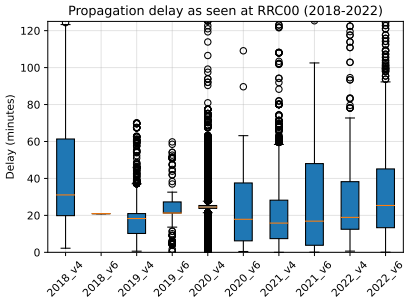
1. **VRP data:** We first collect a list of VRPs (Validated ROA Payloads) from the RIPE RPKI archive [32], which provides historical RPKI data organized by TA (Trust Anchor). Each repository contains the certificates and ROAs classified by date and also provides a list of VRPs for each day. We extract the NotBefore time (t_0) and route (prefix, origin) for each VRP.
2. **RIB files:** We select from RIS RRC00 collector a RIB dump on a randomly selected day in May every year from 2018 to 2022.
3. **BGP update messages:** we extract the BGP update messages from RIS update files and look for BGP withdrawals at time t_1 that correspond to a VRP's prefix and where t_1 is between t_0 and $t_0 + 1$ hour.
4. **BGP delay:** We calculate the BGP delay as $t_1 - t_0$.

Table 4 provides detail about the volume of longitudinal data processed from the RRC00 collector and from the RIPE RPKI archive. It shows the total number of RIB entries, the number of invalid routes and the corresponding number of withdrawals found in BGP data.

Fig. 9a shows an overview of the BGP delay for all data points collected between 2018 and 2022. There is no major difference in median propagation delay between IPv4 and IPv6, but there is greater variability in IPv6. We observe that AFRINIC, APNIC and RIPE had consistently shorter median delays over time while ARIN and LACNIC had higher delays for IPv4. The reason for higher delays for ARIN and LACNIC may be caused by the anomaly in the publication process (see 4.1). However, as we can see from Fig. 9b, the median delay remained usually around 20 minutes between 2019 and 2022. The numbers for 2018 are



(a) Median propagation delay across all RIRs (2018-2022).



(b) Median propagation delay excluding ARIN and LACNIC (2019-2022).

Fig. 9: Median propagation delay retrieved from historical data.

Table 5: Number of unique ROA objects, routes, and signing timestamps from a snapshot on December 31st 2021 of ROAs created in 2021.

	# ROA object	# route	# signing time
AFRINIC	134	207	134
APNIC	5213	74349	5162
ARIN	29213	31307	311
LACNIC	5071	16536	2484
RIPE	25691	145950	22279

slightly higher but overall these results suggest that the Certification Authority to BGP delays at ROA creation have been stable over the past four years.

5.3 ROA anatomy

Finally, this section describes the differences between the ROA payloads generated by the different RIRs and how these can impact ROA publication delay.

Signing time distribution The first notable difference between the ROA payloads of different RIRs is the distribution of signing and NotBefore timestamps. As mentioned in Section 4.1 we found that ARIN is using a hardcoded value for the signing and NotBefore timestamps. Looking at a snapshot of all ROAs on December 31st 2021, we found that the 29213 ROA objects that ARIN signed in 2021 contain only 311 unique signing timestamps (Table 5), which is roughly equal to the number of days in 2021 minus weekends where we rarely see new ROAs. We have also confirmed that this behavior is present since ARIN started its RPKI service in September 2012.

For LACNIC, the results are not as clear. We do observe an abnormally high number of ROAs with the NotBefore time set to 03:00 UTC but not all. This is because it affects only the API, which was released in 2021, and thus only recently used in the LACNIC region.

Unified ROA (APNIC, RIPE) The second difference is the number of routes encapsulated in each ROA object. RFC6482 [33] specifies that a ROA object has only one ASN but a list of prefixes and corresponding maximum length attributes. Hence for prefixes of a single authorized ASN, the RIR can maintain one unified ROA with all prefixes or multiple ROAs with one prefix each.

Grouping multiple prefixes in a single ROA object has the advantage of a simpler file management as there is a single file for each (organization, ASN) pair. The ROA snapshot from 31st December 2021 shows that APNIC and RIPE have opted for this unified ROA management. To illustrate this, Table 5 depicts the number of published ROA objects and the number of corresponding routes (prefix, origin ASN). APNIC has on average 12 routes per ROA (5 routes on average for RIPE), whereas ARIN has mostly ROAs with 1 route. This difference is also visible on RIR portals and APIs. For example, APNIC and RIPE only require the route and max-length value to create a ROA, whereas other RIRs have more specific requirements, including a ROA unique identifier.

