RScope: Unveiling Global ROV Deployments and Dependencies in the Post-ROV Era

Weitong Li*
Virginia Tech
USA
weitongli@vt.edu

Yongzhe Xu*
Virginia Tech
USA
yongzhe@vt.edu

Mingwei Zhang
Cloudflare
USA
mingwei@cloudflare.com

Vasileios Giotsas
Cloudflare
USA
vasilis@cloudflare.com

Taejoong Chung
Virginia Tech
USA
tijay@vt.edu

ABSTRACT

The Resource Public Key Infrastructure (RPKI) and Route Origin Validation (ROV) are critical to securing Internet routing, yet measuring real-world ROV deployment and dependencies remains challenging. Existing methods (control-plane analysis and data-plane probing) often struggle to distinguish local ROV adoption from upstream filtering at scale and face diminishing visibility as ROV adoption grows.

We present RScope, a measurement framework that dynamically manipulates Route Origin Authorizations (ROAs) from controlled publication points to isolate the impact of individual Relying Party (RP) servers. By selectively invalidating prefixes for specific RPs and probing connectivity changes, RScope maps dependencies among 21,827 ASes, their transit providers, and RP infrastructure.

Our findings reveal that 1,127 ASes actively deploy ROV in IPv6, while 1,815 benefit from upstream protection, leaving 69.6% of such "protected" ASes vulnerable to local hijacks. Notably, ROV deployment by major ISPs strongly shape global security, enabling it in the top 100 ASes protects 27.6% of networks, but disabling it in Tier-1s endangers 23.8%. We also uncover operational delays: router enforcement lags ROA updates by 37 minutes on average. These results highlight systemic risks in incremental ROV adoption, emphasizing the need for broader deployment. We open-source tools and datasets to support ongoing routing security efforts.

CCS CONCEPTS

• **General and reference** → **Measurement**; • **Networks** → **Routing protocols**; **Security protocols**.

KEYWORDS

Resource Public Key Infrastructure, RPKI, Route Origin Validation, Network Measurement

*Both authors contributed equally to this research.

IMC '26, October 12–16, 2026, Karlsruhe, Germany

ACM ISBN 979-8-4007-2327-8/26/10
https://doi.org/10.1145/3777912.3809144

ACM Reference Format:
Weitong Li, Yongzhe Xu, Mingwei Zhang, Vasileios Giotsas, and Taejoong Chung. 2026. RScope: Unveiling Global ROV Deployments and Dependencies in the Post-ROV Era. In *Proceedings of the 2026 ACM Internet Measurement Conference (IMC '26), October 12–16, 2026, Karlsruhe, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3777912.3809144

1 INTRODUCTION

The Border Gateway Protocol (BGP) underpins interdomain routing on the global Internet but lacks intrinsic security features, leaving it vulnerable to prefix hijacking when an Autonomous System (AS) illegitimately announces IP prefixes it is not authorized. Such attacks can cause traffic interception, outages, and data breaches, ultimately undermining the stability of global communications [2, 3, 5, 7].

To address these weaknesses, the Resource Public Key Infrastructure (RPKI) [29] provides cryptographic attestations—Route Origin Authorizations (ROAs)—that bind IP prefixes to their legitimate origin ASes, enabling networks to perform Route Origin Validation (ROV) and drop invalid routes.

Despite this capability, real-world ROV deployment remains inconsistent: some ASes enforce ROV locally, others rely on upstream providers for "collateral" filtering, and many implement no ROV at all [32]. Because Internet routing is hierarchical, ROV's overall effectiveness depends not only on direct adoption but also on the complex dependencies among transit providers.

Previous efforts to measure ROV rely on two main strategies: (1) Control-plane methods focus on BGP route collectors to detect dropped prefixes [20, 39], but these collectors offer only partial coverage and cannot easily distinguish local ROV deployment from upstream filtering. (2) Data-plane methods either inject invalid prefixes [9, 21, 26, 45] or leverage real-world invalid prefixes [32] to observe who filters in practice.

Although these approaches reveal whether an AS appears "protected", they cannot always explain *how* that protection arises, especially as widespread ROV adoption increasingly prevents invalid prefixes from propagating on the Internet [46]. Moreover, the RPKI ecosystem itself is multifaceted, involving Publication Points (PPs) that host ROAs, Relying Parties (RPs) that collect and validate them, RTR (RPKI-to-Router) servers that distribute the validated data, and finally routers that enforce filtering policies.

This paper introduces RScope, a measurement framework that systematically uncovers the ROV*ecosystem* by dynamically serving customized ROAs from controlled publication points to specific RP servers, invalidating prefixes for selected ASes. We pair this selective ROA manipulation with large-scale probing to observe how invalid routes are filtered, thereby pinpointing the dependencies between ASes, transit providers, and the RP servers on which they rely. With RScope, we also measure how swiftly ROA updates propagate to routers, providing operational insights critical for strengthening RPKI's reliability. By dissecting the RPKI trust chain from ROA publication to router enforcement, our work highlights both the progress and pitfalls of ROV adoption. Our contributions are:

- *RScope Framework.* We propose a scalable methodology that combines dynamic ROA manipulation with Internet-scale probing, overcoming the limitations of relying on globally propagated RPKI-invalid prefixes to measure ROV.
- *Ecosystem Analysis.* We discover that 1,127 ASes actively deploy ROV in IPv6 while 1,815 rely on upstream protection, revealing systemic vulnerabilities (e.g., local hijacks) among 69.6% of "protected" ASes.
- *Operational Insights.* We conduct the first large-scale measurement of RP-server strategies and validation latency. We also highlight distinct ROV deployment patterns and show that router enforcement lags ROA updates by an average of 37 minutes.

Our findings highlight both progress and pitfalls in the transition to a post-ROV era, where adoption is growing but uneven, and collateral protection often masks lingering vulnerabilities.

To foster reproducibility and support ongoing routing security efforts, we open-source all measurement tools and datasets and integrate our results into the existing RoVista platform [32, 46], enabling the community to continuously track ROV deployment and dependencies:

https://rovista.netsecurelab.org

2 BACKGROUND AND RELATED WORK

2.1 BGP

The Border Gateway Protocol (BGP) is the de facto interdomain routing protocol that enables autonomous systems (ASes) to exchange global reachability information. Conceptually, BGP functions as a path-vector protocol: each AS advertises IP prefixes it can reach, appending its own AS number to the `AS_PATH`. These announcements propagate transitively across the Internet, allowing routers to discover multiple potential paths and select the best route according to local policies and global constraints. A simple BGP announcement might appear as follows:

```
Prefix: 45.3.0.0/16   AS_PATH: AS174   AS40220
```

Since the original BGP does not natively include security features, it is vulnerable to attacks such as *prefix hijacking* [2, 3, 7], where an attacker announces IP space it does not own. Without cryptographic validation, routers simply trust these route advertisements, potentially allowing traffic to be diverted or dropped.

2.2 RPKI

The Resource Public Key Infrastructure (RPKI) [29] addresses BGP's inherent insecurity by binding Internet number resources (IP addresses and AS numbers) to their legitimate holders through cryptographic means. At a high level, RPKI allows organizations to create cryptographically signed objects, publish them to distributed *publication points (PPs)*, and enable ROV to detect and filter bogus routes.

Route Origin Authorization (ROA) A *Route Origin Authorization* (ROA) is a cryptographically signed object that declares which AS is authorized to originate one or more IP prefixes. To create a ROA, a resource holder first obtains a *CA certificate* linking its public key to the allocated IP space or ASNs. It then specifies: (1) the authorized origin AS number and (2) a set of covered IP prefixes, and (3) an optional `maxLength` parameter denoting the largest permitted subnet length. The final ROA is signed with the resource holder's private key and uploaded to *publication points* (PPs) operated by RIRs (e.g., ARIN) or delegated entities (e.g., National and Local Internet Registries, or NIRs/LIRs).

Route Origin Validation (ROV) To enforce BGP security, network operators deploy *Route Origin Validation* (ROV). Relying Party (RP) software periodically fetches all RPKI objects (including ROAs) from publication points using RRDP or RSYNC [34]. After cryptographic checks, it produces a list of validated tuples (ASN, ROA prefix, prefix length), called *Validated ROA Payloads* (VRPs). Routers that support ROV then retrieve these VRPs (via the RPKI-to-Router (RTR) protocol) and label incoming BGP routes as:

- *Unknown*: No covering ROA exists.
- *Valid*: Origin AS matches a ROA, and prefix length is within `maxLength`.
- *Invalid*: Origin AS or prefix length does not match any covering ROA.

Operators can configure routers to drop, de-preference, or permit routes in each category, mitigating hijacks and misconfigurations. Nevertheless, ROV adoption remains uneven across the global Internet.

