

# A First Look at ASPA: Deployment, Route Leaks, and AS Relationships

Mauro Farina  
mauro.farina@phd.units.it  
University of Trieste  
Trieste, Italy

Martino Trevisan  
martino.trevisan@dia.units.it  
University of Trieste  
Trieste, Italy

Alberto Bartoli  
bartoli.alberto@units.it  
University of Trieste  
Trieste, Italy

## Abstract

The Border Gateway Protocol (BGP) lacks native mechanisms to verify that routes adhere to authorized paths. This makes the global infrastructure vulnerable to route leaks, which have caused several major Internet disruptions. Autonomous System Provider Authorization (ASPA) is an emerging technology, currently at Internet-draft stage, that extends the Resource Public Key Infrastructure (RPKI) by allowing ASes to cryptographically declare their upstream providers; prior work has shown its effectiveness in detecting route leaks and certain forms of path manipulation. Yet, as ASPA moves toward standardization and Regional Internet Registries enable it in their RPKI dashboards, its actual deployment, operational dynamics, and alignment with independently inferred relationships remain unstudied. In this paper, we present the first measurement study of real-world ASPA deployment, tracing the evolution of 1646 adopting ASes from October 2023 to March 2026. We further introduce a methodology to apply ASPA AS Path verification—originally designed for live BGP routers—to passively collected BGP data, and use it to identify thousands of potential route leaks across nearly 57 million AS Paths. Finally, we compare ASPA-declared providers against two publicly available datasets of inferred providers, and find a recall up to 93 % when accounting for our limited visibility into the routing ecosystem.

## CCS Concepts

• **Networks** → **Routing protocols**; *Network measurement*; • **Security and privacy** → *Security protocols*.

## Keywords

ASPA, Autonomous Systems, RPKI, BGP

### ACM Reference Format:

Mauro Farina, Martino Trevisan, and Alberto Bartoli. 2026. A First Look at ASPA: Deployment, Route Leaks, and AS Relationships. In *Proceedings of the 2026 ACM Internet Measurement Conference (IMC '26)*, October 12–16, 2026, Karlsruhe, Germany. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3777912.3839812>

## 1 Introduction

The Border Gateway Protocol (BGP) is the foundation of the Internet, as it allows Autonomous Systems (ASes) to announce reachability to IP prefixes and share routes. However, it provides no built-in

mechanisms to verify the authenticity of route announcements or to ensure that the propagation of routes respects the routing policies of the ASes along the path, leaving the Internet routing ecosystem vulnerable to several threats.

The Resource Public Key Infrastructure (RPKI) [32] was introduced to address one such vulnerability: route origin hijacks, in which an AS announces IP prefixes it does not own. By cryptographically binding IP prefixes to their authorized origin ASes via Route Origin Authorizations (ROAs) [56], RPKI enables routers to filter out announcements with unauthorized origin ASes.

Path-level security, however, has remained an open problem. A *route leak* is the propagation of a routing announcement beyond the scope defined by the commercial agreements between two ASes, commonly known as AS Relationships [20, 59]. Because ASes are economically incentivized to prefer routes learned from customers, a single leak can propagate widely, diverting traffic through unintended paths and causing congestion or black holes. Such incidents most often arise from accidental misconfigurations—though deliberate leaks are also possible [59]—and have been responsible for significant Internet disruptions [27, 35, 45, 51, 60].

Several proposals have aimed to address this path-level security gap. The Mutually Agreed Norms for Routing Security (MANRS) initiative has long encouraged network operators to register their routing policies in Internet Routing Registries (IRRs) and to construct filters from IRR data to limit the propagation of route leaks [36]; however, IRR records are often stale, lack reliable authentication [15, 16], and rely on a complex language that is difficult to interpret [26]. More structured defenses exist: Peerlock is often deployed by large Internet carriers [39] but relies on laborious manual coordination via out-of-band channels [3, 39], whereas the OTC attribute [4] mitigates only accidental route leaks but not deliberate policy violations. Furthermore, the research community has proposed a range of detection-oriented methodologies [5, 24, 25, 47, 52, 61, 64].

Autonomous System Provider Authorization (ASPA) [3, 55] is a recent IETF proposal for a RPKI-based route leak detection mechanism. An AS issues an ASPA object to cryptographically declare its set of authorized upstream providers; routers can then verify each hop in an incoming AS Path against these declarations, detecting not only route leaks—including those arising from misconfigurations—but also certain forms of path manipulation [3, 19]. Despite remaining an Internet Draft, ASPA has already attracted significant operational interest: more than 1600 ASes have published ASPA objects in the RPKI, RIPE NCC and ARIN have integrated ASPA into their RPKI dashboards [38, 62], and all five Regional Internet Registries (RIRs) have committed to supporting ASPA by the end of 2026 [58]. While support in open-source implementations arrived as early as 2023, general availability in commercial off-the-shelf



This work is licensed under a Creative Commons Attribution 4.0 International License. *IMC '26, Karlsruhe, Germany*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2327-8/26/10

<https://doi.org/10.1145/3777912.3839812>

routers is expected by 2026, with large ISP deployment to follow by 2027 [37]. Prior work has examined ASPA's theoretical effectiveness and protocol properties [19]; its real-world deployment and its relationship to existing AS relationship datasets, however, remain largely uncharacterized.

This paper presents the first empirical measurement study of ASPA deployment. Our main contributions are:

- A longitudinal analysis of ASPA deployment covering 1646 ASes that published ASPA objects in the RPKI between October 2023 and March 2026, characterizing adoption behavior over time (Section 3).
- A methodology for applying the new ASPA AS Path verification algorithm, which relies on AS relationship information only available to a BGP router, on publicly available datasets of AS Paths where such information is missing (Section 4).
- A comparison of ASPA-declared providers against AS relationships inferred by CAIDA and BGPKIT, assessing inference accuracy and characterizing discrepancies (Section 5).
- An overview of the currently available data sources for research related to ASPA (Section 6).

We release the data and code to reproduce our analyses on a GitHub repository [18].

## 2 Background

The Border Gateway Protocol (BGP) is the standard inter-domain routing protocol. With BGP, Autonomous Systems (ASes) share routes to IP prefixes in the form of *AS Paths*, sequences of Autonomous System Numbers (ASNs) that indicate the path IP packets should follow to reach hosts within the given IP prefix.<sup>1</sup> Routing policies between ASes are typically governed by commercial agreements known as *AS relationships* [22]. The two most common relationships are *customer-to-provider* (c2p), where a customer AS pays a provider AS for transit to the rest of the Internet, and *peer-to-peer* (p2p), where two ASes agree to exchange traffic between them and their respective customers at no cost [20]. ASes typically export routes learned from their customers to all their neighbors (providers, peers, and other customers), as this increases reachability and potential revenue. In contrast, to avoid providing unpaid transit, routes learned from providers or peers are typically exported only to customers. Paths that follow such routing policies exhibit the well-known *valley-free* property [20]: they consist of an *up-ramp* of zero or more customer-to-provider links, an optional peer-to-peer link, and a *down-ramp* of provider-to-customer links (see Figure 7).

Over the years, Internet Exchange Points (IXPs) have reshaped the Internet routing ecosystem towards a less hierarchical structure by providing shared physical facilities at which ASes can interconnect with one another, significantly reducing the cost and complexity of establishing new connections [10]. Many IXPs further simplify peer-to-peer connections by operating *route servers* (RSes): shared BGP speakers that enable multilateral peering, where an *RS-client* AS exchanges routes with all other RS-client ASes through a single session, rather than by establishing individual bilateral peer-to-peer sessions with each. Because RSes are used only to

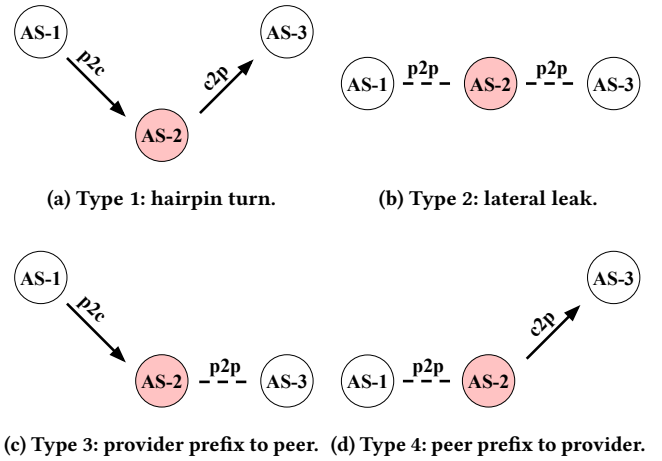


**Figure 1: Four types of route leaks as described in RFC 7908 [59]. The leaking AS is AS-2 in all four cases.**

share routes, not to forward Internet traffic, they typically operate transparently (i.e., do not alter well-known BGP attributes) [29]. However, some RSes work in a *non-transparent* mode, in which the RS modifies BGP attributes, for example, by inserting its own AS number into the AS Path.

A *route leak* occurs when an AS exports a route in violation of the valley-free property [59]. RFC 7908 identifies four such violation types (see Figure 1)<sup>2</sup>: (*Type 1*) a route learned from a provider is exported to another provider; (*Type 2*) a route learned from a peer is exported to another peer; (*Type 3*) a route learned from a provider is exported to a peer; and (*Type 4*) a route learned from a peer is exported to a provider. Because ASes prefer routes received from customers over those from peers or providers, route leaks can propagate and divert traffic through unintended paths, enabling eavesdropping and traffic analysis or causing traffic congestion. Although these incidents often arise from unintentional misconfigurations rather than deliberate attacks, they have been responsible for several major Internet incidents [27, 35, 45, 51, 60]. Yet, BGP provides no built-in mechanisms to detect valley-free violations along a path.

Autonomous System Provider Authorization (ASPA) [3, 55] was specifically designed to address this issue. An ASPA is a cryptographically signed object in which an AS declares its set of authorized providers, as well as non-transparent Route Servers where the AS is an RS-client. ASPA objects are expected to be complete: an AS must list all ASes that may act as upstream providers, including those providing transit for only a subset of prefixes (e.g., only IPv4 or IPv6 prefixes) or those serving as backup transit links.

ASPA objects are hosted in the Resource Public Key Infrastructure (RPKI) [32], a hierarchical Public Key Infrastructure that binds Internet Number Resources such as ASNs and IP prefixes to public keys via certificates. Each Regional Internet Registry (RIR) acts as a trust anchor and offers RPKI repository services under three deployment models, providing resource holders—the organizations

<sup>1</sup>An IP prefix is a contiguous block of IP addresses defined by a network address and a prefix length (e.g., 192.0.2.0/24), effectively denoting an IP subnet.

<sup>2</sup>RFC 7908 further details two route leak types that, however, appear as tampering of the originating AS and are thus detected through Route Origin Validation (ROV).

that own and run ASes—with different levels of control over their RPKI objects. In *hosted* repositories, the RIR operates both the Certificate Authority (CA) and the publication infrastructure on behalf of the resource holder. In *delegated* repositories, the resource holder operates its own CA and publication infrastructure [14]. In *hybrid* repositories, the resource holder operates its own CA while publishing the resulting objects through the RIR’s publication infrastructure [49]. Network operators fetch and cryptographically validate ASPA objects from the RPKI to produce *Validated ASPA Payloads* (VAPs): verified mappings of customer ASes to their authorized providers. VAPs are then distributed to BGP routers, which use them to run the new ASPA AS Path verification procedure [3] and detect valley-free violations in incoming AS Paths.

Beyond route leaks, ASPA also provides partial protection against path manipulation attacks [3, 19]. While ROA [56] and ROV [41] defend against straightforward prefix hijacks, they can be bypassed by adversaries forging paths that include a legitimate origin AS. By validating upstream relationships along the AS path, ASPA can detect such attacks, offering an additional layer of defense against prefix hijacks.

In this paper, we provide a first longitudinal view of ASPA deployment and apply the new AS Path verification procedure to publicly available AS Paths to detect route leaks in the wild. Throughout, we adopt the following terminology:

- **ASPA**: the underlying protocol;
- **ASPA Object**: the cryptographically signed object published in the RPKI containing the list of authorized providers;
- **Validated ASPA Payload (VAP)**: the verified mapping of a Customer AS to its authorized providers, produced by validating an ASPA object;
- **Customer AS**: an AS that holds a cryptographically valid ASPA Object in the RPKI; unless otherwise stated, Customer ASes are relative to a specific day or observation period.

Since a VAP is produced if and only if the underlying ASPA Object passes cryptographic validation, the two are observationally equivalent: a Customer AS *holds a valid ASPA Object* if and only if a corresponding VAP is present in the RPKI. We use *valid ASPA Object* to emphasize the operator’s declaration of authorized providers, and *VAP* to refer to the data artifact used for AS Path verification.