Although APNIC and RIPE unified ROAs provide simpler file management, they substantially complicate ROA signing and revocation mechanisms. For example, given a single ROA authorizing two prefixes originated by AS65536, If an organization requests the creation of a ROA for a new prefix for AS65536, then the Certification Authority has to revoke the previous ROA and create a new ROA including all three prefixes. Similarly, if someone requests the revocation of one of the prefixes, the Certification Authority has to revoke the ROA and create a new ROA with the remaining prefixes. Thus, in both cases, involving two cryptographic operations for a single query. This may explain why APNIC has a 20 minutes batch to allow it to collect multiple user queries and produce a single ROA file.

6 Discussion

Setting up these experiments and maintaining them over several months was an eye-opener to the challenges that operators face with RPKI and RIRs.

First, the procedures and requirements to obtain resources, activate RPKI, and manage ROAs for the five RIRs are all quite different. In addition, the lack of APIs to manage RPKI resources for APNIC and AFRINIC makes automation a lot more challenging. We implemented Selenium scripts for AFRINIC and APNIC beacons, which is not trivial given the security measures employed by RIR portals (e.g., two-factor authentication and password renewal) and need adjustments whenever portals are updated.

Second, the need for continued monitoring of the management, control, and data planes is crucial to ensure proper operation of all components involved

and impacted by RPKI. For example, one of our AFRINIC beacons failed for multiple days because one of the ROA was left un-revoked by the Certification Authority, even though our deletion query succeeded and it had disappeared from the AFRINIC web interface. We only noticed this problem in our data plane measurements thanks to the prefix visualization. APNIC also automates the creation/deletion of an IRR route object corresponding to RPKI operations which had painful effects on our experiments, and likely to cause pain to operators.

Third, the use of redundant RPs is obvious for a commercial ISP but inevitably slows the responsiveness of routers withdrawing prefixes for deleted ROAs. This may be an unexpected behavior that requires operators to experiment with different configurations.

Fourth, the RPKI ecosystem is rapidly evolving. During the course of these experiments, popular RPs had numerous bug fixes, including fixes that may impact the measured delays. It is however hard to track when operators are applying these updates. The RIRs' services have also been evolving, for instance APNIC has recently started experimenting with an API for managing RPKI [34].

Finally, less obvious but still very important is the use of a synchronized clock in the UTC timezone. Certification Authorities, Relying Parties, and any software that deals with ROA creation/deletion/validation should run their operations using a single timezone, UTC as used by the hardware security modules, to prevent delays and mismatches in ROA management as observed in Section 4.1.

7 Related Work

As RPKI deployment is gaining more traction in network operations, understanding the end-to-end delay of the ROV supply chain is extremely important. Previous research has focused mostly on measuring the deployment and adoption of RPKI [14, 35, 36] or on the security of the underlying infrastructure [37, 38], rather than on operational considerations, especially the propagation time.

Recommendations of timing parameters such as Relying Party refresh time are briefly mentioned in RFCs [39]. Other delay factors between the user query and the corresponding impact on BGP have not yet been well investigated. There are currently no BCP (Best Current Practice) documents on how to maintain reasonable RPKI end-to-end delay, aside from a currently inactive Internet-draft [23], which provides some, possibly overly liberal, high-level guidance on the frequency and refresh time intervals for Relying Party software.

One recent study from Kristoff et al. [24] collected access log information from both hosted and delegated RPKI Certification Authorities. This study analyzed the refresh intervals and observed the somewhat erratic fetching behavior of Relying Parties (RPs) - potentially affecting the overall propagation delay. In our study, we go a step further by understanding the end-to-end delay between the user ROA creation and the impact on BGP. We collected data from the RPKI management plane, the BGP control plane, as well as the data plane.

Finally, a study by Hlavacek et al. [40], performed data-plane experiments, in addition to the control plane, to evaluate ROV on the Internet. They analyzed and correlated the results of their study to identify the number of ASes enforcing ROV but no delay characterization was performed.