2.3 ROV Ecosystem

The ROV ecosystem includes multiple interdependent entities and protocols that publish, retrieve, validate, and distribute ROAs. We detail the main components and their roles:

- Publication Points (PPs), which host RPKI objects, including ROAs, CA certificates, and manifests. Each of the five RIRs maintains its own default publication point; some National Internet Registries (NIRs) and Local Internet Registries (LIRs) also run separate delegated repositories. In a *hosted* setup, the RIR or NIR manages all operational details (key material, certificate rotation, and repository hosting) for network operators. By contrast, in a *delegated RPKI* model, an LIR obtains a child CA certificate from its parent RIR and operates an independent PP, retaining full control over private keys and signing processes.
- Relying Parties (RPs), which are the backbone of RPKI validation, retrieving and verifying published ROAs and other RPKI objects from *all* PPs. They typically attempt to download objects over

RRDP first, falling back to *RSYNC* if necessary. After checking cryptographic signatures and certificate chains, they compile the valid ROAs into VRPs. Because the corpus of RPKI data can be sizable, RPs generally operate on fixed intervals (e.g., every 60 minutes for `Routinator`). Although best practices recommend deploying multiple RPs for redundancy, some operators prefer *public* RP services (e.g., Cloudflare [13], OpenBSD [38], RIPE [44], NTT [36]) to reduce management overhead. In practice, how multiple RPs are used depends on local policy; for instance, operators may merge VRPs from all RPs (union), require agreement from all RPs (intersection); this complicates one-to-one mapping between RP servers and the ASes that use them (detailed in §5.6).

- RPKI-to-Router (RTR) Protocol, which delivers VRPs from RPs (or RTR servers) to routers. In many deployments, an *RTR server* polls one or more RPs at intervals (e.g., every 10 minutes) and provides the fetched VRPs to ROV-enabled routers. The routers then periodically query the RTR server for updates, often defaulting to 30–60 minute polling cycles. Although RFC 8210 allows for real-time "Serial Notifies", some router implementations ignore these and adhere strictly to their configured intervals.

2.4 Understanding ROV Status

ROV Deployment Some studies attempt to detect where routes are filtered using BGP route collectors (e.g., RouteViews [47] and RIPE RIS [42]), examining changes in the `AS_PATH` for RPKI-invalid routes. For example, Hlavacek et al. [49] identified only 296 ASes deploying ROV, in part because route collectors are sparse and invalid announcements seldom propagate far—especially as more ASes adopt ROV. Similarly, Chen et al. [8] and Reuter et al. [39] relied on data-plane traceroutes from multiple vantage points, comparing RPKI-invalid and RPKI-valid routes, but still faced limited visibility. While this sheds light on potential route filtering, the approach is still constrained by the number and distribution of traceroute vantage points and need to infer filtering behavior indirectly from observed path changes.

ROV Protection Other works inject or observe RPKI-invalid prefixes directly and test whether these prefixes remain *reachable* from disparate vantage points [9, 21, 26, 32, 45]. Methods include ICMP Echo, traceroute, HTTP requests, and measurements from dedicated platforms like RIPE Atlas [8, 40, 49] or commercial networks like Google Ads [21]. Yet growing ROV adoption—especially among tier-1 ISPs—often prevents invalid announcements from propagating beyond a few AS hops, limiting measurement coverage and obscuring whether ROV is being enforced by the origin AS or a transit/upstream AS [40]. For example, RoVista—an approach that leverages naturally occurring RPKI-invalid prefixes—recently saw its pool of *tNodes* (invalid origins) shrink from 30 in April 2024 to only 7 by April 2025 [46], illustrating how broader ROV adoption can reduce visibility for data-plane measurements.

Figure 1 shows how the visibility of RPKI-invalid prefixes has declined from 2023 to 2025. We analyze RouteViews [47] BGP RIB dumps from all of their collectors on January 1st of 2023, 2024, and 2025, validating routes using RIPE Historic RPKI Data [41]. We then calculated the fraction of RouteViews peers that observed each invalid `(prefix, origin AS)` pair. Whereas in 2023, 22% of invalid prefixes were visible to at least 30% of peers, by 2025 the figure had dropped to 6%. As ROV filtering intensifies, it becomes harder to find globally visible invalid routes, reducing the utility of data-plane probing for large-scale ROV measurement.

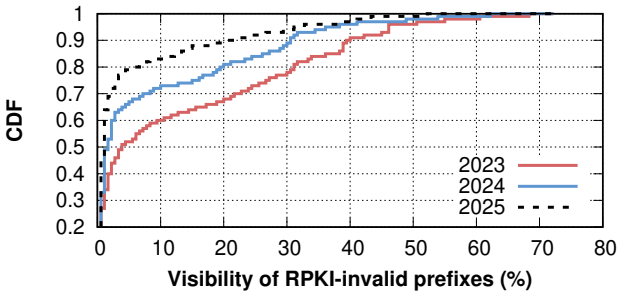

Figure 1: Visibility of RPKI-invalid prefixes has decreased notably from 2023 to 2025, suggesting expanding ROV adoption.

2.5 Toward a Holistic View of the ROV Ecosystem

Beyond identifying ROV deployment or protection, some studies have examined key ecosystem components.

For instance, Hlavacek et al. [22] studied DNS infrastructure supporting RPKI validators, focusing on DNSSEC usage by hosting their own publication points and authoritative DNS servers. Kristof et al. [28] measured publicly visible RP software by running custom PPs and logging validator connections. Fontugne et al. [16] investigated latencies between ROA publication and route filtering in the data plane by updating ROAs in RIR's publication points, and measuring the time each network took to filter the invalid route.

Nevertheless, existing work largely concentrates on publication points, RP software usage, or specific network vantage points in isolation. We lack granular insights into *which* specific RP servers different ASes trust, the degree of multi-RP redundancy, and whether some ASes merely "inherit" valid routes from upstream deployments. This leaves unresolved questions about inter-AS ROV dependencies and the practical configurations that underpin large-scale RPKI adoption.

3 RSCOPE: SYSTEM DESIGN

3.1 Overview

We first present an overview of our measurement framework, RScope. In RScope, we run our own delegated RPKI publication point, allowing us to freely issue and serve ROAs under our control. This setup leverages two fundamental properties of the RPKI ecosystem:

(a) RPKI does not enforce *global consistency* of ROAs, allowing an operator to serve different RPKI objects to different Relying Parties (RPs).

(b) All RPs are expected to fetch all ROAs from *all* publication points, which means any RP in the ecosystem will periodically contact our publication point.

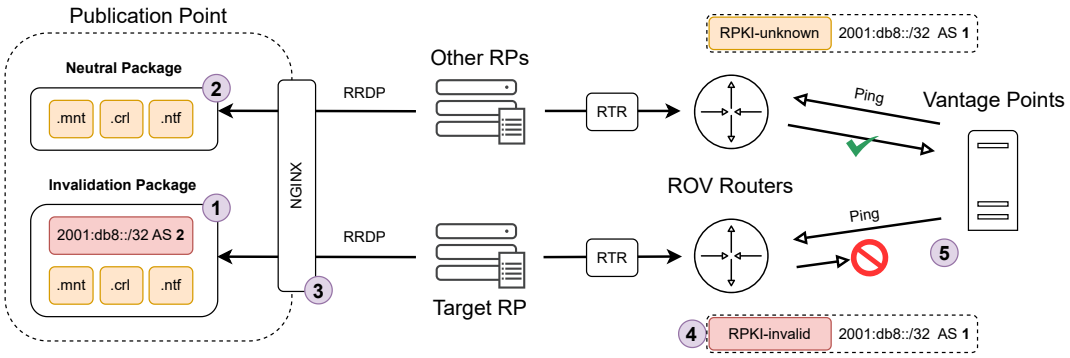

Figure 2: Overview of our measurement framework. We operate a *dynamic publication point* that selectively serves different RPKI objects to specific RP servers, effectively creating "RP-targeted invalid" prefixes for each targeted RP. We then perform active scanning from these prefixes to identify which ASes stop forwarding them, thereby revealing the dependencies between ASes and their relied-upon RP servers.

Building on these properties, we create *targeted invalid* prefixes for a specific RP, which we call targeted invalid prefixes, while leaving them *unknown* to all others. When an RP receives a deliberately crafted ROA that conflicts with our route announcement, it labels the prefix as *RPKI-invalid* for any router(s) it serves. As a result, any AS relying on that RP for RPKI validation will filter the affected prefix, revealing a direct dependency on the targeted RP (or their upstreams doing ROV). Meanwhile, since other RPs do not receive the conflicting ROA, the prefix remains *unknown* (and thus not filtered) for all other ASes.

Figure 2 illustrates our approach. We operate:

(a) A *dynamic publication point*, which selectively returns ROAs that render our BGP announcements *invalid* for targeted RPs or leave them *unknown* for all others.

(b) An *Internet scanner*, which sends active probes (e.g., ICMP pings) periodically from the contested prefix to identify which ASes drop or forward traffic.

By correlating which RPs receive a ROA that invalidates our BGP announcement with the resulting loss of reachability, we can effectively map each AS to the RP(s) it depends on. This model remains robust even if future global ROV deployment becomes so pervasive that truly invalid prefixes fail to propagate: our technique only makes the route invalid to a *single* RP at a time, ensuring the prefix remains routable to the rest of the Internet.