### 3 ASPA adoption

This section examines the evolution of ASPA deployment over time, including the dynamicity of VAP changes, and characterizes the 1505 Customer ASes as of March 31st, 2026. We retrieve VAPs from the RIPE NCC RPKI repository archive [42], which provides daily snapshots of validated payloads dating back to October 2023; we further discuss current data sources for ASPA objects and VAPs in Section 6.

#### 3.1 Temporal evolution

First, we characterize how ASPA adoption has evolved. Figure 2 shows the number of Customer ASes from each Regional Internet Registry (RIR) RPKI repository (including those repositories they delegated) over time. The time series starts from October 10th, 2023, when RIPE NCC began including ASPA objects and VAPs in Routinator’s dumps: only 1 ASPA object was present in the RPKI

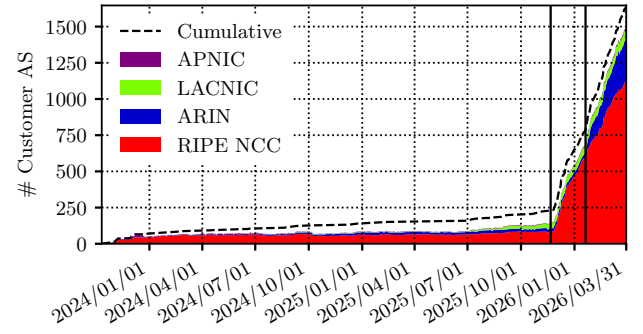


**Figure 2: Time series of the daily number of Customer ASes found in each RIR’s RPKI repository, along with the cumulative number of Customer ASes (dashed black line).**

on that day (AS15562). The figure shows an initial increase in valid ASPA objects from 1 (October 2023) to 60 (February 2024); the growth then slows, reaching 90 only in July 2025—more than a year later. Among the Customer ASes that first adopted ASPA in this early stage, roughly 50 % maintained a valid ASPA object in the RPKI until the end of March 31st. From the time series, we observe two key milestones of ASPA deployment, highlighted by the two vertical solid lines. First, on November 26th, 2025, RIPE NCC officially enabled ASPA in its RPKI Dashboard [62], initiating the ramp that took the number of valid ASPA objects from 152 to 1505 (March 31st, 2026). Second, on January 20th, 2026, ARIN officially enabled ASPA in its RPKI Dashboard [38]. As of March 31st, 2026, there are 1135 Customer ASes from RIPE NCC, 288 from ARIN, 72 from LACNIC, 10 from APNIC, and none from AFRINIC. The sharp increase in adoption following dashboard support from RIPE NCC and ARIN underscores the role of RIRs in facilitating deployment, as their online portals allow operators to create and publish ASPA objects through simple interfaces [40]. Lastly, the black dashed line shows the cumulative number of unique Customer ASes for which we observed a valid ASPA object over time. Since October 2023, we observed a total of 1646 Customer ASes. The discrepancy between this cumulative line and the daily Customer AS count suggests that many ASPA objects were created for testing and were revoked or withdrawn from the RPKI shortly after.

The distribution of valid ASPA objects across RPKI repositories is consistent with the illustrated timeline. On November 25th, 2025, before any RIR enabled ASPA in their dashboard, all ASPA objects were located in delegated (72) or hybrid (76) repositories. This is expected, as object creation in such repositories is performed through the network operator’s own CA and does not require RIR support to create ASPA objects. Among delegated repositories, the one operated by the Brazilian Internet Registry hosted the most ASPA objects (36). By March 31, 2026, instead, ASPA objects in hosted repositories (1290) significantly outnumbered those in delegated (106) or hybrid (109).

We next look into the dynamicity of ASPA objects along four RPKI event types: *removal*, when a Customer AS revokes or withdraws its ASPA object from the RPKI; *re-pub*, when a Customer AS publishes a new ASPA object following a prior removal; *update*,

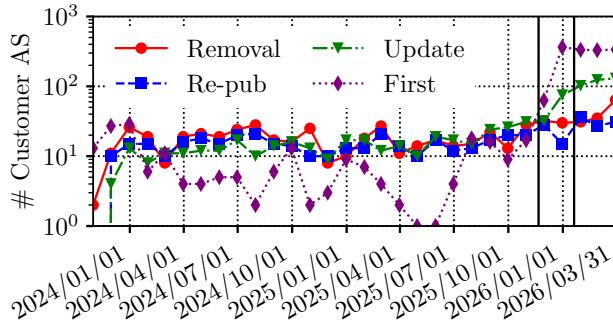


**Figure 3: Number of Customer ASes per month that remove, re-publish, update, or publish a valid ASPA object for the first time.**

when a Customer AS changes the list of its authorized providers; and *first* publications, when a Customer AS publishes a valid ASPA object for the first time.

Figure 3 shows, for each month, the number of Customer ASes contributing to each of the four event types. Note that the four events are not mutually exclusive: an AS can contribute to none, all, or any combination of them within a given month. Again, two solid vertical lines mark the dates on which RIPE NCC (November 26th, 2025) and ARIN (January 20th, 2026) enabled ASPA in their respective RPKI dashboards.

First publications are, by construction, the month-over-month increment in the total Customer AS count, and their sharp rise from January 2026 onward reflects the rapid growth already visible in Figure 2—notice the logarithmic scale. The monthly number of removal and re-publication events follows each other closely throughout the observation period. Both counts remain largely stable at 10–20 Customer ASes per month for most of the period, rising to 64 removals in March 2026—a small fraction of the 1500+ Customer ASes active that month—pointing to a somewhat constant number of short-lived ASPA objects in the RPKI. Updates, by contrast, scaled with the growing number of Customer ASes, suggesting that operators were still refining their configurations during this initial large-scale deployment phase. On average, there were 6.3 updates per day in 2026.

Next, we investigate how and to what extent the content of ASPA objects evolved over the course of March 2026. We restrict this analysis to the 980 Customer ASes who first published a valid ASPA object no later than January 31, 2026, so that any initial corrections to the provider list fall outside the analysis time frame. For each of these Customer ASes, we track day-by-day changes to the provider list across all March 2026 snapshots, and we count the total number of providers added and removed cumulatively over the month. Figure 4 shows the distribution of such Customer ASes according to these two quantities. 89 ASes removed their ASPA object from the RPKI in February and do not appear in the heatmap. The majority of Customer ASes (808) made no changes to their provider list during March 2026, as evidenced by the dominant cell at (0, 0). Of the remaining 83, most either added (37) or removed (8) a single provider. Interestingly, 4 ASes removed a

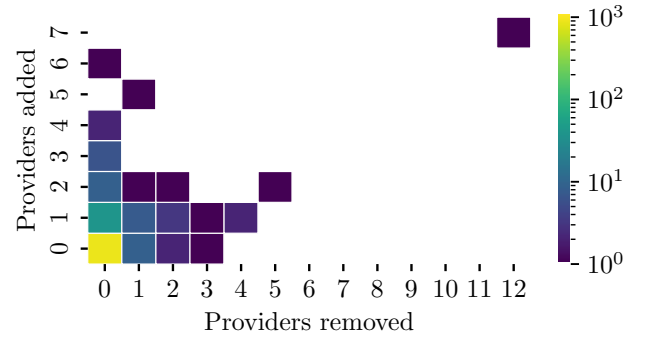


**Figure 4: Distribution of pre-February 2026 Customer ASes based on the cumulative number of providers they added (y-axis) and removed (x-axis) from their ASPA object throughout March 2026.**

provider only to re-introduce it a few days later. The most outlying AS is AS206604, which removed its ASPA object from the RPKI in January and re-published one only in March with a drastically different list of authorized providers: it retained only one of the 13 providers listed in January and added 4 new providers. It also added 3 further providers throughout March. We reached out to AS206604 and the other 15 ASes that added or removed more than 2 providers, asking for insights into their ASPA-related operations throughout March: 5 of them replied. They confirmed that the changes we observed reflected legitimate operations: new business relationships, alignment of ASPA with the current upstream configuration, and infrastructure changes. Overall, ASPA objects show high stability: approximately 96 % of Customer ASes made no or a single change throughout March 2026.

### 3.2 Customer ASes

Finally, we investigate and characterize the early ASPA adopters. For the remainder of this section, we focus on the 1505 Customer ASes having a valid ASPA object published in the RPKI as of March 31st, 2026. Among them, four are Tier-1 ASes: Cogent (AS174), Arelion (AS1299), Liberty Global (AS6830), and AT&T (AS7018).<sup>3</sup> Using the tags that `bgp.tools` assigns to ASes, we find universities (19) and governments (18), personal ASes (250), and ASes hosting websites in the Tranco top 10k (35).

Since multiple ASes can belong to the same organization, we map each AS to the organization that owns it using CAIDA’s AS2Org dataset [8] and analyze ASPA adoption at the organization level. The dataset includes organization mappings for all 1505 Customer ASes, which are mapped to 1253 organizations: most of these organizations (982) own a single AS, 250 own 2–5 ASes, and 15 own 6–9 ASes; 6 organizations own between 24 and 123 ASes. Among the 271 organizations that own more than 1 AS, 127 (47 %) adopted ASPA across all their ASes (*full-adopters*). The largest full-adopters in terms of number of associated ASes are (1) Edgoo Networks (AS6424) and (2) Telia Norge (AS25400); both run 8 ASes. Among

<sup>3</sup>We refer to `bgp.tools`’ list of Tier 1 ASNs available at <https://bgp.tools/kb/what-is-a-upstream>.

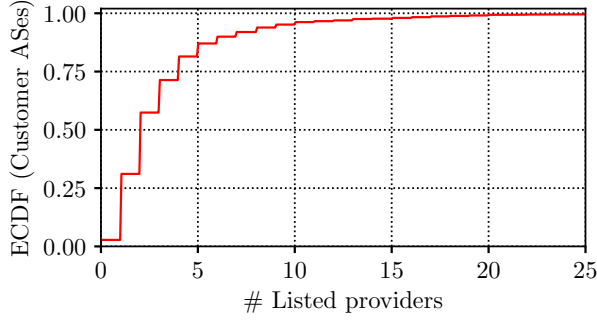


**Figure 5: Number of providers listed in valid ASPA objects across the 1505 Customer ASes of March 31st.**

**Table 1: Top-5 most-declared providers among Customer ASes (CASs), separately for RIPE NCC and ARIN regions.**

| RIPE NCC |        |        | ARIN     |        |        |
|----------|--------|--------|----------|--------|--------|
| Provider | # CASs | % CASs | Provider | # CASs | % CASs |
| AS6939   | 166    | 14.6   | AS6939   | 63     | 21.9   |
| AS1299   | 103    | 9.1    | AS174    | 36     | 12.5   |
| AS174    | 97     | 8.5    | AS20473  | 33     | 11.5   |
| AS34927  | 90     | 7.9    | AS3356   | 19     | 6.6    |
| AS212895 | 70     | 6.2    | AS1299   | 19     | 6.6    |

the 144 *partial-adopters*, ESNet owns 24 ASes, of which 19 are Customer ASes. However, its most relevant AS, AS293, has not adopted ASPA.

We now examine the providers authorized by Customer ASes. In Figure 5 we show the distribution of the number of providers that each Customer AS declares in their ASPA object. 42 Customer ASes have a so-called “AS0 ASPA”, i.e., an ASPA object that lists only AS0 as a provider, meaning that they have no upstream providers and are not RS-clients at any non-transparent route server. Among these, four are the aforementioned Tier-1s and 17 are IXPs—both legitimate cases of AS0 ASPAs, as Tier-1s have no upstream providers and IXPs do not transit traffic on behalf of other ASes. The remaining AS0 ASPAs belong to ASes that, except for AS4402 and AS400095, announce no IP prefixes. Most Customer ASes (84 %) declare between 1 and 5 providers, whereas only 14 declared 20 or more; the most extreme outlier is AS49935, which lists 66 providers in its ASPA object.

Table 1 breaks down the top-5 most-declared providers, separately for Customer ASes from RIPE NCC and ARIN. Hurricane Electric (AS6939, considered a Tier-1 AS for IPv6 traffic) is by far the most commonly listed provider in both regions, authorized as upstream by 14.6 and 21.9 % of the Customer ASes from RIPE NCC and ARIN, respectively. Three more Tier-1s appear in the two top-5s: Cogent (AS174) and Arelion (AS1299) for both regions, and Lumen (3356) only for ARIN.