8 Conclusion

In this paper, we designed wide-ranging experiments to measure the timing and effects of the propagation of ROAs on the management, control, and data planes. This enabled us to track how ROAs are disseminated - starting from the moment creation is triggered through the RIRs' API/portals, then signed by the hosted Certificate Authorities, published at their respective Publication Points, to the moment they are fetched and validated by RPs, consequentially seeing routers announcing new routes in BGP, and then affecting delay and reachability on the data plane. We found ROA management issues for two RIRs and discovered that RIRs usually publish new RPKI information within 5 minutes, except APNIC which is 10 minutes slower. For ISPs, we observe disparate behaviors in the control and data planes between when routes are validated or invalidated by a ROA creation or deletion. At the ISP level, we observed that the reaction time following a ROA deletion is much longer due to BGP and multi-RP deployment that require complete ROA withdrawals on all RPs for a route to be withdrawn. Predicting prefix reachability and the BGP convergence time is getting even harder as it requires insights about which networks are implementing ROV and how quickly each reacts to RPKI changes. This study reveals some of the complexity added by RPKI to basic routing operations.

Ethics This work does not raise ethical issues. It is focussed on the reachability of experimental prefixes delegated to us by the RIRs specifically for the time of the experiment. These prefixes were cleared for advertisements by our providers and documented in the IRR databases. A webpage describing the experiment was available throughout the experiment (https://github.com/romain-fontugne/rov-timing). In addition, our work does not involve personal identification data.

Acknowledgments We thank the anonymous reviewers and our shepherd, Kyle Schomp, for their helpful comments. This research was supported in part by the MANRS Fellowship Program. We would like to thank the five RIRs for providing experimental prefixes to carry out this study. Special thanks to the engineers from ARIN and LACNIC to have responded in a timely manner to our queries with regards to the issues discovered in the ROA signing time, and to the network operators who helped confirm our hypotheses and details of their operations.

Bibliography

[1] Yakov Rekhter, Susan Hares, and Tony Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, January 2006.

[2] Charles Lynn. X.509 Extensions for Authorization of IP Addresses, AS Numbers, and Routers within an AS. Internet-Draft draft-clynn-bgp-x509-auth-00, Internet Engineering Task Force.

[3] Matt Lepinski and Stephen Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, February 2012.

[4] Prodosh Mohapatra, John Scudder, David Ward, Randy Bush, and Rob Austein. BGP Prefix Origin Validation. RFC 6811, January 2013.

[5] Z Morley Mao, Randy Bush, Timothy G Griffin, and Matthew Roughan. BGP beacons. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 1–14, 2003.

[6] Alberto Garcia-Martinez and Marcelo Bagnulo. Measuring bgp route propagation times. *IEEE Communications Letters*, 23(12):2432–2436, 2019.

[7] Bahaa Al-Musawi. Common pitfalls in RPKI deployment and how to avoid them, Apr 2021.

[8] Tomas Hlavacek, Italo Cunha, Yossi Gilad, Amir Herzberg, Ethan Katz-Bassett, Michael Schapira, and Haya Shulman. DISCO: Sidestepping RPKI's deployment barriers. In *Network and Distributed System Security Symposium (NDSS)*, 2020.

[9] Daniele Iamartino, Cristel Pelsser, and Randy Bush. Measuring BGP route origin registration and validation. In *International Conference on Passive and Active Network Measurement*, pages 28–40. Springer, 2015.

[10] Massimo Candela. A One-Year Review of RPKI Operations, RIPE 84, May 2022.

[11] Massimo Candela. One Does Not Simply "Deploy RPKI", MANRS blog, July 2022.

[12] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. ARTEMIS: Neutralizing BGP hijacking within a minute. *IEEE/ACM Transactions on Networking*, 26(6):2471–2486, 2018.

[13] Taiji Kimura. Long Chopsticks in Heaven - When Packets Dropped Using ROA, May 2019.

[14] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are we there yet? on RPKI's deployment and security. *Cryptology ePrint Archive*, 2016.

[15] RIPE NCC. Routing Information Service (RIS), May 2022.

[16] RIPE NCC. RIPE Atlas, May 2022.

[17] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and David Cooper. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008.

[18] University Oregon. Route Views, September 2022.