3.2 Dynamic Publication Point: Creating RP-Targeted Invalid Prefixes

We operate our own RPKI CA and publication point (in delegated mode) under a parent RIR, allowing us to freely issue and distribute RPKI objects for our assigned prefix. For each prefix, we publish two distinct *packages* of RPKI objects:

- *Invalidation Package*, which contains a ROA that intentionally specifies an *incorrect* origin ASN (one that is not actually announcing the route). Cryptographically, this ROA is perfectly valid (signed by our CA), but from the perspective of any RP that receives it, our announced prefix will be judged RPKI-invalid. (Figure 2 ①)
- *Neutral Package*, which omits the ROA for our announced prefix, making it Unknown in terms of RPKI validation (Figure 2 ②). Because no ROA covers the prefix, it is neither RPKI-Valid nor RPKI-Invalid for RPs that see only this package.

Since RPs typically fetch RPKI objects via RRDP over HTTP or HTTPS [48], we configure an HTTP server (e.g., NGINX) behind our delegated publication point (Figure 2 ③). This server *redirects* each RP to either the Invalidation or the Neutral package based on the RP's IP address or User-Agent string. In doing so, we ensure that only *one* targeted RP sees the ROA making our announced RPKI-invalid, while all other RPs mark it as Unknown. This selective invalidation isolates the effect to those ASes relying on that specific RP, allowing our prefix to continue propagating in the rest of the Internet.

3.3 Internet Scanner: Mapping ASes to RPs

After establishing our "targeted invalid" prefix for a given RP, we use active data-plane probing to discover which ASes filter the route, thereby revealing their reliance on that RP's validation. Our workflow is as follows:

(a) *Baseline Reachability:* From the prefix we control, we send ICMP Echo or other lightweight probes (e.g., TCP SYN) to a large set of publicly responsive IP addresses across numerous ASes. We record which hosts respond successfully and note their origin ASes.

(b) *Selective Invalid ROA Deployment:* We then serve the *Invalidation Package* exclusively to the *targeted RP*, while continuing to serve the *Neutral Package* to all other RPs. As the targeted RP updates its VRP set (typically within 10~60 minutes [16]), any AS relying on that RP for validation will see the route become RPKI-Invalid and drop traffic (Figure 2 ④).

(c) *Re-Probing:* Once we expect the invalidation to have propagated (accounting for RP polling intervals, RTR updates, etc.), we send the same probes again from our prefix. Hosts that previously responded but now fail to reply are considered *"disconnected"* due to the invalidation. We label the ASes of these hosts as reliant on the targeted RP (Figure 2 ⑤).

(d) *Iteration Over All RPs:* We repeat the above steps for each RP of interest—serving the Invalidation Package to one RP at a time. This process gradually builds a map of AS–RP dependencies, as well as highlights the presence of multi-RP configurations (if an AS only filters the prefix when *all* of its RPs receive the invalidation, for example).

Because we can dynamically revise the objects returned by our delegated publication point, this process scales to a large number of RPs. Moreover, even if an AS is a downstream customer of another ROV-enabled AS, we observe the same disconnecting effect, letting us detect not just direct ROV deployments but also *upstream-based* protection. In later sections, we discuss additional refinements to improve accuracy, and also how we distinguish between ROV deployment and upstream protection.

4 RSCOPE: IMPLEMENTATION

In this section, we detail our real-world deployment of the RScope framework and present initial findings.

4.1 Prefix and RPKI setup

Delegated RPKI. We operate as an LIR under ARIN in a delegated RPKI model, managing our own RPKI CA and publication point. Using Krill and an NGINX HTTP server, we can issue or revoke ROAs at will and serve them selectively.

Discovering RP Servers. After deploying our initial publication point, we monitored which IP addresses fetched our published RPKI objects over a two-week period in December 2024. We recorded:

- 4,322 unique IPs from 2,031 ASes connecting to our publication point.
- However, not every incoming IP address corresponded to a genuine RP (e.g., bots). After discarding hosts that polled less than once per day, we retained 3,516 IPs drawn from 1,952 ASes, which we regard as bona fide RPs.

We acknowledge the theoretical possibility that a single RP server might use multiple IP addresses over time, or that multiple RP servers might share one IP address. However, our manual validation with network operators (the ones in §5.2) found no evidence of such deployment strategies. Therefore, in this paper, we operationally define an "RP server" as a unique IP address, under the assumption that it typically corresponds to a single RP server instance in practice.

RP Servers per ASN. Among the 1,952 ASes that host RP servers, we find that 68.3% host only a single server's IP, whereas 619 ASes deploy multiple. Notably, some networks maintain a large number of RP servers; for example, Amazon Web Services (AS 16509) runs about 72, and Vultr (AS 20473) operates around 58. This may reflect (1) ROV-enabled ASes distributing multiple RP servers for redundancy, or (2) organizations hosting RP servers in public cloud environments. Thus, ASes depending on multiple RP servers may exhibit different behaviors *if VRPs differ across them*; we revisit these multi-RP strategies in §5.6.

Motivation for IPv6. Deploying comprehensive measurements against the 3,516 RP servers we discover poses two major challenges:

(a) *IPv4 address scarcity.* Routing in IPv4 typically requires at least a /24 prefix, making it expensive or impractical to obtain *multiple* /24 prefixes for parallel tests. One might consider sequentially testing each RP with a single /24 prefix, but RPKI propagation delays mean each round can take more than 100 minutes (see below). If we had used only one IPv4 /24 prefix to test these RP servers sequentially, we would need ∼586 days to complete the experiments-impractical at global scale.

(b) *Propagation delays.* Beyond address-space concerns, RPKI updates (e.g., newly published or revoked ROAs) can require 100+ minutes [16] to fully propagate across the global set of RPs and their downstream routers, significantly prolonging sequential experiments.

To overcome these constraints, we rely primarily on IPv6. We obtained a /40 IPv6 allocation from ARIN and split it into 256 /48 subprefixes, enabling 256 parallel experiments per round. We continue to serve our RPKI publication point over *both* IPv4 and IPv6 to maximize reachability, but the *announced measurement prefixes* used for invalidation tests are exclusively IPv6. A consequence is that ASes enforcing ROV only on IPv4 fall outside our direct observation; to assess parity across address families, we additionally ran RScope with four IPv4 /24 prefixes against the top-100 ASes (by AS Rank) that our IPv6 experiments identified as ROV-deployed; *all of these also enforced ROV in IPv4 (detailed in §7.2).*

Measurement Prefix Announcements. We announce these 256 /48s from four vantage points: three Vultr data centers (Amsterdam, Seoul, Atlanta) and one node on a peering testbed in Brazil (UFMG). Each vantage point also has a separate RPKI-valid IPv6 address (used primarily as a control channel for verifying connectivity and mitigating route filtering on the way out).

Upstream Influence on Measurements. While RScope accurately attributes ROV to the AS that actually enforces it (rather than to in-the-middle ASes), the direct upstreams of our vantage points can still affect measurements in two ways:

- A partial-ROV upstream could be mislabeled as non-ROV if it whitelists prefixes from our origin AS (or enforces ROV only on IPv4), while filtering other links in practice.
- The count of ROV-protected ASes could be inflated for those upstreams from which we announce our experiment prefixes.

Our /48s are announced from four vantage points in the PEERING testbed—three Vultr data centers (Amsterdam, Seoul, Atlanta) and one node at UFMG. Across these, 14 ASes serve as direct or indirect upstreams, including tier-1 networks. To mitigate vantage-specific bias, *we use UFMG to test the upstreams of the Vultr vantage points and* vice versa. Two ASNs (AS1299 and AS2914), however, are indirect upstreams of all vantage points and cannot be eliminated from influence.

4.2 Dynamic Publication Points

A core component of our measurement methodology involves *selectively* invalidating each of our 256 /48 prefixes for precisely one targeted RP at a time, while leaving those same prefixes *unknown* to all other RPs. This design allows us to isolate the impact of local invalidation and map exactly which networks rely on each specific RP for RPKI-based route filtering.

Batched RPs. With a total of 3,516 RP servers discovered in our logs, we divide them into 14 groups, each containing up to 256 servers. In every experimental round, we:

(a) Publish the 256 invalidation packages and the single neutral package.
(b) Wait 3 hours to allow RP servers to fetch our updated ROAs and propagate any invalidation to their routers.[1]
(c) Conduct data-plane measurements (i.e., ICMP pings) for 1 hour, checking which ASes drop our invalid /48 prefix.
(d) Conclude the round, refresh all RPKI objects, and proceed to the next group.

After completing 14 rounds, we have effectively tested all 3,516 RP servers.

Refreshing ROAs Between Rounds. Each RP server receives 14 different sets of RPKI objects—one per round—every 4 (3+1) hours. We must ensure that no RP server continues to cache *old* objects in subsequent rounds.

Under the RRDP protocol, an RP server may perform incremental ("delta") updates only if the notification file references past serial numbers that align with its local cache. To *disable* delta updates, we omit all previous serial numbers from the notification file in each new round; additionally, once a round ends, we add all previously used objects to the Certificate Revocation List (CRL) [30] for the next round. This forces every RP server to *fully refetch* and revalidate the complete set of RPKI objects, thereby ensuring they discard any outdated ROAs.