Next, we consider the size of Customer ASes in terms of both customer cone size (i.e., the number of direct and indirect customers they have [34]), and the number of IP prefixes they announce (/24

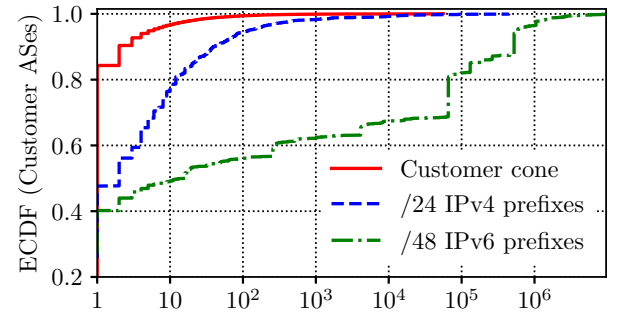


**Figure 6: Distribution of the size of Customer ASes based on their customer cone size, and the number of IPv4 /24 and IPv6 /48 subnets they announce.**

and /48 for IPv4 and IPv6, respectively). Although the customer cone size is an inferred metric, it serves as a good proxy for an AS’s influence in the routing ecosystem and is used by CAIDA to obtain its ranking of ASes (ASRank), which places the AS with the largest customer cone in first place (in this case AS3356, Lumen). We utilize the customer cones inferred by CAIDA in March 2026 [9] and associate each AS with IPv4 and IPv6 prefixes based on RIPE’s RISwhois dumps from March 31st, 2026 [48]. Since each AS is usually considered part of its own customer cone, leaf ASes (i.e., ASes with no customers) will have a customer cone of 1. Figure 6 shows the distribution across the current Customer ASes of the customer cone size (solid red line), number of /24 IPv4 subnets (blue dashed line), and /48 IPv6 subnets (green dash-dotted line). Looking first at the customer cone size distribution, we find that 1134 (75 %) Customer ASes are leaf ASes. This result follows the natural distribution of AS sizes on the Internet: of the 79.5k ASes in the dataset of customer cones, 67k (84 %) are leaf ASes. In terms of ASRank, 9 Customer ASes are in the top-100 and an additional 33 in the top-1000. Turning to the IP prefixes distributions: 48 % of Customer ASes announce at most one /24 IPv4 subnet (325 announce exactly one, 391 announce none), and 40 % announce at most one /48 IPv6 subnet (191 announce exactly one, 414 announce none). Overall, 176 Customer ASes announce no IP prefixes at all.

Lastly, we assess to what degree Customer ASes adopt the two RPKI-based mechanisms for route origin security: ROA and ROV. For ROA, we retrieve from Cloudflare Radar [11] the ratio of ROA-covered announced address space as of March 31st, 2026, for each of the 1329 Customer ASes that announce at least one IP prefix. Most (1100) are fully ROA-authorized, while some are authorized for none (28) or less than half (23) of their announced address space. The most outstanding case is that of AS14654, which announces 6618 /24 IPv4 prefixes and 65 536 /48 IPv6 prefixes (specifically, a single /32 IPv6 prefix), but covers none of them with ROAs.

For ROV, we use the RoVista dataset [33], which *estimates* the fraction of ROA-invalid routes an AS filters based on active measurement. Indeed, unlike ROA, which is a signed object that can be directly retrieved and validated, ROV is an operational behavior that can only be estimated through active or passive measurement. RoVista provides data for only 346 (26 %) of the 1505 Customer ASes,

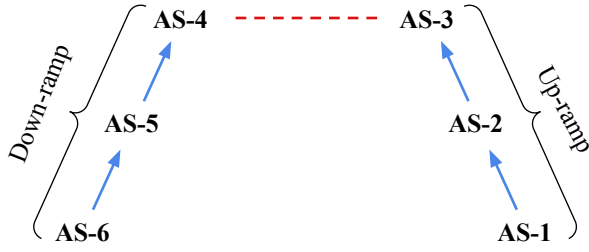


Figure 7: Illustration of the up-ramp and down-ramp for the compressed AS Path  $\langle \text{AS-6 AS-5 AS-4 AS-3 AS-2 AS-1} \rangle$ . The arrows indicate the AS relationship (blue solid for customer-to-provider, red dashed for peer-to-peer) attested by VAPs, whereas the sequence of numbers indicates the order of ASes in the AS Path, with AS-1 being the originator of the announcement and AS-6 the most recently added.

reflecting the limited visibility inherent to inferring ROV compared to reading ROA objects directly. Among these, 123 Customer ASes filter all invalid routes, 215 show no filtering, and the remaining 8 filter partially.

**Take Away:** *ASPA adoption rapidly increased when RIPE enabled it in the RPKI dashboard. There are some very large networks among the early adopters, including four Tier-1 ASes.*

## 4 Finding ASPA route leaks in the wild

Alongside ASPA objects, the IETF draft also proposes the AS Path verification procedure [3] that allows BGP routers to detect route leaks by verifying that a received AS Path is valley-free (according to the available VAPs). In this section, we illustrate this procedure, present a methodology for measuring route leaks in the wild, and report on our results.

### 4.1 ASPA AS Path verification

The ASPA AS Path verification procedure described in the Internet draft [3] is intended for live BGP edge routers to validate announcements received from neighboring ASes.<sup>4</sup> Consider, for example, the AS Path illustrated in Figure 7. Starting from the AS that originates the announcement (AS-1), the procedure walks the path upward following attested customer-to-provider hops<sup>5</sup> (illustrated with blue arrows); this stretch is the *up-ramp*. Symmetrically, the *down-ramp* is obtained by walking the path backwards, starting from the most recently added AS (AS-6), and following attested customer-to-provider relationships<sup>6</sup>. A valley-free path requires that the up-ramp and down-ramp together cover the entire path, allowing at most a single peer-to-peer link at their junction (e.g., the hop from AS-3 to AS-4); if instead the two ramps leave a gap (i.e., one or more ASes that are not part of either ramp), the path contains a valley and is flagged as *Invalid*. Such invalid paths are

<sup>4</sup>The procedure operates on the received *compressed* AS Path, i.e., the sequence of unique ASNs obtained after removing duplicated ASNs.

<sup>5</sup>AS-1  $\rightarrow$  AS-2 is an attested customer-to-provider hop if AS-2 is in the VAP of AS-1.

<sup>6</sup>Because ASPA does not allow providers to be constrained to certain prefixes or a single IP version, the ramps provide a conservative estimate of the announcement's true customer-provider hops.

Table 2: Inferred BGP roles that Customer AS AS-N may assume when sending a route to Customer AS AS-R, and the resulting algorithm choice.

| $N \in \text{VAP}(R)$ | $R \in \text{VAP}(N)$ | BGP Roles of AS-N |              |              | Algorithm |
|-----------------------|-----------------------|-------------------|--------------|--------------|-----------|
|                       |                       | Prov.             | Cust.        | Peer         |           |
| $\times$              | $\times$              | $\times$          | $\times$     | $\checkmark$ | Upstream  |
| $\times$              | $\checkmark$          | $\times$          | $\checkmark$ | $\checkmark$ | Upstream  |
| $\checkmark$          | $\times$              | $\checkmark$      | $\times$     | $\checkmark$ | N/A       |
| $\checkmark$          | $\checkmark$          | $\checkmark$      | $\checkmark$ | $\checkmark$ | N/A       |

likely route leaks and should be considered ineligible for route selection [3].

Whether a received AS Path is Valid or Invalid depends not only on the received path, but also on the relationship between the receiving AS (which runs the verification) and the neighboring AS that is sending the route. Consider again the path in Figure 7, sent by AS-6 to a neighboring AS, which we denote AS-7<sup>7</sup>. If AS-6 is a provider of AS-7, the path is Valid, because it exhibits the expected valley-free structure—an up-ramp, a peer link, and a down-ramp. On the other hand, if AS-6 is a customer or peer of AS-7, the same path is Invalid (respectively, a type 1 and type 3 leak, see Figure 1), because a route received from a customer or peer should consist solely of an up-ramp.

Such a role in the relationship must be declared either via router-specific configuration or via the BGP Role Capability [4], a recent enhancement of the BGP OPEN message that allows neighboring ASes to mutually declare and validate their relationship (e.g., provider, customer, peer). Based on the declared role, the ASPA AS Path verification procedure comprises two distinct algorithms. A BGP router selects which algorithm to apply based on the BGP role of the neighboring AS that sent the route [3, 4]. The *downstream verification algorithm* applies to routes received from a provider: such routes are traveling *down* the hierarchy toward the verifying AS, so the path may legitimately include both an up-ramp and a down-ramp connected by an optional peer link. The *upstream verification algorithm* applies instead to routes received from a customer or peer: such routes are traveling *up* the hierarchy, so the received path must consist entirely of an up-ramp, and any provider-to-customer or peer-to-peer link is an immediate indicator of a route leak.

Because some ASes in the path may not be associated with any VAP, the ASPA AS Path verification algorithms distinguish three outcomes, *Valid*, *Unknown*, and *Invalid*, where Unknown indicates that available attestations are insufficient to confirm or rule out a route leak. Appendix A further discusses the upstream and downstream verification algorithms, and shows how the ASPA AS Path Verification procedure detects the four types of route leaks illustrated in Figure 1.

Applying ASPA verification to publicly available BGP data requires adapting the procedure described above. Unlike a live BGP router, which knows its own AS relationships and can select the appropriate algorithm accordingly, an offline analysis has no access

<sup>7</sup>Note that the receiving (and verifying) AS does not belong to either ramp.

to the BGP role under which any route was exchanged between consecutive ASes. This limitation is further compounded by *complex relationships*, in which the BGP roles between two ASes may differ depending on the announced IP prefix [20]. Consequently, it is not immediately clear how ASPA AS Path verification can be applied to publicly available BGP data.

However, when two consecutive ASes in a path have a valid ASPA object, we can partially resolve this ambiguity, as summarized in Table 2. Consider a route received by AS-R from its neighbor AS-N. If AS-N is *not* listed as an authorized provider in VAP(AS-R) (first two rows of the table), then AS-N cannot be a provider of AS-R; because AS-N can only be a customer or a peer, AS-R should apply the upstream verification algorithm. If instead AS-N is listed as a provider in VAP(AS-R) and AS-R is not listed in VAP(AS-N) (third row), the algorithm choice remains ambiguous because a peer-to-peer relationship cannot be ruled out due to complex relationships. Finally, if both ASes list each other as providers (fourth row), the relationship is unresolvable: AS-N could be a provider, customer, or peer of AS-R, and no algorithm can be selected unambiguously.<sup>8</sup> Note that Table 2 never rules out the possibility of a complex relationship between AS-N and AS-R: a verification algorithm is selected only when it applies regardless of whether the relationship is simple or complex.

Building on the above analysis, Algorithm 1 presents the procedure to verify AS Paths from publicly available AS Paths, where the BGP role under which each adjacent pair of ASes exchanged the route is unknown. Given an AS Path, the algorithm treats the most recently added AS as the receiving AS-R and the remainder as the received path. If AS-R has a valid ASPA object and AS-N is not listed among its authorized providers, AS-N can only be a customer or a peer, and upstream verification is applied. Otherwise, the algorithm relies on a fallback mechanism. Both upstream and downstream verification are run; if they agree, the shared outcome—one of *Valid*, *Unknown*, or *Invalid*—is returned; otherwise, *N/A* is returned.

Note that the two branches of the algorithm produce outcomes of different scope. When only upstream verification is applied (line 5), the result is specific to AS-R: the path is valid, unknown, or invalid as received by this particular AS. When upstream and downstream verification agree (line 10), by contrast, the shared outcome holds regardless of AS-R: the path is structurally valid, unknown, or invalid. We distinguish *Unknown* from *N/A* accordingly. The former is a legitimate outcome of the ASPA verification procedure, indicating that the available VAPs are insufficient to validate or invalidate the path. The latter arises specifically from the fallback mechanism, when upstream and downstream verification disagree: the outcome depends on whether the path is treated as heading upstream or downstream, and no conclusion can be drawn deterministically.

To understand the *N/A* outcome, consider the following input to Algorithm 1:  $P = \langle \text{AS-3 AS-2 AS-1} \rangle$ ,  $VAPs = \emptyset$ . Because AS-3, the receiver, has no VAP, the fallback mechanism is used. Upstream verification returns *Unknown*: it cannot rule out that AS-1 to AS-2 is a provider-to-customer link, which would make the path invalid under the upstream assumption. Downstream verification, by contrast, returns *Valid*: a two-AS downstream path imposes no constraints

<sup>8</sup>Among the 1505 Customer ASes of March 31st, 2026, there are 43 pairs of ASes that list each other as authorized providers. These could represent complex relationships, backup links for either AS, or an ASPA misconfiguration.

**Algorithm 1** AS Path verification procedure on passively-obtained AS Paths, where the BGP role of the neighboring AS is not known.