[19] Selenium. Selenium webdriver, September 2022.
[20] Job Snijders. RPKIviews, May 2022.
[21] OpenBSD. rpki-client, May 2022.
[22] R. Housley. Cryptographic Message Syntax (CMS). RFC 5652, September 2009.
[23] Randy Bush, Jay Borkenhagen, Tim Bruijnzeels, and Job Snijders. Timing Parameters in the RPKI based Route Origin Validation Supply Chain. Internet-Draft draft-ietf-sidrops-rpki-rov-timing-06, Internet Engineering Task Force, February 2022. Work in Progress.
[24] John Kristoff, Randy Bush, Chris Kanich, George Michaelson, Amreesh Phokeer, Thomas C. Schmidt, and Matthias Wählisch. On Measuring RPKI Relying Parties. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, page 484–491, New York, NY, USA, 2020. Association for Computing Machinery.
[25] Thomas Alfroy, Thomas Holterbach, and Cristel Pelsser. MVP: Measuring Internet routing from the most valuable points. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 770–771, New York, NY, USA, 2022. Association for Computing Machinery.
[26] Romain Fontugne, Anant Shah, and Emile Aben. The (thin) bridges of AS connectivity: Measuring dependency using AS hegemony. In *International Conference on Passive and Active Network Measurement*, pages 216–227. Springer, 2018.
[27] Porapat Ongkanchana, Romain Fontugne, Hiroshi Esaki, Job Snijders, and Emile Aben. Hunting BGP zombies in the wild. In *Proceedings of the Applied Networking Research Workshop*, pages 1–7, 2021.
[28] Cloudflare. Is BGP safe yet? No., May 2022.
[29] Routinator. Changelog (v0.11.2), April 2022.
[30] Romain Fontugne. The Routing Game: Hunting Invalid Routes., November 2021.
[31] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. AS relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 243–256, 2013.
[32] RIPE NCC. RIPE NCC's RPKI repository archive, May 2022.
[33] Matt Lepinski, Derrick Kong, and Stephen Kent. A Profile for Route Origin Authorizations (ROAs). RFC 6482, February 2012.
[34] Tom Harrison. APNIC Registry API, APNIC blog, March 2022.
[35] Andreas Reuter, Randy Bush, Italo Cunha, Ethan Katz-Bassett, Thomas C Schmidt, and Matthias Wählisch. Towards a rigorous methodology for measuring adoption of rpki route validation and filtering. *ACM SIGCOMM Computer Communication Review*, 48(1):19–27, 2018.
[36] Taejoong Chung, Emile Aben, Tim Bruijnzeels, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, Roland van Rijswijk-Deij, John Rula, et al. RPKI is coming of age: A longitudinal study of RPKI deployment and invalid route origins. In *Proceedings of the Internet Measurement Conference*, pages 406–419, 2019.

[37] Yossi Gilad, Omar Sagga, and Sharon Goldberg. Maxlength considered harmful to the RPKI. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 101–107, 2017.

[38] Tomas Hlavacek, Philipp Jeitner, Donika Mirdita, Haya Shulman, and Michael Waidner. Stalloris: RPKI downgrade attack. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, August 2022. USENIX Association.

[39] Randy Bush. Origin validation operation based on the Resource Public Key Infrastructure (RPKI). *IETF RFC7115 (January 2014)*, 2014.

[40] Tomas Hlavacek, Amir Herzberg, Haya Shulman, and Michael Waidner. Practical experience: Methodologies for measuring route origin validation. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 634–641. IEEE, 2018.