4.3 Data-Plane Measurement

To detect which ASes filter the invalidated prefixes, we perform active data-plane measurements with ICMP echo requests. Specifically, we use a IPv6 Hitlist [25], select one responding address per /48, and limit ourselves to five targets per AS to minimize probing traffic. In total, we collect 51,075 responding IPv6 addresses from 21,827 ASes across 169 countries. At the time of measurement, there are 34,579 ASes visible in the global IPv6 routing table [24], our measurement covers 63.1% of them.

Measurement Procedure. For each experimental round:

(a) We send ICMP echo requests from the 256 /48 prefixes, each associated with a different invalidation package or the neutral package.
(b) Each ICMP request is retried up to 3 times (1-minute intervals) to mitigate transient network issues. A host is marked "unreachable" if *all* attempts fail.
(c) As a control, we also ping the same target addresses using a distinct RPKI-valid IPv6 source address before and after each round. This lets us confirm whether hosts remain online.

We ran this experiment in January 2025 and identified 51,075 responsive targets across 21,827 ASes. Of these, 2,911 ASes (13.3%) exhibited *disconnection after returning Invalidation Package to an RP*, indicating they rely on one or more of the 1,672 RPs for ROV protection. The next section delves into a deeper analysis of these RPs and the affected ASes.

[1]According to prior measurements [16], 97% of RPs reflect ROA changes within 100 minutes.

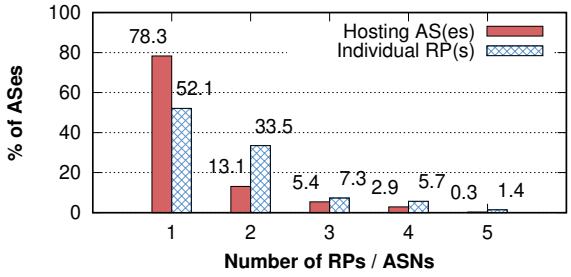

Figure 3: Fraction of ROV-protected ASes that rely on different RP servers, grouped by individual server versus hosting ASN.

5 MEASUREMENT RESULTS

In this section, we present the key findings from RScope, emphasizing how different ASes rely on particular RP servers for ROV. We use the umbrella term *ROV-Protected* for any AS that rejects BGP routes marked RPKI-Invalid.

5.1 ROV-Protected ASes and RP Servers

We begin by examining ASes that consistently filter our invalid prefixes; we identify 2,942 ASes, which consume VRPs from 1,672 distinct RP servers hosted in 1,319 ASes.

Figure 3 shows how many individual RP servers each ROV-protected AS relies on; we also consider grouping those servers by hosting ASN. Overall, we find that: (1) 1,533 (52.1%) ASes consume VRPs from exactly one RP server IP[2], (2) 777 (26.4%) of ASes use multiple RP servers hosted in the *same* ASN (e.g., a large ISP or cloud provider with redundant RP servers), and (3) the remaining 632 (21.5%) rely on multiple RP servers hosted in *different* ASNs—some connecting to as many as 5 distinct RPs. While deploying multiple RPs from separate networks is more robust (as recommended by RFCs [4]), the degree of redundancy appears operator-specific.

5.2 ROV Deployment vs. Upstream-Protected ROV

Although multiple ASes may filter invalid prefixes via the *same* RP, it remains unclear whether each AS has deployed ROV locally or simply benefits from an upstream that discards invalid routes before they arrive. We define two scenarios:

- *ROV Deployment:* The AS runs its own RTR software and ROV-capable routers, even if it fetches VRPs from a shared or public RP.
- *Upstream-Protected ROV (Collateral Benefit):* An upstream provider deploys ROV, so its downstream customers never receive invalid routes. Although these downstream ASes appear protected, they have not *themselves* implemented ROV.

[2]As noted in §4.1, we operationally equate one IP address with one RP server. Multiple RP instances behind a single IP—or a single RP rotating across IPs—would not be distinguishable in our measurements; the counts here are therefore point-in-time observations that may undercount actual redundancy.

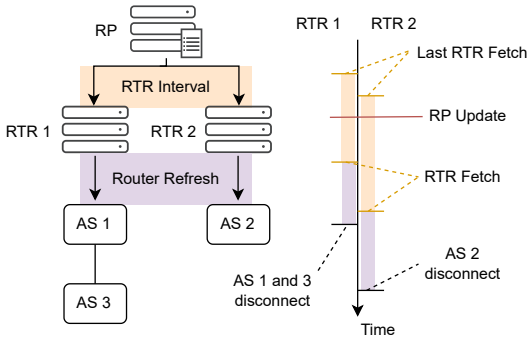

Figure 4: Both AS 1 and AS 2 rely on the same RP. AS 1 and its downstream AS 3 disconnect from the targeted invalid prefix simultaneously, whereas AS 2 filters it at a different time, indicating that AS 3 inherits ROV from AS 1 while AS 2 enforces it independently.

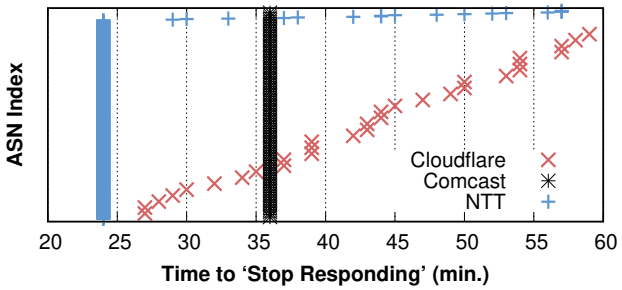

Figure 5: "Stop responding" time for each AS relying on three RPs.

Earlier data-plane methodologies (e.g., [32]) struggled to differentiate between ROV deployment and upstream-protected filtering. Both cases manifest similarly: an AS does not forward traffic to an invalid prefix. However, our dynamic invalidation approach and *timing-based analysis* overcome this limitation.

Timing-Based Insight: Our key insight is that an AS relying on a single upstream filter will be *disconnected* from the invalid prefix *precisely* when that upstream AS starts filtering. By contrast, truly *independent* ROV deployments exhibit differences in *when* each AS discards the invalid route, due to unaligned polling intervals or distinct internal router-update schedules. Figure 4 depicts three ASes (AS1, AS2, AS3) that share a single RP server's data:

- *AS1 and AS2 (ROV Deployment):* They drop the invalid route at slightly different times because each runs its own RTR instance and router stack.
- *AS3 (Upstream-Protected ROV):* It stops forwarding traffic exactly when AS1 filters the route, reflecting that AS1 is a transit provider discarding the prefix.

Figure 5 shows when ASes lost connectivity to our intentionally invalid /48 prefix after we targeted three prominent RPs (NTT, Comcast, Cloudflare), a result we also verified through direct communication with their teams.

- *Comcast:* This private RP is only accessible within Comcast's network. Consequently, *all* of their 94 ASes using it filter the prefix at exactly the same moment. We infer that these ASes are either internal Comcast networks or direct customers of Comcast who inherit upstream filtering.
- *Cloudflare:* Cloudflare operates a well-known publicly available RP [13], which has been used as a default RP source in Cloudflare's GoRTR [12]. ASes that rely on this public RP need to implement ROV locally. Their drop times vary, sometimes overlapping by chance within short observation windows (e.g., one minute); however, across repeated trials, these overlaps rarely persist, indicating independent ROV deployments. We observe 39 ASes using Cloudflare's RP, with 34 (87.2%) never overlapping in their drop times, and none of them are downstream customers of Cloudflare in CAIDA AS relationship dataset [11].
- *NTT:* NTT offers a public RP server yet also has numerous downstreams. As a result, the majority 377 (95.7%) ASes show synchronized drop times (likely downstreams relying on NTT's enforcement), whereas others 17 (4.3%) diverge (i.e., ROV deployment).

These observations illustrate how *timing-based analysis* can distinguish ASes that individually run ROV (their filtering times differ) from those that inherit filtering from a single upstream ROV deployment (they drop simultaneously).

Next, we re-run RScope on the 1,672 RPs previously identified, each associated with at least one ROV-protected AS. Rather than probing each host only once, we now continuously monitor them every minute to detect exactly *when* they stop forwarding the invalid prefix. This step is repeated five times in January 2025 to reduce accidental overlaps in drop times; by capturing the precise "stop-responding" moment, we gain deeper insight into which networks deploy local ROV and which simply inherit it from upstream providers.

Pinpointing the Actual ROV Deployer: Even if a group of ASes share an identical "stop-responding" timestamp, we must still pinpoint which AS actually *deploys* ROV. We follow a three-step process:

(a) *RP Hosting ASN:* if one of those ASes hosts the relevant RP, we assume it is the actual ROV deployer.
(b) *AS Relationship Datasets:* if one AS is an upstream for all others in that group, we mark it as the ROV enforcer; the rest inherit from it.
(c) *Traceroutes from Our Vantage:* in ambiguous cases, we run traceroute from our four vantage points. If all paths to the group's ASes pass through a particular ASN, we conclude that ASN is performing upstream ROV.

By combining these methods, we aim to systematically identify which AS truly deploys ROV and which ASes merely inherit protection, offering a more accurate view of global ROV adoption than data-plane reachability tests alone.