**Require:** Compressed AS Path  $P = \langle AS_n, \dots, AS_2, AS_1 \rangle$  of length  $n$ ; Validated ASPA Payloads  $VAPs$

**Ensure:** Verification outcome: *Valid*, *Unknown*, *Invalid*, or *N/A*

```

1: receiver  $\leftarrow AS_n$             $\triangleright AS_n$  is the most recently added AS
2: neighbor  $\leftarrow AS_{n-1}$ 
3: received  $\leftarrow \langle AS_{n-1}, \dots, AS_1 \rangle$ 
4: if receiver has a valid ASPA object and neighbor  $\notin$ 
   VAP(receiver) then
5:   return UPSTREAMVERIFICATION(received, VAP)
6: else                                $\triangleright$  Fallback mechanism
7:   resultup  $\leftarrow$  UPSTREAMVERIFICATION(received, VAP)
8:   resultdown  $\leftarrow$  DOWNSTREAMVERIFICATION(received, VAP)
9:   if resultup = resultdown then
10:    return resultup
11:   else
12:    return N/A
13:   end if
14: end if
```

on the relationship between AS-1 and AS-2. The two verifications disagree, and the algorithm returns *N/A*.

The possibility of running such offline analyses might prove important to understanding ASPA deployment status or studying route leaks at scale. Although ASPA is designed to run on routers, we argue that it can be successfully applied to BGP routes captured by collectors to provide meaningful insights.

## 4.2 Findings

To measure route leaks in the wild, we apply Algorithm 1 to CAIDA's dataset of all known AS Paths from March 2026 [9], which CAIDA obtains by downloading route collectors' data during the first five days of each month [34]. Additionally, the dataset includes the route collector that observed each path and the announced IP prefix.

The dataset contains 49 907 751 unique AS Paths, which we pre-process and filter as follows. First, we compress each AS Path by removing duplicate ASNs, as Algorithm 1 operates on compressed paths. Second, we extract all subpaths by iteratively removing the most recently added AS from each compressed path. Since the dataset already contains paths that are subpaths of others, this step normalises the dataset so that all subpaths are represented uniformly, regardless of the paths originally observed. Each resulting subpath corresponds to a route exchanged between two ASes, and is therefore a valid input to our algorithm. Third, we discard paths shorter than three ASes, because Algorithm 1 would treat any two-AS path  $\langle AS-2 AS-1 \rangle$  as AS-2 receiving the path  $\langle AS-1 \rangle$ , which is valley-free regardless of any ASPA objects. After pre-processing, we obtain 56 708 620 unique paths.

To align with CAIDA's five-day observation window, we restrict the analysis to Customer ASes holding a valid ASPA object on at least one day during March 1–5, 2026. For each of the 1252 such Customer ASes, we select the most recent VAP observed within that window, as it best reflects the operator's intended policy at the time the AS Paths were collected. We then apply Algorithm 1 to

**Table 3: Outcome of Algorithm 1 on the pre-processed AS paths.**

| Outcome | # Paths    | % Paths |
|---------|------------|---------|
| Valid   | 231 842    | 0.41    |
| Invalid | 36 655     | 0.06    |
| Unknown | 42 625 780 | 75.17   |
| N/A     | 13 814 343 | 24.36   |
| Total   | 56 708 620 | 100.00  |

each of the 56 708 620 pre-processed paths, using these 1252 VAPs as the attestation source, and attribute the returned outcome to the entire input path, including the receiving (verifying) AS. Table 3 shows the results.

The vast majority of paths (99.5 %) yield *Unknown* or *N/A* as the verification result. This is a direct consequence of the current state of ASPA adoption: the 1252 Customer ASes represent 1.44 % of all ASes observed in the dataset. There are 268 497 paths with a definitive result: 231 842 (86 %) are *Valid*, 36 655 (14 %) are *Invalid*. Of the 231 842 valid paths, 26 211 were verified using upstream verification alone (i.e., the receiving AS has a valid ASPA object and its neighbor is not listed as a provider; see Table 2), while the remaining 205 631, nearly eight times as many, resulted from the fallback mechanism where both upstream and downstream verification yielded *Valid*. For the invalid paths, the split is more balanced: 13 956 were flagged by upstream verification alone and 22 699 by the fallback mechanism.

The 13 814 343 *N/A* paths arise from three cases where upstream and downstream verification disagree: 7 628 944 paths are *Unknown* upstream and *Valid* downstream, 6 052 607 are *Invalid* upstream and *Unknown* downstream, and 132 792 are *Invalid* upstream and *Valid* downstream. In Section B, we attempt to resolve these paths by inferring their direction from external AS relationship datasets.

Matching the invalid paths against the initial CAIDA dataset [9], which also indicates the IP prefix associated with each path, we find that 452 of them are ROA-invalid. They correspond to 24 distinct announcements, i.e., 24 unique (origin AS, prefix) pairs, that originate from 13 distinct ASes. Among these, one stands out: AS61574, which originates 8 ROA-invalid announcements.

Recall that an invalid path according to Algorithm 1 can be due to: (1) one of the ASes shared the path outside of its intended scope (thus causing a route leak), or (2) one of the ASes—including the receiving AS—has misconfigured its ASPA object by not listing one of its authorized providers. The ASPA AS Path verification procedure, however, cannot necessarily determine the AS that caused the route leak [3].

Although verifying the correctness of currently deployed ASPA objects is beyond the scope of this paper, we highlight one interesting case: 354 invalid paths consistently include AS6777 (AMS-IX), one of the largest IXPs in Europe. According to AMS-IX’s own documentation [1], its Route Servers operate transparently, i.e., they do not add their own ASN when relaying announcements between clients. The current ASPA draft [55] specifies that transparent Route Servers do not need to be listed as authorized providers, since they are not expected to appear in the AS Path. Consistent with this, of

AMS-IX’s 813 member ASes<sup>9</sup>, 60 held a valid ASPA object during the observation window, and none of them listed AS6777 as an authorized provider. We reached out to AMS-IX directly for clarification. According to their operators, a subset of member networks explicitly prepend AS6777 to their AS Path before propagating updates internally (internal BGP, iBGP). Depending on each member’s export policy, this manually added hop may persist when these routes are later announced to their peers.

Since ASPA is still under standardization and adoption is relatively low, these practices are currently harmless (with respect to ASPA). However, once live BGP routers begin enforcing ASPA validity on received paths, such routing policies may isolate an AS and all its customers, since an invalid AS Path is to be treated as ineligible for route selection and therefore not forwarded to neighboring ASes[3].

To validate the set of 36 655 identified invalid paths, we cross-reference our results with route leak events reported by Cloudflare during March 1–5, 2026. Cloudflare reports each leak as a three-AS tuple  $\langle \text{sender}, \text{leaker}, \text{receiver} \rangle$ , here referred to as *leak tuples*, rather than a full AS Path. Of the 730 distinct leak tuples reported over the period, 299 (41 %) do not appear in any of the 56 708 620 paths that compose our dataset; this is likely because they occurred outside of CAIDA’s path collection window. Among the 431 leak tuples that do appear in the dataset, 29 are found in 657 of our 36 655 invalid paths (1.8 % coverage). Conversely, only 1 leak tuple appears in a path that our algorithm classified as *Valid*: (AS210464, AS205941, AS213449). The remaining 401 leaks only appear in paths with outcome *Unknown* or *N/A*.

Next, we conduct a temporal analysis of the 36 655 invalid paths over the course of March 2026. For each day, we classify each invalid path into one of three categories: *Not announced* (no longer observed at any route collector), *Announced, Invalid* (still observed and still yielding *Invalid* under updated ASPA objects), and *Announced, Other* (still observed but no longer yielding *Invalid*). To track announcement status, we download daily BGP route dumps from RIPE NCC and RouteViews route collectors throughout March 2026.<sup>10</sup> To track validity, we re-evaluate each path daily against the most recent ASPA objects from the same 1252 Customer ASes, so that a change in outcome reflects only updates to those ASPA objects.

Figure 8 shows the classification over time as a stacked area chart, where each area represents the percentage of the 36 655 invalid paths falling into each category, aggregated in non-overlapping five-day windows aligned with CAIDA’s observation period, with the last day of each window as the representative timestamp. The *Not announced* area (green) grows steadily throughout March, reaching approximately 35 % by March 30th, indicating that roughly one-third of invalid paths disappeared from route collectors within the month. The *Announced, Other* area (blue) remains narrow throughout, stabilizing at around 3 % by March 30th, reflecting that ASPA updates correcting provider lists are rare but do occur. No path that transitioned out of *Announced, Invalid* subsequently reverted to it.

<sup>9</sup>The list of RS-clients of AMS-IX was obtained from <https://www.ams-ix.net/ams/connected-networks>.

<sup>10</sup>We restrict the collection to the vantage points that observed the 36 655 invalid paths, avoiding the need to download and parse full dumps from all available route collectors.

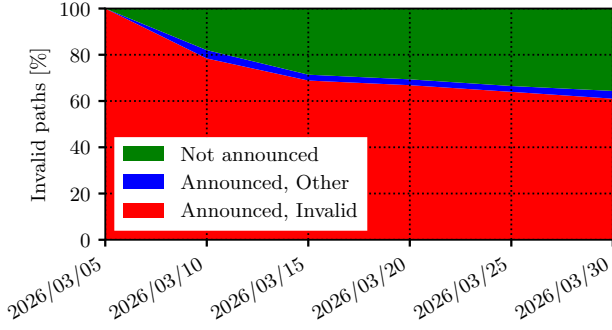


**Figure 8: Breakdown of the 36 655 invalid paths based on their presence in route collectors (Announced / Not announced) and their validity.**

Lastly, we analyze invalid paths to gain additional insights into their origin and propagation. To this end, we identify, for each of the 36 655 invalid paths, its smallest invalid subpath, which we call its *minimal invalid path* (MIP). Formally, a MIP is a path for which Algorithm 1 yields *Invalid*, while the next smaller subpath yields *Valid*, *Unknown*, or *N/A*. Consider for example a generic invalid path  $\langle \text{AS-5 AS-4 AS-3 AS-2 AS-1} \rangle$ . If Algorithm 1 marks as *Invalid* the subpath  $\langle \text{AS-4 AS-3 AS-2 AS-1} \rangle$  but not the subpath  $\langle \text{AS-3 AS-2 AS-1} \rangle$ , then  $\langle \text{AS-4 AS-3 AS-2 AS-1} \rangle$  is the MIP, and AS-4 is the MIP’s verifying AS. The MIP therefore marks the earliest point along an invalid path at which Algorithm 1 identifies it as a likely route leak; any propagation past the MIP’s verifying AS reflects how far a path continues to be forwarded after it has already become detectable as such.

We find that the 36 655 invalid paths reduce to 13 406 distinct MIPs; they originate from 3818 distinct ASes, with no single AS dominating significantly: AS61574 and AS271150 are the most frequent sources, contributing 210 and 106 MIPs, respectively. Only 490 of these 3818 ASes held a valid ASPA object during the observation window, suggesting that most invalid paths are not caused by ASPA misconfigurations in the originating ASes.

Most MIPs are short: 5822 consist of four ASes, 4392 consist of five, and 1482 of three. Only 32 of the remaining 1710 MIPs reach a length of nine or more ASes, with a single outlier made of 19 ASes. Since these numbers suggest that most MIPs are detectable close to their origin, we next examine how far they propagate. Figure 9 shows, for each MIP, the total number of ASes that received it (solid red line) as well as the number of ASes that received it directly from the MIP’s verifying AS (dashed blue line). We observe that the overall propagation of MIPs is largely limited: 10 829 (81 %) MIPs are not propagated beyond their verifying AS. Of the remaining 2577 MIPs, 1816 (14 %) reach exactly one further AS and only 313 (2 %) are received by more than 10 ASes; three outliers are received by more than 300 ASes. Turning to direct propagation, 1981 (77 %) of the 2577 propagating MIPs are sent by the verifying AS to exactly one neighbor, indicating that even among MIPs that do propagate, direct spread is typically narrow. There are 11 MIPs that are directly propagated to more than 100 neighbors: all of them have a large AS as the verifying AS of the MIP (namely, AS174, AS1299, AS5405, and

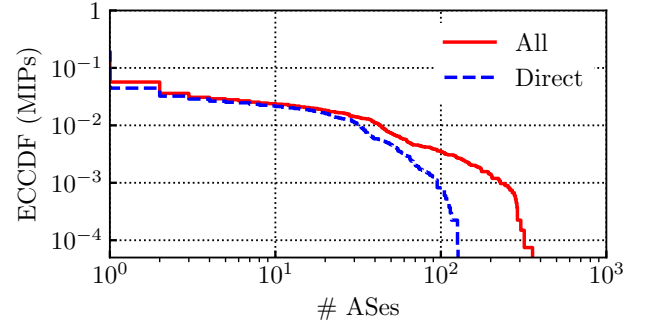


**Figure 9: Distribution of the number of ASes that received each of the 13 406 MIPs (solid red line), as well as the number of ASes that received the MIP directly (dashed blue line).**

AS33891). It should be noted that these distributions are strongly influenced by the vantage points from which AS Paths are collected.