A Appendix

A.1 Data plane availability in IPv6

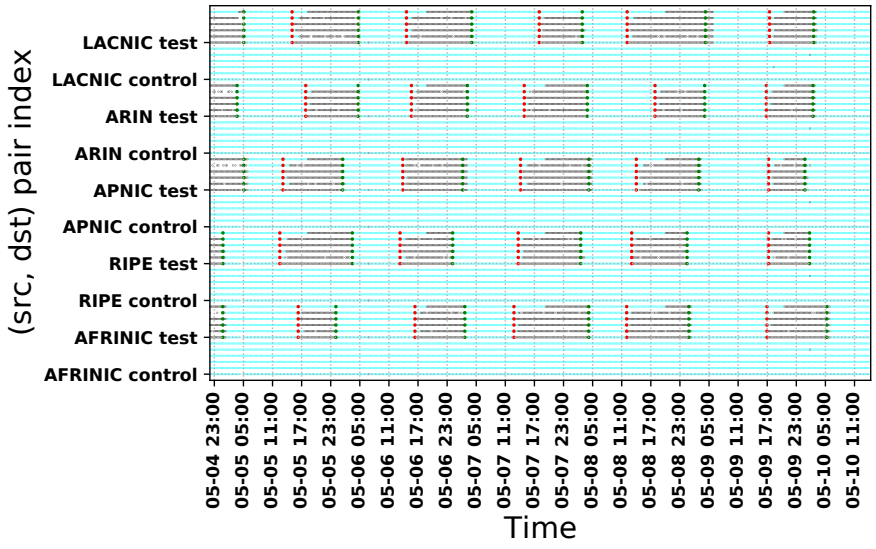


Fig. 10: IPv6: Effects of ROA creation (green dots) and ROA deletion (red dots) on prefix reachability (cyan dot) and unreachability (black dot) in traceroute. Each line shows a different Atlas probe/prefix pair. Delay between ROA deletion and unreachability highly varies depending on the topology.

A.2 BGP Update delay after ARIN fix

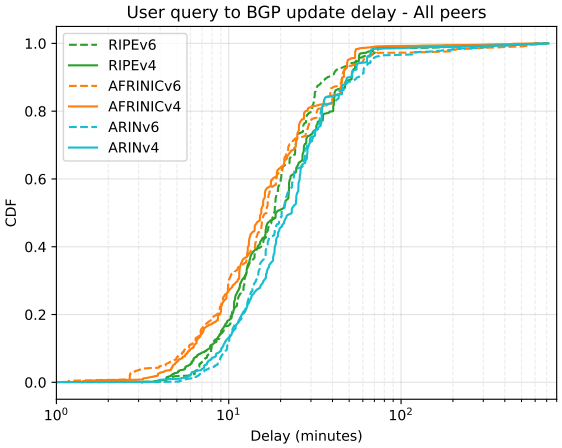


Fig. 11: ROA creation after ARIN's fix. RRC00 and RRC01 peers from April 21st to May 15th 2022. APNIC and LACNIC are not plotted to improve readability. ARIN's user query to BGP delay distributions became similar to the ones of AFRINIC and RIPE.

A.3 Reproducibility

Our experimental data is publicly available in order to make the results of this work entirely reproducible. Our source code and logs of user query time are available at https://github.com/romain-fontugne/rov-timing.

The list of experimental prefixes obtained from the five RIRs are shown in Table 6.

The list of RIPE Atlas measurement IDs corresponding to the traceroute measurements for this study are in Table 7.

Table 6: IP prefixes used for our first experiment.

RIR	type	IPv4	IPv6
AFRINIC	control	102.218.96.0/24	2001:43f8:df0::/48
AFRINIC	test	102.218.97.0/24	2001:43f8:df1::/48
APNIC	control	103.171.218.0/24	2001:DF7:5380::/48
APNIC	test	103.171.219.0/24	2001:DF7:5381::/48
ARIN	control	165.140.104.0/24	2620:9E:6000::/48
ARIN	test	165.140.105.0/24	2620:9E:6001::/48
LACNIC	control	201.219.252.0/24	2801:1e:1800::/48
LACNIC	test	201.219.253.0/24	2801:1e:1801::/48
RIPE	control	151.216.4.0/24	2001:7fc:2::/48
RIPE	test	151.216.5.0/24	2001:7fc:3::/48

Table 7: RIPE Atlas measurement IDs corresponding to traceroute data analyzed in this study.

ID	target	RIR	type
40388150	103.171.218.1	APNIC	control
40388151	103.171.219.1	APNIC	test
40388152	2001:DF7:5380::1	APNIC	control
40388153	2001:DF7:5381::1	APNIC	test
40388154	151.216.4.1	RIPE	control
40388155	151.216.5.1	RIPE	test
40388156	2001:7fc:2::1	RIPE	control
40388157	2001:7fc:3::1	RIPE	test
40388158	102.218.96.1	AFRINIC	control
40388159	102.218.97.1	AFRINIC	test
40388160	2001:43f8:df0::1	AFRINIC	control
40388161	2001:43f8:df1::1	AFRINIC	test
40388162	165.140.104.1	ARIN	control
40388163	165.140.105.1	ARIN	test
40388164	2620:9E:6000::1	ARIN	control
40388165	2620:9E:6001::1	ARIN	test
40388166	201.219.252.1	LACNIC	control
40388167	201.219.253.1	LACNIC	test
40388168	2801:1e:1800::1	LACNIC	control
40388169	2801:1e:1801::1	LACNIC	test