5.3 Quantifying ROV Deployments and Upstream Benefits

After applying the introduced steps, we label 1,127 ASes as actively deploying ROV and find that 1,815 ASes gain ROV protection

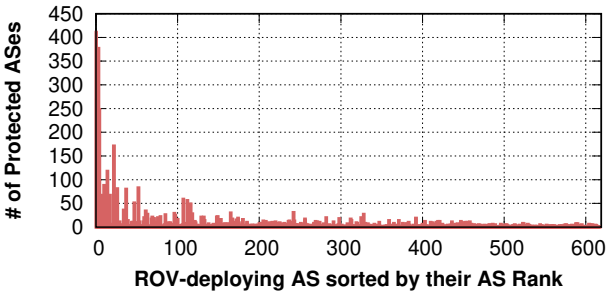

Figure 6: The number of downstream ASes protected by ASes deployed ROV tend to increase as their ASRank rises.

solely benefit from upstream enforcement. In this section, we detail the characteristics of each group and compare our findings with external datasets.

ROV Deployment: Among the 1,127 ASes that *independently* implement ROV, we find that (1) 57.2% consume VRPs from exactly one RP server IP at measurement time[3], (2) 28.6% utilize multiple RP servers, but all within the *same* ASN. (3) 14.2% deploy multiple RP servers across *different* ASNs, presumably as a stronger safeguard against single points of failure.

With CAIDA's `as2org` datasets, we can further find that (1) 66.1% of ROV-enabled ASes host at least one RP server within their own AS, (2) 4.9% host their RP server(s) in a *different* AS that belongs to the same parent organization (e.g., sister companies under one conglomerate), and (3) The remaining 29.0% rely on servers outside their organization, often at cloud providers (e.g., AWS, Vultr) or even in an upstream network.

These patterns highlight that, while a sizable fraction of ASes maintain fully self-managed RPKI validation, many still outsource partially or entirely to third-party hosts, raising questions about operational flexibility versus potential risks (e.g., trust dependencies or commercial lock-in).

Benefits from upstream ROV: We next examine the 1,815 ASes that do not deploy ROV themselves but nonetheless appear to filter invalid routes. In these cases, *another* AS enforces ROV on their behalf, leading to *collateral* (or "inherited") protection.

Although 1,815 ASes with collateral benefits, we find that only 619 ROV-deploying ASes collectively protect them. Table 1 shows the top five "protector" ASes by the number of downstream ASes they shield; large providers, such as Level3 or NTT, can each secure hundreds of customers.

We also use ASRank data [31] to compare each ROV-deploying AS's rank against the number of dependent ASes it protects. Figure 6 suggests a strong correlation: higher-ranked (i.e., larger) ISPs tend to have more downstream ASes relying on them; this finding reinforces the significant global impact when tier-1 or large regional ISPs adopt ROV —they effectively safeguard an extensive subtree of the Internet.

[3]This count reflects observed RP server IPs and does not preclude hidden redundancy such as multiple RP instances behind a single IP or cold-standby servers that were not active during our measurement window.

ASN	Name	AS Rank	Protected ASes
3356	Level 3 Parent, LLC	1	411
2914	NTT America, Inc.	5	377
174	Cogent Communications	3	268
6939	Hurricane Electric LLC	6	248
7018	AT&T Services, Inc.	27	171

Table 1: Top five ASes ranked by the number of downstream ASes they protect.

5.4 Cross-Validation

We compare our results against prominent ROV measurement datasets that predominantly focus on *IPv4*; in principle, an operator could enforce different ROV policies for IPv4 and IPv6, so a perfect one-to-one match may not be guaranteed.

ROV Deployment Datasets: We gathered two public references of ROV-deploying ASes:

- *Manually Curated List* [32], comprising 36 ASes that publicly claim ROV, collected via operational forums, blogs, and announcements.
- *Control-plane Inference* [23], which uses route collector analyses similar to [49]; This snapshot labels 104 ASes as ROV-enforcing.

Combining these yields 113 ASes supposedly deploying ROV; our IPv6-based measurements cover 94 (83.1%) of them, and in *every* case, we also detect ROV deployment in RScope; this perfect overlap (94/94) suggests that our method is accurately identifying ASes that deploy ROV themselves. Among them, 6 ASes are upstreams of our origin ASes discussed in §4.1.

ROV Protection Datasets: We next compare with two well-known *ROV protection* datasets: RoVista [32] and APNIC [43]. We take a single-day snapshot from each, encompassing 4,792 ASes that appear "ROV-protected" by their methodology. We first find that 983 of them (20.5%) are labeled as "deploying ROV " in our experiment, 3,283 (68.5%) appear to rely on upstream-pased ROV, and 526 (11.0%) fall outside our IPv6 vantage or do not respond. We bucket these ASes by ASRank (in ranges of 5,000) and compute the fraction that deploy ROV themselves versus those inheriting upstream filtering; Figure 7 shows a strong correlation with rank: top-tier ASes are far more likely to deploy ROV themselves, whereas smaller ASes rely primarily on upstream filters.

In particular, top 1~5,000 ASes show ~74.4% ROV deployment, whereas ASes ranked below 50,000 are mostly reliant on upstream protection (only 3.7% deploy themselves). Nonetheless, a handful of low-ranked ASes do run ROV themselves, indicating these best practices can be adopted at any scale.

5.5 Shared Relying Parties

Although most networks in our dataset host their own RP servers, a noticeable portion instead (or additionally) rely on *shared* RP infrastructure maintained by third parties. This approach reduces the overhead of deploying and maintaining one's own validation infrastructure. Yet it also raises significant *security and reliability* concerns: if a public RP server is taken offline, compromised, or has its connection disrupted, every AS depending on it loses ROV protection.

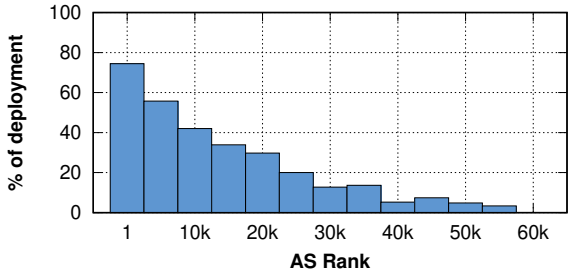

Figure 7: Fraction of ROV-protected ASes that deploy ROV locally, binned by ASRank.

Name	RP URL	ASes	
		All	Exclusive
Cisco	unknown (hosted in AS109)	82	53
OpenBSD	console.rpki-client.org/rpki.json	56	39
Cloudflare	rpki.cloudflare.com/rpki.json	34	17
RIPE	rpki-validator.ripe.net/json	28	22
NTT	rpki.gin.ntt.net/api/export.json	17[2]	13

Table 2: Top five publicly visible RP servers by total AS usage ("All ASes") and by ASes that depend on them exclusively ("Exclusive ASes").

Prevalence: Among the 1,672 RP servers utilized by the 1,127 ASes that deploy ROV, we first examine how many servers serve *multiple* ASes; the vast majority (93.2%) are used exclusively by one AS, but 114 RP servers are shared across multiple ASes, prompting a deeper look at *organizational* boundaries. By applying CAIDA's `as2org` dataset [10], we filter out cases where multiple ASNs belong to the same parent entity, leaving 14 RP servers that genuinely serve more than one organization.

Prominent Public Servers: Table 2 lists the top five RP servers that connect to multiple external ASes. Apart from the Cisco server hosted in AS 109—for which we were unable to locate a publicly advertised VRP endpoint—the remaining servers (OpenBSD, Cloudflare, RIPE, NTT) are well-known public providers with documented APIs or JSON export endpoints. Our manual inspection and logs confirm that these RP servers each serve a variety of external networks, highlighting the degree to which some ASes outsource RPKI validation.

Single RP Reliance: We next identify ASes that rely *solely* on one of these public RP servers, meaning they do not have any other RPKI validation source. In total, 144 ASes fit this description. Such exclusive dependence introduces a distinct *single point of failure*: if the public RP server becomes unavailable or suffers a security breach, these ASes effectively lose their ROV protection. The relatively small but non-trivial fraction of ASes in this situation underscores the trade-off between ease of adoption (outsourcing RPKI validation) and operational resilience.

[2]Because NTT also has downstream ASes, we exclude them from this analysis.

5.6 ROV Deployment Strategies for Multiple RP Servers

RFC guidance recommends configuring multiple RP servers for resilience [4]. In practice, operators integrate and prioritize these servers in several ways. We outline four common strategies and note what RScope can and cannot detect.

One-at-a-Time: The AS actively uses *only one* RP at a time, periodically switching among available RP servers. We found that certain public services (e.g., OpenBSD's public RP infrastructure) introduce this behavior, causing an AS to potentially filter different prefixes at different times. By repeatedly running RScope over two weeks, we observed 73 ROV-enabled ASes that changed which RP they used during the measurement period, most of which were connected to OpenBSD's five-server public RP setup. Because OpenBSD rotates VRP files in a round-robin fashion, these ASes experience fluctuating route validation outcomes. Due to the measurement window and scan interval, we may not capture every RP used under this strategy.