**Take Away:** *Although ASPA is designed for live BGP routers, AS Path verification can be successfully applied to passively collected BGP data, enabling route leak detection at Internet scale. Not all invalid paths reflect genuine route leaks: some are attributable to ASPA misconfigurations, as evidenced by operator updates to provider lists that resolved invalidity during the observation period.*

## 5 Comparing ASPA with inferred providers

We finally evaluate how the relationships declared through ASPA objects compare with those inferred by popular and well-established initiatives. AS Relationships are commercial agreements between two ASes that dictate both routing decisions and the possible BGP Roles the two ASes may assume when exchanging routes. Because such agreements are typically confidential, researchers have long proposed methodologies to infer them from publicly available data sources [20, 23, 30, 31, 34], and inferred relationships have since been widely used to study the structure, security, and dynamics of Internet routing [2, 12, 39, 46, 53]. However, inference methods rely uniquely on publicly available BGP data, namely AS Paths retrieved from RouteViews and RIPE RIS route collectors, which provide an incomplete view of the global routing ecosystem. Furthermore, no ground-truth data has previously been available to systematically validate inferred relationships at scale [46]. ASPA objects represent the first opportunity to address this gap, as they are intended to publicly disclose such information about the authorized upstream providers of an AS [17].

In this section, we compare the self-declared upstream providers listed in ASPA objects with those inferred by CAIDA [9] and BGP-KIT [7] to measure agreement and characterize discrepancies. Both infer AS relationships from the same publicly available data sources: AS Paths collected by RouteViews and RIPE RIS route collectors. However, the two differ in the way customer-to-provider relationships are inferred. CAIDA relies on a multi-step algorithm (originally described in [34], though it has undergone minor changes since) that first identifies providers from observed AS-path triplets, and then extends these inferences using transit-degree comparisons and previously inferred relationships. BGPKIT, on the other hand,

uses a more straightforward approach [13], marking all links from the origin toward the first Tier-1 as customer-to-provider.<sup>11</sup>

### 5.1 Provider sets

We focus on the customer-to-provider relationships inferred in March 2026 and align CAIDA’s and BGPKIT’s datasets to the common observation window of March 1st to 5th. Thus, we restrict the following analyses to the 1252 Customer ASes that maintained a valid ASPA during March 1–5, 2026. For each, we define four provider sets: two derived from inference algorithms (**CAIDA** and **BGPKIT**) and two from ASPA declarations (**ASPA** and **ASPA<sub>vis</sub>**), the latter serving as reference sets for evaluation.

**CAIDA:** CAIDA infers AS relationships monthly from AS Paths collected during the first five days of each month, releasing separate datasets for IPv4 and IPv6, reflecting that ASes may rely on different upstream providers depending on the IP address family [34]. Because ASPA objects do not support per-address-family provider declarations—a deliberate design choice to simplify both registration and verification [3]—we define **CAIDA** as the union of providers from both datasets. Thus, AS-2 is considered a provider of AS-1 if CAIDA classifies it as such in at least one of the two datasets. Notice that these relationships are inferred from the same AS Path dataset used in Section 4.2 to identify invalid paths [9].

**BGPKIT:** BGPKIT releases three daily AS relationship datasets: one for IPv4, one for IPv6, and a combined generic dataset [7]. We use the combined generic dataset and define **BGPKIT** as the union across the five daily snapshots covering March 1–5: for a given Customer AS, AS-2 is in **BGPKIT** if BGPKIT classifies it as a provider on at least one of those days.

**ASPA:** For a given Customer AS, **ASPA** is the set of upstream providers declared in the most recent valid ASPA object observed within the March 1–5 window (i.e., the content of the most recent associated VAP).

**ASPA<sub>vis</sub>:** In practice, inference algorithms classify a relationship between two ASes only if they appear as adjacent hops in at least one observed AS Path. However, a provider listed in an ASPA object may not appear in *any* observed AS Path for several reasons, including backup links that are not used during the observation window, limited coverage of available route collectors’ vantage points, or misconfigurations of the ASPA object itself. Comparing inferred providers directly against **ASPA** is therefore asymmetric: **ASPA** may include entries that are practically invisible to inference methods. To account for this, for each Customer AS, we define **ASPA<sub>vis</sub>** as the intersection of its **ASPA** with the set of ASes appearing adjacent to that customer in at least one AS Path in CAIDA’s March 1–5 dataset. **ASPA<sub>vis</sub>** thus retains only operator-declared providers observable in BGP data during the evaluation period, providing a fairer baseline than **ASPA** for measuring inference accuracy.

First, we analyze how the four provider sets differ in size. Figure 10 shows the distribution of the number of providers per Customer AS according to **ASPA**, **ASPA<sub>vis</sub>**, **CAIDA**, and **BGPKIT**. Between the two inference datasets, BGPKIT assigns provider relationships less strictly than CAIDA, with the **BGPKIT** distribution closely tracking that of **ASPA<sub>vis</sub>** in terms of provider counts. CAIDA is the most conservative, with the **CAIDA** distribution

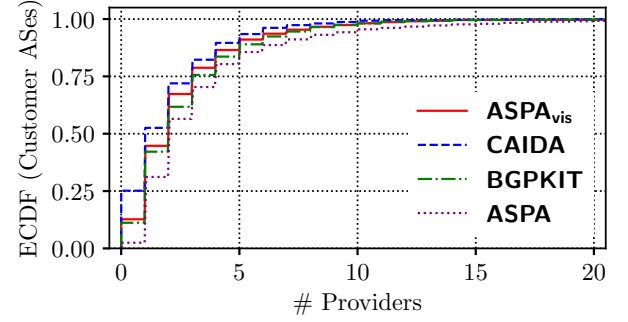


Figure 10: Distribution across Customer ASes of the number of providers according to ASPA, ASPA<sub>vis</sub>, CAIDA, and BGPKIT.

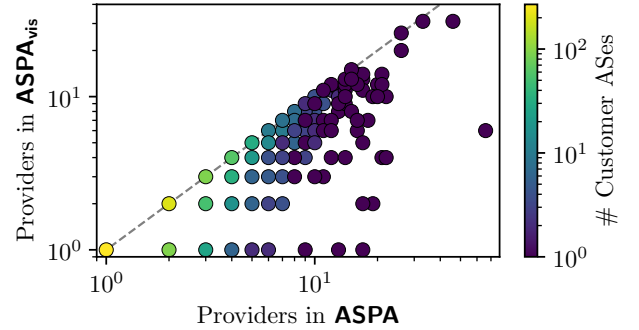


Figure 11: Distribution of Customer ASes based on the size difference of their ASPA and ASPA<sub>vis</sub>.

shifted toward lower provider counts, reflecting a stricter threshold for classifying a relationship as customer-to-provider. The gap between **ASPA** and **ASPA<sub>vis</sub>** is substantial; the number of Customer ASes with no providers grows from 31 for **ASPA** (i.e., AS0 ASes) to 159 for **ASPA<sub>vis</sub>**.

We finally take a closer look at the difference between **ASPA** and **ASPA<sub>vis</sub>** for the 1093 Customer ASes with non-empty **ASPA<sub>vis</sub>**. Figure 11 shows the distribution of Customer ASes based on the difference in size between their **ASPA** (x-axis) and **ASPA<sub>vis</sub>** (y-axis) sets. There are a total of 687 Customer ASes for which **ASPA** and **ASPA<sub>vis</sub>** are identical; these are represented by the dots that lie on the dashed grey line. Most of the remaining 406 Customer ASes are close but underneath such a line, with their **ASPA<sub>vis</sub>** missing only one (237) or two (73) of the providers declared through ASPA. The most extreme outlier is AS49935: as discussed in Section 3, it lists 66 providers, of which only 6 appear in its **ASPA<sub>vis</sub>** set. Overall, the difference between **ASPA** and **ASPA<sub>vis</sub>** sets could be due to numerous reasons, including unobserved backup links, overpermissive provider declarations, or the limited view provided by route collectors.

<sup>11</sup>The code is available at <https://github.com/bgpkit/peer-stats>.

## 5.2 Inference accuracy evaluation

We now compare how similar the sets are by framing the problem as a classification task. We thus evaluate the accuracy of **CAIDA** and **BGPKIT** as a binary classification problem, evaluating each against two reference sets, **ASPA**, and **ASPA<sub>vis</sub>**. This is not an assertion that ASPA objects are correct: they reflect operator declarations, which may be incomplete or misconfigured, as discussed in Section 4.2. For a given Customer AS and reference set  $\mathcal{R}$  (one of **ASPA** and **ASPA<sub>vis</sub>**), let  $\mathcal{I}$  denote the inferred provider set (**CAIDA** or **BGPKIT**):

- A true positive (TP) is an AS in both  $\mathcal{I}$  and  $\mathcal{R}$ ;
- A false positive (FP) is an AS in  $\mathcal{I}$  but not in  $\mathcal{R}$ ;
- A false negative (FN) is an AS in  $\mathcal{R}$  but not in  $\mathcal{I}$ .

We exclude from the following analyses the 159 Customer ASes (out of 1252) for which **ASPA<sub>vis</sub>** is empty, as they provide no meaningful reference for this comparison.

We first compute a global view of inference accuracy by summing TP, FP, and FN counts across all Customer ASes—so that each provider relationship contributes equally regardless of the Customer AS it belongs to—and deriving precision, recall, and F1 from these totals; results are reported in Table 4. Precision is consistently high for both inference algorithms and against both reference sets, indicating that inferred providers rarely include ASes not in **ASPA**. The metric alone, however, hides the difference in TP and FP counts achieved by the two. CAIDA exhibits a more cautious behavior in assigning customer-to-provider relationships (2175 TPs and 289 FPs) compared to BGPKIT (2868 TPs and 405 FPs), which captures more relationships at the cost of increased false positives. 100 FPs were common to both.

Although the precision achieved by CAIDA and BGPKIT is already fairly high, 7 and 14 false positives (for CAIDA and BGPKIT, respectively) are confirmed as true positives following updates that Customer ASes made to their ASPA objects between March 6th and 31st. This suggests that the values reported in Table 4 are likely lower bounds.

Recall tells a different story: both datasets miss a substantial fraction of declared providers when evaluated against **ASPA**, with CAIDA’s recall (0.54) significantly lower than BGPKIT’s (0.71). This is partly expected, as many customer-to-provider relationships declared in ASPA objects are not detectable from BGP data alone. Indeed, we observe an improvement in recall when comparing inferred relationships with **ASPA<sub>vis</sub>**—particularly for BGPKIT, whose recall rises to 0.93—suggesting that a large share of missed providers were simply not observable in available BGP data rather than genuinely missed by the inference algorithm. The lower recall achieved by CAIDA further corroborates its stricter threshold for classifying a relationship between two ASes as customer-to-provider. Overall, BGPKIT achieves higher recall and F1 score against both reference sets, suggesting that its inference approach captures customer-provider relationships more completely than CAIDA’s. The minor drop (−0.01) in precision achieved by BGPKIT against **ASPA<sub>vis</sub>** is discussed later in the section.

Recall that **BGPKIT** is defined as the union of providers inferred across the five days. To verify whether the union-based logic gives BGPKIT’s inference algorithm an advantage over CAIDA’s, we repeated the analysis by considering two alternative definitions for

**Table 4: Precision, recall, and F1 score achieved by CAIDA and BGPKIT inference algorithms, evaluated against ASPA and ASPA<sub>vis</sub> as reference sets.**

| Dataset | Reference                 | Precision | Recall | F1   |
|---------|---------------------------|-----------|--------|------|
| CAIDA   | <b>ASPA</b>               | 0.88      | 0.54   | 0.67 |
| CAIDA   | <b>ASPA<sub>vis</sub></b> | 0.88      | 0.71   | 0.79 |
| BGPKIT  | <b>ASPA</b>               | 0.88      | 0.71   | 0.78 |
| BGPKIT  | <b>ASPA<sub>vis</sub></b> | 0.87      | 0.93   | 0.90 |

**BGPKIT**: (1) only the providers inferred on a single day, and (2) the *intersection* of providers inferred across the five days. In both cases, BGPKIT achieves very similar accuracy scores to the ones reported in Table 4. When compared to **ASPA** and **ASPA<sub>vis</sub>**, the F1 score decreases by 0.01 for (1), consistently across all five days, and by 0.02 for (2). These scores remain substantially higher than CAIDA’s (0.67 and 0.79), indicating that BGPKIT’s higher accuracy is not a direct consequence of the union-based definition of **BGPKIT**.

Lastly, we analyze inference accuracy on a per-AS basis. Figure 12 shows four heatmaps—one for each combination of inferred and reference provider sets—where each cell  $(x, y)$  reports the number of Customer ASes for which the inference yields  $x$  false negatives and  $y$  false positives. Customer ASes for which the inferred set perfectly matches the reference set appear at  $(0, 0)$ . Because for some Customer ASes the numbers of false positives and false negatives can be as large as 9 and 64 respectively, the figure aggregates them when they are greater than or equal to 5.