Combine All (Union): An AS *always* merges VRPs from *all* configured RPs. If a prefix is RPKI-invalid according to *any* RP, the route is filtered; for example, some router implementations (e.g., BIRD [6], FRR [17]) adopt this behavior by default when using multiple RTR sources, taking the union of all VRPs to enforce maximum strictness. Among the 1,127 ASes we identified as ROV-deploying ASes, 483 (42.8%) employ multiple RPs under this "union" model.

Primary/Backup Failover: An AS configures multiple RPs but designates only one as "active", switching to a secondary RP *only* if the primary fails or becomes unreachable. Because we do not *induce failures*, we cannot quantify how often this strategy appears; networks that seem to consume VRPs from a single RP may still have cold standbys.

Consensus (Intersection): An AS filters a route *only if* all of its configured RP servers mark the route invalid: any single "clean" verdict effectively overrides conflicting (invalid) ones. This method reduces false positives (e.g., from a single erroneous ROA) but can leave the AS vulnerable if a compromised or incorrect RP labels the route as valid.

Detecting these "consensus" ASes ideally requires testing *every* possible combination of RPs (e.g., feeding identical ROAs to each subset of the 3,516 servers), which is impractical. As a middle ground, we focus on ASes hosting multiple RP servers. Specifically, we group the 3,516 RP servers by the 1,127 ASes in which they reside, feeding each group an invalidating package tied to a unique prefix. Then, we run data-plane measurements (ICMP pings) to see whether an AS drops the prefix *only* after *all* of its local RP servers label it invalid. We discover 205 ASes exhibiting this behavior; of these 205 ASes, 24 are known to deploy ROV (e.g., IIJ), while the remaining 181 do not deploy ROV but rely on an upstream AS that follows this intersection-based strategy.

5.7 RP-to-ROV Latency

In addition to mapping which RPs each AS relies on, we also examine the *timeliness* of route filtering after a ROA change. Previous studies measured the delay between publishing a ROA at an

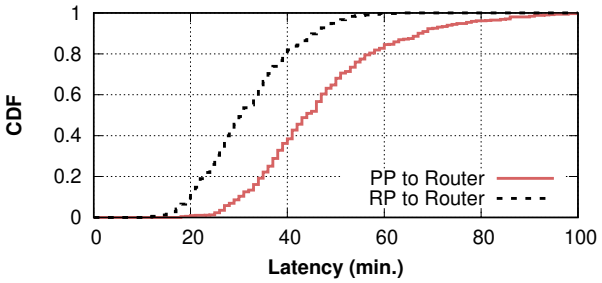

Figure 8: PP-to-Router vs. RP-to-Router latency distributions; the latter averages ~ 37 minutes, highlighting delays *after* the RP has fetched new ROA data.

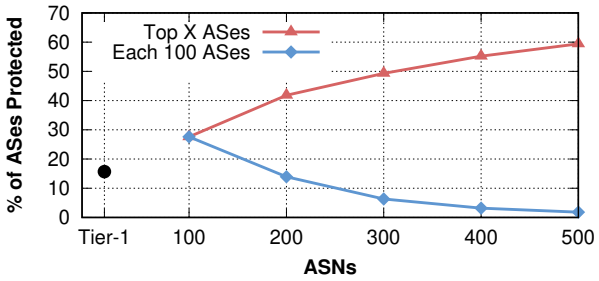

Figure 9: As more higher-ranked ASes deploy ROV, a larger share of the Internet filters the RPKI-invalid prefix; for example, when the top 100 ranked ASes enable ROV, 27.6% of ASes becomes ROV-protected.

RIR's publication point and observing its effect on the data plane (i.e., routes being filtered) [16]. However, accurately capturing the *latency from an RP server to the actual routers* requires knowing precisely which ASes depend on which RP—an aspect that prior work has lacked.

Two-Stage Latency Breakdown. By leveraging our controlled publication point and our *RP–AS* dependency mappings, we quantify the time in two stages:

(a) *ROA Publication to RP Fetch:* The interval between the moment we publish or update a ROA (invalidating package) and the time an RP server completes fetching it from our based publication point.

(b) *RP to ROV Enforcement.* The interval between the RP server's successful retrieval of the updated ROA data and the first observed data-plane evidence that an AS has dropped the invalid route (i.e., "stop responding" time in our ICMP probes).

First, we confirm that our measured *end-to-end* latency (from ROA publication to data-plane filtering) aligns well with prior findings; Figure 8 (red line) shows that most ASes filter invalid prefixes within roughly 60~120 minutes of a ROA update, consistent with earlier reports [16, 22]. An important difference is that our setup bypasses potential delays at RIR-operated publication points (since we publish ROAs on our own server), so the time from "ROA creation" to "ROA availability" is effectively zero in our measurements.

RP-to-Router Delays. Next, we isolate the *second* stage: from when a particular RP server finishes fetching our ROA objects to when its dependent ASes actually start filtering the route. We use NGINX access logs to pinpoint when each RP server completes retrieving the updated files. We then correlate this timestamp with the *stop-responding* time of each AS that relies on that RP. Figure 8 (black line) illustrates that the average delay in this stage can be quite significant—around 37 minutes. This gap reflects: (1) the RP's internal validation cycle (e.g., cryptographic checks, manifest verifications), (2) the time required for RTR daemons to fetch VRPs from the RP server, and (3) additional delays in router software processing or scheduling before actually dropping the invalid route.

Our findings reveal that *even after* an RP obtains fresh ROA data, network operators often wait an additional 30+ minutes before their routers adjust forwarding. This lag reflects a combination of RTR polling delays, operational intervals, software constraints, and perhaps conservative policies meant to avoid route flaps. While this caution may reduce instability from transient ROA errors, it also implies that newly discovered invalid or hijacked routes can remain active for tens of minutes before being blocked.

6 IMPACT OF INCREMENTAL ROV DEPLOYMENT

We now shift from individual ROV-enabled ASes to the Internet-wide implications of *incremental* ROV adoption. In particular, we revisit a core question: *How would the Internet be protected if a selected subset of ASes were to deploy—or disable—ROV?*

Prior studies (e.g., [20]) explored this via simulations to gauge how partial adoption can significantly reduce hijack success rates; however, such models cannot fully capture real routing policies. By contrast, RScope manipulates which ROV-enabled ASes *actually* filter a given prefix, providing empirical insight. One caveat is that our vantage points, while distributed, do not encompass the entire Internet; thus, our observations may be skewed by local routing policies, upstream decisions, or peering arrangements. Despite this limitation, we believe our findings still offer insights into how incremental ROV adoption among large ISPs can influence global routing security.

6.1 Enabling ROV for Selected ASes

We begin with scenarios where only a specific group of ASes enforce ROV. Concretely, we select various subsets of the *1,127 ASes* that independently deploy ROV (Section 5):

- *Tier-1 ASes only*: 13 ROV ASes covered by RScope out of 16 tier-1 ASes [50].
- *Top N ASes by rank:* top 100, 200, 300, 400, 500 among those that deploy ROV, based on CAIDA's ASRank.
- *Ranked slices*: ASes ranked 1–100, then 101–200, etc.

To avoid skew from our vantage points' *own* upstreams, we announce each scenario's invalid prefix from four existing vantage points plus two additional nodes on the Peering Testbed (AMS-IX, Seattle-IX).

For each scenario, we enable invalidation by serving the Invalidation Package *only* to the RP servers of the chosen set of ASes, and all other RPs receive a Neutral Package, leaving the prefix as

ROV Disabled at	Reachability (%)
None (All remain enabled)	0%
Direct Upstreams Only	3.7%
+ Tier 1 ISPs	23.8%
+ Tier 1 ISPs + Top 100 ASes	37.7%

Table 3: As major networks disable ROV, more of the Internet can reach our invalid prefix.

RPKI-Unknown. Then, after waiting for three hours to let our ROA packages propagated, we perform ICMP-based probing from six vantage points and record the fraction of IPs that *stop responding*.

Observations. Figure 9 illustrates the percentage of responsive hosts that become unreachable under different subsets.

We first find that 13 Tier-1 ASes alone filtering our prefix already protect 15.7% of the targets, underscoring the significant global influence of tier-1 ISPs. Top 100 ROV-enabled ASes filtering yield 27.6% coverage, while including the top 500 ASes lifts this coverage to 59.4%. Lower-ranked ASes have comparatively smaller individual impact but can compound existing coverage when added to a larger group; for example, adding ranks 401–500 *after* the top 400 ASes filters another 4.2% of targets, emphasizing that the effect of smaller ASes depends on whether higher-ranked ASes already filter.

6.2 Disabling ROV for Selected ASes

We next assess the flip side: *What if certain major ROV ASes disable their filtering?* This simulates RPKI outages or conscious policy downgrades, which can expose the Internet to hijacks when downstream ASes rely on that upstream filter. To this end, we perform experiments as follows:

(a) First, we publish an Invalidation Package to *all known* ROV-enabled ASes, ensuring our prefix is globally seen as invalid. We confirm total drop of the prefix, as none of our vantage points receive responses from the ping targets (i.e., 100% drop).