When the reference set shifts from **ASPA** to **ASPA<sub>vis</sub>**, the distribution of Customer ASes moves towards the first column (i.e., fewer false negatives); this is consistent with the larger recall previously observed in Table 4. Because **ASPA<sub>vis</sub>** only removes ASPA-declared providers that are not in the available AS Paths, the false positives remain largely unchanged, with the exception of 1 and 3 Customer ASes, respectively for CAIDA and BGPKIT (the latter being the reason for the minor decrease of precision previously documented). The heatmaps also highlight different behaviors between the two inference algorithms: CAIDA tends to produce fewer false positives than BGPKIT, suggesting again a more conservative approach to assigning customer-to-provider relationships. Against **ASPA<sub>vis</sub>**, BGPKIT achieves a perfect match for 739 Customer ASes ( $\approx 68\%$ ), and, for a further 216, it correctly identifies all declared providers while introducing one or more false positives. CAIDA, instead, yields a perfect match for only 580 Customer ASes ( $\approx 53\%$ ).

**Take Away:** *Inference algorithms achieve high precision against ASPA-declared providers; recall is limited by the incomplete view of the global routing ecosystem provided by publicly available BGP data. Many false positives point to incomplete or misconfigured ASPA objects rather than inference error.*

## 6 Data sources

ASPA objects are the latest addition to the RPKI, which was standardized in 2012 [32]. Since then, several data sources and measurement tools have been developed around the RPKI, historically focused on ROA objects but increasingly covering ASPA data as well.

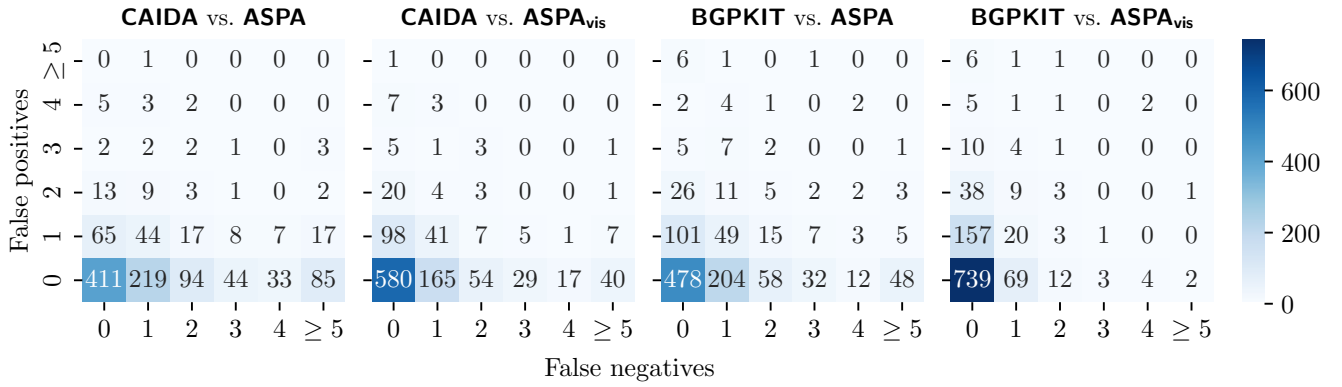


**Figure 12: Distributions of Customer ASes based on the number of false negatives (x-axis) and false positives (y-axis) for each combination of inferred and reference provider set.**

**Table 5: Data sources for ASPAs.**

| Source      | ASPA Objects | Historical | Frequency |
|-------------|--------------|------------|-----------|
| RP software | Always       | No         | On demand |
| RIPE NCC    | Optional     | Yes        | Daily     |
| RPKIViews   | Always       | Yes        | 15 mins   |
| Cloudflare  | Never        | Yes        | Daily     |

In this section, we provide a comprehensive overview of the currently available sources of ASPA data, including what each provides, its temporal granularity, and who operates it. Table 5 summarizes the key features of each data source: whether it contains the raw ASPA objects, has access to historical data, and the frequency at which ASPA data is gathered.

The most direct sources of ASPA data are the RPKI repositories operated by the five Regional Internet Registries (RIRs). These repositories are publicly accessible via `rsync` or the RPKI Repository Delta Protocol (RRDP), and contain the full set of published RPKI objects, including ASPAs. However, retrieving and validating these objects directly requires significant effort; in practice, operators use RP software, which handles fetching from both RIR and delegated repositories, cryptographic validation, and local caching out of the box. The two most popular RP software are `rpki-client` [44] (developed by the OpenBSD Project) and `Routinator` [43] (developed by NLnet Labs), both of which are actively maintained and available as Docker images. They introduced support for fetching and validating ASPA objects in September 2022 (`rpki-client` 8.0) and September 2023 (`Routinator` 0.13), respectively.

Running an RP software directly provides the most current view of the RPKI, but yields no historical data on its own. RIPE NCC maintains a public archive of daily `Routinator` dumps, each containing both the raw cached RPKI objects and the corresponding validated payloads, providing a longitudinal record of RPKI data. ASPA objects have been included since October 10, 2023 [42]. Each daily snapshot is available either as a full dump of the cached RPKI objects ( $\approx 200$  MB in `.tar.xz` format) or as a lighter file containing only validated payloads ( $\approx 1$  MB in `.json.xz` format).

Similarly, the RPKIViews project [54] hosts multiple public repositories containing dumps of `rpki-client`'s output at sub-hourly intervals, which have been gradually reduced from 30 minutes to roughly 15 to 10 minutes today, depending on the specific repository. Although RPKIViews repositories have included ASPA objects since July 2023—three months before the RIPE NCC archive—its snapshots are considerably larger ( $\approx 500$  MB in `.tgz` format) than those from the RIPE NCC Archive. RPKIViews is also piloting the recently proposed `rpkipool` format [57], an experimental, space-efficient format for archiving RPKI data and validation results.

Lastly, in February 2026, Cloudflare extended its Radar API with three ASPA-specific endpoints [40], presumably relying on `rpki-client` to retrieve and validate RPKI data: (1) *snapshot*: returns the set of VAPs, optionally filtered by ASN, on any day after October 1st, 2023. (2) *changes*: reports, on a per-day basis, the number of newly created and deleted ASPA objects, as well as changes in ASPA objects (i.e., additions or removals of providers within an existing ASPA object). (3) *time series*: provides a daily count of Customer ASes, enabling longitudinal analysis of deployment growth.

Unlike the archives described above, the Cloudflare Radar API offers a lightweight, queryable interface of pre-aggregated data. Notably, the *changes* endpoint can be used by any AS to monitor whether non-customer ASes erroneously declare them as a provider, thus reducing route leak detection capabilities [3].

To understand how many ASPA-related RPKI events might go unnoticed when sampling RPKI data daily rather than multiple times an hour, we conducted the following experiment. We selected three one-week periods—the first week of February 2024, February 2025, and February 2026—representing different stages of ASPA deployment, and analyzed `rpki-client` output dumped at sub-hourly intervals. For each day, we looked for two kinds of events:

**Undetected ASPA:** a valid ASPA object that is present in both the first and last dump of the day but absent from one or more intermediate dumps, and

**Undetected change:** a provider set that is identical in the first and last dump of the day but undergoes one or more changes in between.

No undetected ASPA objects are found across all three periods. Undetected provider changes are likewise absent in 2024 and 2025, when the number of valid ASPA objects was small (approximately 60 and 90, respectively). In February 2026, with over 900 valid ASPA objects, undetected provider changes occur on 5 of the 7 days, affecting one or two ASes per day and typically involving the addition or removal of one to a few providers mid-day. Daily sampling is therefore sufficient for most longitudinal analyses, but sub-hourly data may be warranted when studying the fine-grained dynamics of individual ASPA updates.

## 7 Related work

Although ASPA remains an Internet Draft, it has already attracted considerable attention, with existing work falling into two broad categories: evaluating ASPA’s effectiveness in simulated environments, and incorporating ASPA objects into otherwise ASPA-independent route leak detection methodologies. This work instead focuses on the adoption of ASPA from 1505 Customer ASes and on adapting the ASPA AS Path verification procedure to route collector data.

Several studies have evaluated the effectiveness of ASPA in detecting route leaks and compared it with alternative BGP security proposals [6, 19, 28, 50, 63]. The most thorough analysis [19] shows that ASPA matches the Only-To-Customer (OTC) attribute [4] in defending against accidental route leaks while providing stronger guarantees against intentional ones, and further demonstrates its effectiveness against several path manipulation attacks. Although earlier work found ASPA ineffective under partial deployment [50, 63], more comprehensive simulation scenarios yield the opposite conclusion [19]. In contrast, our work presents the first empirical analysis of ASPA deployment in the RPKI, characterizing how network operators have adopted it in practice.

Route leaks occur frequently and have been responsible for several major Internet disruptions, motivating a long line of research on detection methods [24, 47, 52, 61, 64]. Two recent proposals are particularly relevant to our work, as they combine inferred AS relationships with ASPA to improve detection. ASIRIA [5] infers AS relationships from Internet Routing Registry (IRR) data and detects leaks by identifying violations of the valley-free property; the framework is designed to integrate with ASPA, using inferred relationships as a fallback when ASPA information is unavailable. PathProb [25] takes a probabilistic approach, assigning relationship distributions to AS links and computing a continuous legitimacy score per path rather than a binary classification; it similarly incorporates ASPA-declared relationships where available, resorting to probabilistic inference otherwise. Our work is distinct in scope: rather than proposing a detection methodology, we apply the ASPA verification procedure [3] to publicly available AS Paths in which the BGP role under which a route was exchanged is unknown. We show that ASPA objects can partially resolve this ambiguity, and present a verification procedure that returns *N/A* where the algorithm choice remains unresolvable. Furthermore, our work compares ASPA-declared provider relationships against established AS relationship inference datasets and assesses the extent to which they align.

## 8 Conclusions and discussion

In this paper, we presented a first longitudinal study of ASPA deployment, including its practical implications for route leak detection and evaluation of AS relationship inference algorithms. We tracked the evolution of ASPA objects over the past 2.5 years, observing a total of 1646 Customer ASes; ever since RIPE NCC and ARIN enabled ASPA in their respective RPKI dashboards, adoption has been rapidly increasing, and the remaining RIRs are expected to follow in 2026 [58].

Our results provide evidence that ASPA is not yet uniformly well understood. As we observed for some members of AMS-IX, some ASes, as part of their internal BGP practices, manually prepend the ASN of a transparent Route Server to received paths, causing them to be classified as Invalid upon ASPA verification. Over March 2026, ASPA object updates resolved a share of previously invalid paths; moreover, providers inferred by CAIDA and BGPKIT but initially absent from the corresponding ASPAs were subsequently added during the same period, suggesting that a fraction of apparent inference errors in fact reflected incomplete ASPA listings. As adoption grows and the ASPA AS Path verification procedure moves closer to production deployment, systematic evaluation of ASPA correctness becomes increasingly important: an incomplete provider listing may isolate the affected AS and its entire customer base, while an oversized one reduces the capacity of route leak detection. Developing methodologies to validate deployed ASPA objects is therefore an important direction for future work.

One way to evaluate ASPA object correctness is to analyze publicly available AS Paths to identify possible route leaks. This work proposes Algorithm 1 to validate any AS Path—regardless of where it was collected—in a deterministic manner. Appendix B complements this with a heuristic that uses inferred AS Relationships to determine path direction (upstream or downstream). Future work could instead infer the BGP role that an AS has toward each Route Collector it feeds, and use this to perform per-session ASPA validation.

Beyond route leak detection, ASPA provides a degree of protection against path manipulation attacks such as BGP hijacks, which bypass origin validation through ROV [3]. However, ASPA’s verification model leaves peer-to-peer links at the apex of valley-free paths unattested. ASRA (Autonomous System Relationship Authorization) [21], a complementary RPKI extension currently in early draft stages, would allow ASes to cryptographically authorize their lateral peers and customers, directly closing this gap. Combined, ASPA and ASRA could enable near-complete verification of AS Path structure and substantially strengthen the Internet’s defenses against a broad range of routing attacks. However, their adoption will not come immediately and is complicated by technological obstacles, as this paper testifies. It is nonetheless crucial to foster their adoption, as partial deployment reduces their benefits, and to continuously monitor, measure, and evaluate them.

## References

- [1] AMS-IX. 2026. AMS-IX Route Servers. <https://www.ams-ix.net/ams/documentation/ams-ix-route-servers>. Accessed: 2026-08-18.
- [2] Maria Apostolaki, Gian Marti, Jan Müller, and Laurent Vanbever. 2019. SABRE: Protecting Bitcoin against Routing Attacks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Internet Society, San Diego, CA, USA.

- [3] Alexander Azimov, Eugene Bogomazov, Randy Bush, Keyur Patel, Job Snijders, and Kotikalapudi Sriram. 2026. *BGP AS\_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects*. Internet-Draft draft-ietf-sidrops-aspa-verification-25. Internet Engineering Task Force. Work in Progress. <https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/25/>
- [4] Alexander Azimov, Eugene Bogomazov, Randy Bush, Keyur Patel, and Kotikalapudi Sriram. 2022. Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages. RFC 9234. <https://www.rfc-editor.org/info/rfc9234>
- [5] Marcelo Bagnulo, Alberto Garcia-Martinez, Stefano Angieri, Andra Lutu, and Jinze Yang. 2022. Practicable route leak detection and protection with ASIRIA. *Computer Networks* 211 (2022), 108966.
- [6] Seth Barrett, Calvin Idom, German Zavala Villafuerte, Andrew Byers, and Berk Gulmezoglu. 2024. Ain't How You Deploy: An Analysis of BGP Security Policies Performance Against Various Attack Scenarios with Differing Deployment Strategies. In *2024 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 1–8.
- [7] BGPKIT. [n. d.]. BGPKIT as2rel data. <https://data.bgpkit.com/as2rel>. Accessed on March 31st, 2026.
- [8] CAIDA. 2026. The CAIDA AS Organizations Dataset, April 2026. <https://www.caida.org/catalog/datasets/as-organizations>
- [9] CAIDA. 2026. The CAIDA AS Relationships Dataset, March 2026. <https://www.caida.org/catalog/datasets/as-relationships/>
- [10] Nikolaos Chatzis, Georgios Smaragdakis, Anja Feldmann, and Walter Willinger. 2013. There is more to IXPs than meets the eye. *ACM SIGCOMM Computer Communication Review* 43, 5 (2013), 19–28.
- [11] Cloudflare. 2026. New routing widgets on Cloudflare Radar. <https://developers.cloudflare.com/changelog/post/2026-05-04-radar-routing-widgets>. "Cloudflare Docs".
- [12] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. 2016. Jump-starting BGP security with path-end validation. In *Proceedings of the 2016 ACM SIGCOMM Conference*. 342–355.
- [13] Ben Cox. [n. d.]. How do we calculate Down/Upstreams? <https://bgp.tools/kb/what-is-a-upstream>. Accessed on March 31st, 2026.
- [14] Ties Dirksen. 2026. Internship report: Who's running all those tiny RPKI servers? <https://www.sidnlabs.nl/en/news-and-blogs/internship-report-whos-running-all-those-tiny-rpki-servers>
- [15] Ben Du, Gautam Akiwate, Thomas Krenc, Cecilia Testart, Alexander Marder, Bradley Huffaker, Alex C Snoeren, and Kimberly C Claffy. 2022. IRR Hygiene in the RPKI Era. In *International Conference on Passive and Active Network Measurement*. Springer, 321–337.
- [16] Ben Du, Katherine Izhikevich, Sumanth Rao, Guatam Akiwate, Cecilia Testart, Alex C Snoeren, and KC Claffy. 2023. Irregularities in the internet routing registry. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. 104–110.
- [17] Mauro Farina, Martino Trevisan, and Alberto Bartoli. 2025. Poster: Evaluating CAIDA's Inferred AS Relationships Through ASPAs. In *Proceedings of the 21st International Conference on emerging Networking EXperiments and Technologies*. 7–8.
- [18] Mauro Farina, Martino Trevisan, and Alberto Bartoli. 2026. Code and Data for "A First Look at ASPA: Deployment, Route Leaks, and AS Relationships". <https://github.com/mauro-farina/first-look-aspa>.
- [19] Justin Furuness, Cameron Morris, Reynaldo Morillo, Arvind Kasiliya, Bing Wang, and Amir Herzberg. 2025. Securing BGP ASAP: ASPA and other Post-ROV Defenses. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Internet Society, San Diego, CA, USA.
- [20] Lixin Gao. 2001. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on networking* 9, 6 (2001), 733–745.
- [21] Nan Geng, Kotikalapudi Sriram, and Mingqing(Michael) Huang. 2026. *A Profile for Autonomous System Relationship Authorization (ASRA)*. Internet-Draft draft-geng-sidrops-asra-profile-03. Internet Engineering Task Force. Work in Progress. <https://datatracker.ietf.org/doc/draft-geng-sidrops-asra-profile/03/>
- [22] Phillipa Gill, Michael Schapira, and Sharon Goldberg. 2013. A survey of interdomain routing policies. *ACM SIGCOMM Computer Communication Review* 44, 1 (2013), 28–34.
- [23] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and KC Claffy. 2014. Inferring complex AS relationships. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. 23–30.
- [24] Vasileios Giotsas and Shi Zhou. 2012. Valley-free violation in Internet routing – Analysis based on BGP Community data. In *2012 IEEE International Conference on Communications (ICC)*. 1193–1197.
- [25] Yingqian Hao, Hui Zou, Lu Zhou, Yuxuan Chen, and Yanbiao Li. 2026. PathProb: Probabilistic Inference and Path Scoring for Enhanced and Flexible BGP Route Leak Detection. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Internet Society, San Diego, CA, USA.
- [26] Sichang He, Italo Cunha, and Ethan Katz-Bassett. 2024. RPSlyzer: Characterization and Verification of Policies in Internet Routing Registries. In *Proceedings of the 2024 ACM on Internet Measurement Conference*. 365–374.
- [27] Rahul Hiran, Niklas Carlsson, and Phillipa Gill. 2013. Characterizing large-scale routing anomalies: A case study of the china telecom incident. In *International Conference on Passive and Active Network Measurement*. Springer, 229–238.
- [28] Nils Höger, Nils Rodday, and Gabi Dreö Rodosek. 2025. Mitigating BGP route leaks with attributes and communities: A stopgap solution for path plausibility. *International Journal of Network Management* 35, 2 (2025), e70002.
- [29] Elisa Jasinska, Nick Hilliard, Robert Raszuk, and Niels Bakker. 2016. Internet Exchange BGP Route Server. RFC 7947. <https://www.rfc-editor.org/info/rfc7947>
- [30] Yuchen Jin, Colin Scott, Amogh Dhamdhere, Vasileios Giotsas, Arvind Krishnamurthy, and Scott Shenker. 2019. Stable and practical {AS} relationship inference with {ProbLink}. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. 581–598.
- [31] Zitong Jin, Xingang Shi, Yan Yang, Xia Yin, Zhiliang Wang, and Jianping Wu. 2020. Toposcope: Recover as relationships from fragmentary observations. In *Proceedings of the ACM Internet Measurement Conference*. 266–280.
- [32] Matt Lepinski and Stephen Kent. 2012. An Infrastructure to Support Secure Internet Routing. RFC 6480. <https://www.rfc-editor.org/info/rfc6480>
- [33] Weitong Li, Zhexiong Lin, Md Ishtiaq Ashiq, Emile Aben, Romain Fontugne, Amreesh Phokeer, and Taejoong Chung. 2023. RoVista: Measuring and analyzing the route origin validation (ROV) in RPKI. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. 73–88.
- [34] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and KC Claffy. 2013. AS relationships, customer cones, and validation. In *Proceedings of the 2013 conference on Internet measurement conference*. 243–256.
- [35] Doug Madory. 2019. Large European Routing Leak Sends Traffic Through China Telecom. <https://manrs.org/2019/06/large-european-routing-leak-sends-traffic-through-china-telecom>. "MANRS Blog".
- [36] MANRS. [n. d.]. MANRS Implementation Guide for Network Operators. <https://manrs.org/netops/guide>. Version 1.1 (updated May 12th, 2021).
- [37] MANRS. 2023. Estimating the Timeline for ASPA Deployment. <https://manrs.org/2023/05/estimating-the-timeline-for-aspa-deployment>.
- [38] Mark Koster. 2026. New Features Added to ARIN Online. <https://www.arin.net/announcements/20260120>. ARIN announces ASPA availability in ARIN Online.
- [39] Tyler McDaniel, Jared M Smith, and Max Schuchard. 2020. Flexsealing BGP against route leaks: peerlock active measurement and analysis. *arXiv preprint arXiv:2006.06576* (2020).
- [40] Bryton Herdes Mingwei Zhang. 2026. ASPA: making Internet routing more secure. <https://blog.cloudflare.com/aspa-secure-internet>. "Cloudflare Blog".
- [41] Prodosh Mohapatra, John Scudder, David Ward, Randy Bush, and Rob Austein. 2013. BGP Prefix Origin Validation. RFC 6811. <https://www.rfc-editor.org/info/rfc6811>
- [42] RIPE NCC. 2026. RPKI repository archive. <https://github.com/RIPE-NCC/internet-dataset-descriptions/blob/main/rpki-repo-archive.md#2023-10-10>.
- [43] NLnet Labs. 2026. Routinator. <https://routinator.docs.nlnetlabs.nl/en/stable>. Accessed: 2026-03-11.
- [44] OpenBSD Project. 2026. rpki-client. <https://www.rpki-client.org>. Accessed: 2026-03-11.
- [45] Tom Paseka. 2018. How a Nigerian ISP Accidentally Knocked Google Offline. <https://blog.cloudflare.com/how-a-nigerian-isp-knocked-google-offline>. "Cloudflare Blog".
- [46] Lars Prehn and Anja Feldmann. 2021. How biased is our validation (data) for AS relationships? In *Proceedings of the 21st ACM Internet Measurement Conference*. 612–620.
- [47] Sophie Y Qiu, Patrick D McDaniel, and Fabian Monrose. 2007. Toward valley-free inter-domain routing. In *2007 IEEE International Conference on Communications*. IEEE, 2009–2016.
- [48] RIPE NCC [n. d.]. RIPE NCC RISwhois dataset. <https://ris.ripe.net/docs/ris-whois>. Accessed on March 31st, 2026.
- [49] RIPE NCC. 2023. RPKI Publication as a Service Available. <https://www.ripe.net/about-us/news/rpki-publication-as-a-service-available/>. Accessed August 17, 2026.
- [50] Nils Rodday, Gabi Dreö Rodosek, Aiko Pras, and Roland van Rijswijk-Deij. 2024. Exploring the benefit of path plausibility algorithms in BGP. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. IEEE, 1–10.
- [51] Aftab Siddiqui. 2021. A Major BGP Hijack by AS55410-Vodafone Idea Ltd. <https://manrs.org/2021/04/a-major-bgp-hijack-by-as55410-vodafone-idea-ltd>. "MANRS Blog".
- [52] M. S. Siddiqui, D. Montero, M. Yannuzzi, R. Serral-Gracià, and X. Masip-Bruin. 2014. Route leak identification: A step toward making inter-domain routing more reliable. In *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN)*. 1–8.
- [53] Jared M Smith and Max Schuchard. 2018. Routing around congestion: Defeating DDoS attacks and adverse network conditions via reactive BGP routing. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 599–617.
- [54] Job Snijders. 2026. RPKIViews. <https://www.rpkiviews.org/>.
- [55] Job Snijders, Alexander Azimov, Eugene Uskov, Randy Bush, Russ Housley, and Ben Maddison. 2026. *A Profile for Autonomous System Provider Authorization*. Internet-Draft draft-ietf-sidrops-aspa-profile-26. Internet Engineering Task Force. Work in Progress. <https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-profile/26/>

- [56] Job Snijders, Ben Maddison, Matt Lepinski, Derrick Kong, and Stephen Kent. 2024. A Profile for Route Origin Authorizations (ROAs). RFC 9582. <https://www.rfc-editor.org/info/rfc9582>
- [57] Job Snijders and Fedor Vomppe. 2026. *The RPKISPOOL Format for Materializing Resource Public Key Infrastructure (RPKI) Data*. Internet-Draft draft-snijders-rpkispool-format-00. Internet Engineering Task Force. Work in Progress. <https://datatracker.ietf.org/doc/draft-snijders-rpkispool-format/00/>
- [58] Sofia Silva Berenguer. 2025. NRO RPKI Program - 2025 in Review. [https://labs.ripe.net/author/sofia\\_silva\\_berenguer/nro-rpki-program-2025-in-review](https://labs.ripe.net/author/sofia_silva_berenguer/nro-rpki-program-2025-in-review). RIPE Labs article.
- [59] Kotikalapudi Sriram, Doug Montgomery, Danny R. McPherson, Eric Osterweil, and Brian Dickson. 2016. Problem Definition and Classification of BGP Route Leaks. RFC 7908. <https://www.rfc-editor.org/info/rfc7908>
- [60] Tom Strickx. 2019. How Verizon and a BGP Optimizer Knocked Large Parts of the Internet Offline Today. <https://blog.cloudflare.com/how-verizon-and-a-bgp-optimizer-knocked-large-parts-of-the-internet-offline-today>. "Cloudflare Blog".
- [61] L. Subramanian, S. Agarwal, J. Rexford, and R.H. Katz. 2002. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2. 618–627 vol.2.
- [62] Tim Bruijnzeels. 2025. ASPA support enabled in the RIPE NCC RPKI Dashboard. [https://mailarchive.ietf.org/arch/msg/sidrops/9L8pdng57m\\_4Y1347jDERTXefXI](https://mailarchive.ietf.org/arch/msg/sidrops/9L8pdng57m_4Y1347jDERTXefXI). Message to the IETF SIDROPS mailing list.
- [63] Naoki Umeda, Taiji Kimura, and Naoto Yanai. 2023. The juice is worth the squeeze: Analysis of autonomous system provider authorization in partial deployment. *IEEE Open Journal of the Communications Society* 4 (2023), 269–306.
- [64] Yuancheng Xie, Zhaoxin Zhang, Ning Li, and Haoyang Gao. 2025. LeakFocus: Catching the perpetrator in routing leak event. *Computers & Security* 150 (2025), 104300. doi:10.1016/j.cose.2024.104300

## A ASPA AS Path verification under partial ASPA adoption

This appendix provides further details regarding the upstream and downstream ASPA AS Path verification algorithms [3]. Specifically, we cover their handling of partial ASPA adoption and the case of non-transparent route servers.