(b) We then *remove* the Invalidation Package from selected subsets of ASes, making them treat our prefix as Unknown (or valid). This simulates them "disabling" ROV.

(c) We repeat the data-plane probing to gauge how many targets become reachable once these ASes no longer filter the prefix.

Observation. Table 3 summarizes the fraction of ping targets that regain connectivity under different scenarios:

- Removing invalidation from the direct upstreams of our vantage points has a modest effect (prefix still blocked for 96.3% of hosts).
- Disabling ROV at tier-1 ISPs, however, restores reachability of RPKI-invalid prefixes for 23.8% of the Internet.
- Extending that removal to the top 100 ASes raises it to 37.7%, illustrating how large ISPs can significantly affect global hijack protection.

These "disable" experiments confirm that the presence (or absence) of a *small subset* of powerful ROV enforcers substantially alters global routing security.

6.3 Vulnerability to "Local" Hijacks

Although networks that do not deploy ROV locally can gain protection from ROV-enabled upstreams or peers—particularly smaller ASes, as shown in §5.3 and §5.4—such coverage may fail under a *local* hijack that remains within a cluster of non-ROV ASes. In these scenarios, traffic never passes through any upstream enforcer.

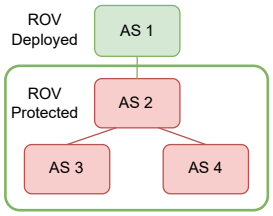

Figure 10: Even though AS 3 is "protected" by AS 1, it is still vulnerable to a local hijack from AS 4, due to the lack of ROV in AS 2.

Figure 10 shows an example in which AS 1 enforces ROV for downstream ASes 2, 3, and 4, yet AS 4 can launch an RPKI-invalid announcement that AS 2 and AS 3 do not filter. Because the route between AS 3 and AS 4 never traverses AS 1, AS 3 becomes hijackable by AS 4.

Methodology. We begin by constructing an AS-level topology from CAIDA's AS relationship dataset [11] containing 76,644 ASes, then applying a standard valley-free model [18, 19] to label forwarding paths. Although these methods may not perfectly reflect real-world routing on the Internet, we believe this widely used method [20, 35] can still provide a valuable measurement. Next, we classify ASes into three groups based on prior data-plane ROV measurements (RoVista [32] and APNIC [43]; see §5.4) and RScope:

(a) *Non-ROV ASes:* ASes for which both RoVista's ROV Score [32] and APNIC's I-RoV metric [43] are zero, indicating no evidence of ROV protection. This group contains 37,183 ASes.

(b) *Upstream-Protected ASes:* ASes that do not deploy ROV but are *protected* by a transit provider's ROV enforcement; combining results from RScope and protection measurements [32, 43], we identify 3,283 such ASes.

(c) *Self-Deploying ASes:* 1,127 ASes confirmed to run ROV locally from our measurements. For any AS not appearing in prior datasets or our own results, we assume it deploys ROV as well, ensuring a conservative analysis that does not overstate potential vulnerabilities.

Our goal is to see whether any of the 3,283 upstream-protected ASes could be reached by a RPKI-invalid announcement entirely within the *non-ROV portion* of the topology. Concretely, we look for any path *fully contained* in the combined set of 40,465 (37,183 + 3,282) non-ROV ASes; if such a path exists, the target AS can be hijacked *without* traversing an ROV filter. We exclude attacks originating from a direct transit provider, since a malicious upstream can trivially perform hijacking or man-in-the-middle with or without ROV.

Observation. Figure 11 illustrates how many non-ROV ASes could potentially launch a local hijack against each upstream-protected AS. While 30.4% of these ASes face no threat of local hijacking (i.e., no path is fully contained within non-ROV ASes), 32.1% remain

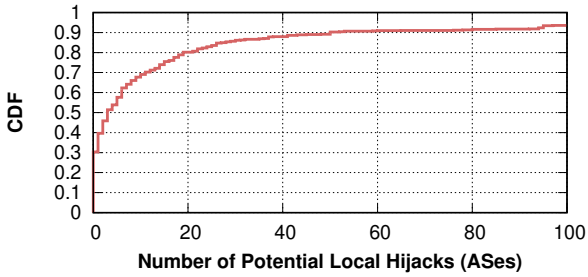

Figure 11: Number of potential non-ROV ASes that can *locally* hijack each upstream-protected AS. Over 30% face threats from more than 10 ASes.

vulnerable to attacks from more than 10 non-ROV ASes. These findings show that relying solely on an upstream's ROV enforcement can block many large-scale hijacks yet leaves networks open to "lateral" attacks that bypass a global ROV filter. In the end, deploying ROV locally offers more comprehensive protection than depending exclusively on upstream filtering.

7 DISCUSSION AND LIMITATIONS

In this section, we highlight main challenges and limitations in our methodology and outline potential directions for future work.

7.1 Measuring ROV in IXPs

Our main technique requires *responsive hosts* within a network to detect whether invalid prefixes are filtered; this prerequisite becomes problematic for IXPs, as they do not typically originate IP addresses for end hosts that can respond to pings or other probes. As a result, our RP-mapping approach does not readily apply to IXPs.

Alternative Approach via Vantage Points. Placing a vantage point *inside* an IXP allows one to selectively feed invalidating ROAs and see whether the IXP filters or permits traffic. In our platform, we partially tested this idea using a Peering Testbed vantage point at AMS-IX and Seattle-IX. First, we split 3,516 RP servers into 256 groups and applied our methodology to identify which RP servers each IXP relies on, then ran an additional round to pinpoint individual servers; we found *both AMS-IX and Seattle-IX enforce ROV* with two RP servers each. While promising, this method is constrained to IXPs where vantage points are available. leaving a global IXP-level ROV assessment as future work.

7.2 IPv6 and Measurement Scaling

Although most prior ROV research (e.g., [26, 32]) has centered on IPv4, IPv6 is increasingly crucial for Internet growth, and new initiatives (e.g., JPNIC [27]) now include IPv6-based testing. Our decision to rely on IPv6 prefixes stems from both practical and methodological considerations: while a minority of networks may enforce ROV inconsistently across address families, in practice most ROV deployments appear to cover both IPv4 and IPv6 similarly [33, 37]. To validate parity, we also ran RScope using four IPv4 /24 prefixes against the top 100 ASes that our IPv6 measurements labeled as ROV-deployed and found that all 100 also enforced ROV in IPv4; limited IPv4 space prevented extending this check to all ASes. Still, a future global study might apply RScope using multiple IPv4 /24 blocks, although IPv4 address scarcity and inherent time overhead pose challenges to scaling. Overall, we believe RScope advances the ability to measure real ROV deployments, yet measuring IXPs without responsive hosts and guaranteeing parity between IPv4 and IPv6 remain key hurdles for future work.

7.3 Limitations

Limitations of AS-Relationship Datasets Our analysis of AS relationships and AS-to-organization mappings in §5 relies on CAIDA datasets [10, 11]. As with any inferred Internet-wide dataset, these mappings may be incomplete, inaccurate, or outdated, which could affect the precision of our results on inter-AS relationships and organizational affiliations. Incorporating newer alternatives [1, 51] may further improve the accuracy of this part of the analysis.

Multiple RP Servers Behind a Single IP In §5, we analyze the number of RP servers used by each AS and identify ASes that appear to rely on only a single RP server. However, there may be corner cases in which an operator deploys multiple RP servers behind a single IP address. In such cases, our methodology would observe only one visible RP endpoint even though some internal redundancy exists. At the same time, dependence on a single visible IP may still introduce a single point of failure at some level. We leave it to future work to better understand how common such deployments are and how they affect the robustness of ROV.

Deployments with Consensus Requirements As discussed in §5.6, some ASes may require consensus across multiple RP servers before accepting a VRP for ROV. Detecting such deployments requires testing different combinations of RP servers and observing the resulting connectivity changes. However, given the large number of RP servers and the combinatorial explosion of possible subsets, exhaustively exploring all combinations is impractical within a reasonable time frame. In this work, we therefore approximate this analysis by testing combinations formed by all RP servers hosted within the same AS.

8 CONCLUSION

In this paper, we introduced RScope, a new methodology for measuring ROV dependencies across the global Internet. By distributing different ROAs to different RPs, RScope enables us to map which ASes depend on which RP at a large scale. Using RScope, we successfully identify 1,127 ASes that actively deploy ROV out of 21,827 examined ASes, achieving *100%* agreement with public data sources.

Our measurements reveal that although ROV protection is expanding, many ASes rely on collateral benefit from upstream transit providers for filtering RPKI-invalid prefixes, leaving them vulnerable to local hijacks. We also find that a substantial fraction of ROV-enforcing ASes outsource validation to third-party RP infrastructure or were observed consuming VRPs from a single RP server IP during our measurements, highlighting operational dependencies that merit further attention.

As ROV transitions from a niche security measure to a widely adopted best practice, our findings clarify real-world deployment

patterns and the dependencies between major ROV networks and their RP servers.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. We are grateful to Ties de Kock for technical discussions, and to Tony Tauber (Comcast) for operational insights and validation support. We are grateful to Italo Cunha and Ethan Katz-Bassett for supporting us to use the PEERING testbed. This research was supported in part by NSF grants CNS-2339378 and CNS-2323137, and the Comcast Innovation Fund.

REFERENCES

[1] A. Arturi, E. Carisimo, and F. E. Bustamante. as2org+: Enriching as-to-Organization Mappings with PeeringDB. *PAM*, 2023.
[2] H. Ballani, P. Francis, and X. Zhang. A study of prefix hijacking and interception in the internet. *SIGCOMM*, 2007.
[3] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford. A survey of BGP security issues and solutions. *Proceedings of the IEEE*, 98(1), IEEE, 2010.
[4] R. Bush. Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI). RFC 7115, IETF, 2014.
[5] M. A. Brown. Pakistan hijacks YouTube. 2008. https://dyn.com/blog/pakistan-hijacks-youtube-1/.
[6] BGP Routing Daemons with RPKI/RTR. https://rtrlib.readthedocs.io/en/latest/bgprd.html.
[7] J. Cowie. China's 18-Minute Mystery. 2010. https://dyn.com/blog/chinas-18-minute-mystery/.
[8] W. Chen, Z. Wang, D. Han, C. Duan, X. Yin, J. Yang, and X. Shi. ROV-MI: Large-Scale, Accurate and Efficient Measurement of ROV Deployment. *NDSS*, 2022.
[9] B. Cartwright-Cox. Measuring RPKI Adoption via the data-plane. NLNOG Day 2018. https://nlnog.net/static/nlnogday2018/8_Measuring_RPKI_ben_NLNOG_2018.pdf.
[10] CAIDA ASOrganizations Dataset. http://www.caida.org/data/as-organizations/.
[11] CAIDA ASRelationships Dataset. http://www.caida.org/data/as-relationships/.
[12] GoRTR. https://github.com/cloudflare/gortr.
[13] Cloudflare's RPKI Toolkit. https://rpki.cloudflare.com/rpki.json.
[14] D. Dittrich and E. Kenneally. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. 2012. https://www.dhs.gov/sites/default/files/publications/CSD-MenloPrinciplesCORE-20120803_1.pdf.
[15] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-Wide Scanning and its Security Applications. *USENIX Security*, 2013.
[16] R. Fontugne, A. Phokeer, C. Pelsser, K. Vermeulen, and R. Bush. RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes. *PAM*, 2023.
[17] FRR - Prefix Origin Validation Using RPKI. https://github.com/FRRouting/frr/blob/master/doc/user/rpki.rst.
[18] P. Gill, M. Schapira, and S. Goldberg. Let the market drive deployment: A strategy for transitioning to BGP security. *ACM SIGCOMM computer communication review*, 41(4), ACM New York, NY, USA, 2011.
[19] P. Gill, M. Schapira, and Goldberg. Modeling on quicksand: dealing with the scarcity of ground truth in interdomain routing data. *ACM SIGCOMM Computer Communication Review*, 42(1), ACM New York, NY, USA, 2012.
[20] Y. Gilad, A. Cohen, A. Herzberg, M. Schapira, and H. Shulman. Are We There Yet? On RPKI's Deployment and Security. *NDSS*, 2017.
[21] G. Huston and J. Damas. Measuring Route Origin Validation. 2020. https://www.potaroo.net/ispcol/2020-06/rov.html.
[22] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman, and M. Waidner. Behind the scenes of RPKI. *CCS*, 2022.
[23] T. Hlavacek, H. Shulman, and M. Waidner. "Keep Your Friends Close, but Your Routeservers Closer: Insights into RPKI Validation in the Internet. *USENIX Security*, 2023.
[24] IPv6 CIDR REPORT. https://www.cidr-report.org/v6/as2.0/.
[25] IPv6 Histlist Service. https://ipv6hitlist.github.io/.
[26] Is BGP Safe Yet? https://isbgpsafeyet.com/.
[27] JPNIC. JPNIC rov-checkt. https://rov-check.nic.ad.jp/.
[28] J. Kristoff, R. Bush, C. Kanich, G. Michaelson, A. Phokeer, T. C. Schmidt, and M. Wahlisch. OnMeasuring RPKI Relying Parties. *IMC*, 2020.
[29] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, IETF, 2012.
[30] M. Lepinski, S. Kent, G. Huston, and R. Austein. Manifests for the Resource Public Key Infrastructure (RPKI). RFC 6486, IETF, 2012.
[31] M. Luckie, B. Huffaker, K. Claffy, A. Dhamdhere, and V. Giotsas. AS Relationships, Customer Cones, and Validation. *IMC*, 2013.
[32] W. Li, Z. Lin, M. I. A. Khan, E. Aben, R. Fontugne, A. Phokeer, and T. Chung. RoVista: Measuring and Understanding the Route Origin Validation (ROV) in RPKI. *IMC*, 2023.
[33] D. Madory. RPKI ROV Deployment Reaches Major Milestone. https://manrs.org/2024/05/rpki-rov-deployment-reaches-major-milestone/.
[34] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein. BGP Prefix Origin Validation. RFC 6811, IETF, 2013.
[35] R. Morillo, J. Furuness, C. Morris, J. Breslin, A. Herzberg, and B. Wang. ROV++: Improved Deployable Defense against BGP Hijacking. *NDSS*, 2021.
[36] NTT RPKI VRP. https://rpki.gin.ntt.net/api/export.json.
[37] W. Ohgai. Is your network protected? The rov-check project by JPNIC. 2025. https://blog.apnic.net/2025/03/25/is-your-network-protected-the-rov-check-project-by-jpnic/.
[38] OpenBSP RPKI-Client VRP. https://console.rpki-client.org/rpki.json.
[39] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt, and M. Whlisch. Towards a Rigorous Methodology for Measuring Adoption of RPKI Route Validation and Filtering. *CCR*, 48(1), 2018.
[40] N. Rodday, I. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt, and M. Wählisch. Revisiting RPKI Route Origin Validation on the Data Plane. *TMA*, 2021.
[41] RIPE Historic RPKI Data Archive. https://ftp.ripe.net/rpki/.
[42] RIPE Routing Information Service (RIS). http://www.ripe.net/projects/ris/rawdata.html.
[43] RPKI I-Rov Filtering Rate. https://stats.labs.apnic.net.
[44] RPKI Validator API. https://rpki-validator.ripe.net/api/export.json.
[45] RIPE. RPKI Test. https://www.ripe.net/s/rpki-test/.
[46] RoVista. http://rovista.netsecurelab.org/.
[47] University of Oregon RouteViews project. http://www.routeviews.org/.
[48] B. Tim, M. Oleg, W. Bryan, and A. Rob. The RPKI repository delta protocol (RRDP). RFC 8182, IETF, 2017.

[49] H. Tomas, H. Amir, S. Haya, and W. Michael. Practical experience: Methodologies for measuring route origin validation. *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018.
[50] Wikipedia - Tier 1 network. https://en.wikipedia.org/wiki/Tier_1_network.
[51] Y. Xu, W. Li, E. Umrani, and T. Chung. ASINT: Learning as-to-Organization Mapping from Internet Metadata (Under Review). *WWW*, 2026.

A ETHICS

Our work involves controlled manipulation of RPKI objects and data-plane probing, both of which raise potential ethical concerns. In this section, we explain our steps to ensure that our methods are responsible and compliant with community standards.

A.1 Controlled RPKI Publication Point and ROAs

RScope operates a dedicated RPKI publication point and generates specialized ROAs for RP servers connecting to our repository. We configure the publication point with valid RPKI objects and a legitimate HTTPS certificate, ensuring that neither neutral nor invalidating packages cause any disruption for RPs in the global ecosystem. Our NGINX server redirects RPs to the appropriate ROA objects (i.e., the neutral or invalidation package), adding minimal latency overhead.

We use at most one invalidation package per RP at any time. Considering that there are over 300k ROAs in today's RPKI [38], the additional load on validation processes is negligible. We announced the details of our publication point to the broader RPKI community before commencing measurements and received no concerns. All IP prefixes and AS numbers utilized in our study are allocated for research purposes only; we have explicit authorization to use two different ASNs—one originates the announced prefix, and the other appears in the invalidation package. Consequently, the RPKI-invalid status we induce does not affect external networks.

A.2 Data-Plane Probing

RScope also involves large-scale data-plane probing to detect whether certain prefixes are filtered. Specifically, §4.3 describes how we send ICMP echo requests to 51,075 IPv6 hosts. We adhere to ethical scanning guidelines [15] and the Menlo Report principles [14] by:

- Sending only minimal ICMP echo packets (no payload),
- Limiting each IP address to at most 60 ICMP packets per day, spaced at a minimum rate of one per minute,
- Selecting a maximum of five responsive IPv6 addresses per AS from an existing hitlist instead of brute-forcing the address space,
- Coordinating in advance with local network administrators to handle potential inquiries.

We also restrict our scanning to 10 Mbps to avoid saturating upstream providers and generate only the traffic necessary to meet our measurement goals. These steps minimize potential risks or disruptions to the broader Internet, reflecting our commitment to conducting responsible research.