The ASPA specification defines two distinct verification algorithms, each applied based on the BGP role of the neighboring AS that sent the route [3]. The *upstream verification algorithm* applies to routes received from a customer, a peer, or a route server: since such routes are already traveling up the hierarchy toward the verifying AS, the received path must consist entirely of an up-ramp, and any provider-to-customer or peer-to-peer link is an immediate indicator of a route leak. The *downstream verification algorithm* applies to routes received from a provider: since such routes are traveling down the hierarchy, the path may legitimately include both an up-ramp and a down-ramp connected by at most one peer-to-peer link at their apex. In both cases, the verifying AS applies the algorithm to the *compressed* AS Path—the sequence of unique ASNs after removing consecutive duplicates—received from its neighbor.

Both algorithms are built on a *provider authorization function*,  $\text{authorized}(AS_x, AS_y)$ , which checks whether  $AS_y$  is an attested provider of  $AS_x$  according to the VAPs available to the router. The function returns one of three outcomes:

- *Provider+*:  $AS_x$  has a valid ASPA object and  $AS_y$  is listed among its authorized providers.
- *Not Provider+*:  $AS_x$  has a valid ASPA object but  $AS_y$  is not listed among its authorized providers.
- *No Attestation*:  $AS_x$  has no valid ASPA object, so no claim can be made about the relationship.

The distinction between the last two outcomes is critical under partial ASPA adoption: a *Not Provider+* result is definitive: the hop is attested to not be a customer-to-provider link, whereas a

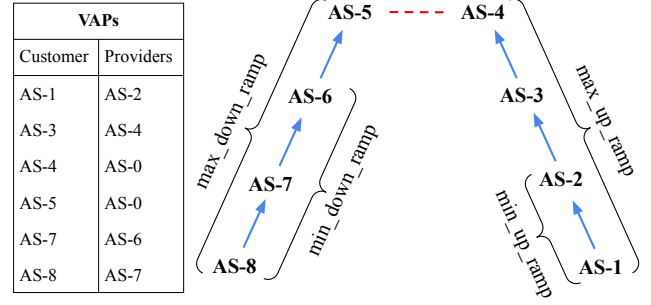


**Figure 13: Illustration of an AS Path of length 8, along with the computed min- / max-, up- / down- ramps based on the available VAPs. Note that AS-2 and AS-6 are not associated with any VAP.**

*No Attestation* result is ambiguous: the hop may or may not be customer-to-provider. Thus, the algorithms operate on *bounds* of the up-ramp and down-ramp lengths rather than exact values.

Consider the AS Path illustrated in Figure 13 sent by AS-8 to AS-9, as well as the VAPs available to the receiving BGP router of AS-9. Using the  $\text{authorized}()$  function, the router computes the upper and lower bounds of both up-ramp and down-ramp. The up-ramp bounds are computed by walking the path from the origin AS-1 upward. The *minimum up-ramp* ( $\text{min\_up\_ramp}$ ) advances as long as each hop is attested as *Provider+*, stopping at the first hop that is either *Not Provider+* or *No Attestation*; it thus represents the length of the up-ramp that can be confirmed. The *maximum up-ramp* ( $\text{max\_up\_ramp}$ ) is more permissive: it treats *No Attestation* hops as potentially valid and advances until it encounters a definitive *Not Provider+*; it thus represents the length of the up-ramp that cannot be ruled out.

Symmetrically, the down-ramp bounds ( $\text{min\_down\_ramp}$  and  $\text{max\_down\_ramp}$ ) are computed by walking the path backward from AS-8 (the AS that was most recently added to the path) toward AS-1, applying the same logic. Intuitively, the maximum bounds reflect how far each ramp could plausibly extend given the available VAPs, while the minimum bounds capture how far it is fully attested.

Algorithms 2 and 3 formalize the upstream and downstream verification procedures, respectively. Both take as input a compressed AS Path and the available VAPs, and return either *Valid*, *Unknown*, or *Invalid* based on whether the ramp bounds are consistent with a valley-free path.

Lastly, Figure 14 shows an example of how the ASPA AS Path Verification procedure can detect a route leak despite some uncertainty (i.e., the lack of VAPs associated with two ASes in the path, AS-2 and AS-6). Illustration (a) represents the VAPs available to the verifying AS, which are valid across all (b), (c), and (d). Because the maximum up and down ramps terminate once they reach AS-3 and AS-4, respectively, whether the leaking AS-9x is associated with a VAP is irrelevant. Both upstream and downstream verification will return 'Invalid', despite the lack of VAPs for AS-2 and AS-6, as the two more permissive ramps ( $\text{max\_up\_ramp}$  and  $\text{max\_down\_ramp}$ ) do not cover the entire path.

**Algorithm 2** Upstream verification algorithm.

**Require:** Compressed AS Path  $P = \langle AS_n, \dots, AS_2, AS_1 \rangle$  of length  $n$ , where  $AS_n$  is the neighbor and  $AS_1$  is the origin; Validated ASPA Payloads VAPs

**Ensure:** Verification outcome: *Valid*, *Unknown*, or *Invalid*

```

1: if  $n = 0$  then                                ▶ Check if the path is empty
2:   return Invalid
3: end if
4: Compute  $min\_up\_ramp$  and  $max\_up\_ramp$  from  $P$  using VAPs
5: if  $max\_up\_ramp < n$  then
6:   return Invalid
7: else if  $min\_up\_ramp < n$  then
8:   return Unknown
9: else
10:  return Valid
11: end if

```

**Algorithm 3** Downstream verification algorithm.

**Require:** Compressed AS Path  $P = \langle AS_n, \dots, AS_2, AS_1 \rangle$  of length  $n$ , where  $AS_n$  is the neighbor and  $AS_1$  is the origin; Validated ASPA Payloads VAPs

**Ensure:** Verification outcome: *Valid*, *Unknown*, or *Invalid*

```

1: if  $n = 0$  then                                ▶ Check if the path is empty
2:   return Invalid
3: end if
4: Compute  $min\_up\_ramp$ ,  $max\_up\_ramp$ ,  $min\_down\_ramp$ ,
    $max\_down\_ramp$  from  $P$  using VAPs
5: if  $max\_up\_ramp + max\_down\_ramp < n$  then
6:   return Invalid
7: else if  $min\_up\_ramp + min\_down\_ramp < n$  then
8:   return Unknown
9: else
10:  return Valid
11: end if

```

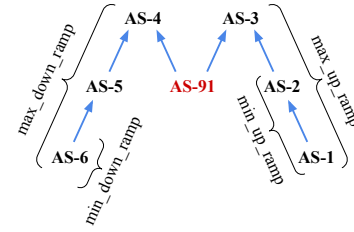
**B Understanding the N/A outcome**

As explained in Section 4.1, paths are labelled as N/A when the upstream and downstream algorithms yield different outcomes, meaning that the actual outcome depends on whether the path is following an upstream (e.g., from customer to provider) or downstream (e.g., from provider to customer) direction. In this appendix, we infer the path direction for the 13.8 million paths that Algorithm 1 labelled as N/A using inferred AS relationships.

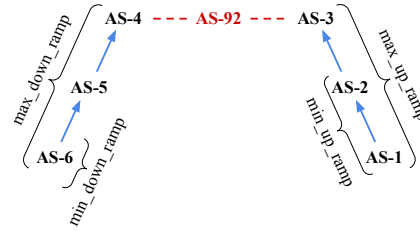
Because these can be inaccurate (as shown in Section 5), we separate this analysis from the main manuscript, where all results are derived from the purely deterministic proposed methodology. We use the AS relationships inferred by CAIDA and BGPKIT following the same methodology detailed in Section 5, and derive, for each AS (not only Customer ASes), its two sets of providers denoted as **CAIDA** and **BGPKIT**. The resulting ASPA-based AS Path verification with the path direction given by an inferred relationship is detailed in Algorithm 4, and it follows a simple logic. If the neighbor is inferred as a provider of the receiver, the path is verified as downstream; otherwise, it is verified as upstream. When the receiver and neighbor are both inferred as providers of each

| VAPs      |      |      |      |      |
|-----------|------|------|------|------|
| Customer  | AS-1 | AS-3 | AS-4 | AS-5 |
| Providers | AS-2 | AS-0 | AS-0 | AS-4 |

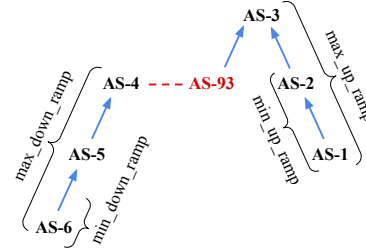
(a) Available VAPs.



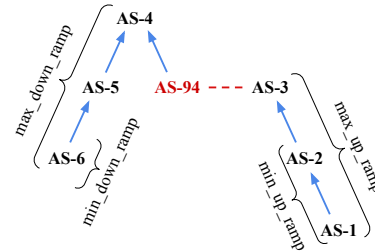
(b) Type 1 route leak: hairpin turn.



(c) Type 2 route leak: lateral leak.



(d) Type 3 route leak: provider prefix to peer.



(e) Type 4 route leak: peer prefix to provider.

**Figure 14: Example of how minimum and maximum ramps are computed on each of the four types of route leak illustrated in Figure 1. AS-9x is the AS causing the route leak.**

**Algorithm 4** AS Path verification procedure with path direction obtained from an inferred AS relationships dataset.

**Require:** Compressed AS Path  $P = \langle AS_n, \dots, AS_2, AS_1 \rangle$  of length  $n$ ; Validated ASPA Payloads  $VAPs$ ; mappings of each AS to its inferred providers **PROV**

**Ensure:** Verification outcome: *Valid*, *Unknown*, *Invalid*, or *Unresolved*

```

1: receiver  $\leftarrow AS_n$ 
2: neighbor  $\leftarrow AS_{n-1}$ 
3: received  $\leftarrow \langle AS_{n-1}, \dots, AS_1 \rangle$ 
4: if neighbor  $\in \text{PROV}(\text{receiver})$  and receiver  $\in \text{PROV}(\text{neighbor})$  then  $\triangleright$  Mutually inferred as providers
5:   return Unresolved
6: else if neighbor  $\in \text{PROV}(\text{receiver})$  then  $\triangleright$  Downstream path
7:   return DOWNSTREAMVERIFICATION(received, VAP)
8: else  $\triangleright$  Upstream path (from customer or peer)
9:   return UPSTREAMVERIFICATION(received, VAP)
10: end if
```

**Table 6: Outcome of Algorithm 4 applied to the 13 814 343 paths labeled *N/A* by Algorithm 1.**

|            | CAIDA      |         | BGPKIT     |         |
|------------|------------|---------|------------|---------|
|            | # Paths    | % Paths | # Paths    | % Paths |
| Valid      | 2 459 495  | 17.8    | 3 398 401  | 24.6    |
| Invalid    | 381 258    | 2.8     | 251 845    | 1.8     |
| Unknown    | 10 972 644 | 79.4    | 10 014 537 | 72.5    |
| Unresolved | 946        | 0.0     | 149 560    | 1.1     |

other (i.e., a complex relationship), direction cannot be resolved

and the path is labelled as *Unresolved*. The algorithm does not take into account a complex relationship where the neighbor may act as a provider or as a peer depending on the specific prefix (the third row in Table 2), making the outcome obtained from downstream verification even less reliable.

We apply Algorithm 4 to the 13 814 343 paths labeled *N/A* by Algorithm 1, using first **CAIDA** and then **BGPKIT** providers: results are summarized in Table 6. Due to the small number of available VAPs, most paths remain *Unknown* for both CAIDA (79.4 %) and BGPKIT (72.5 %). Of the 381 258 and 251 845 paths labeled *Invalid* under CAIDA and BGPKIT inference, respectively, only 169 433 are in common.

## C Ethics

This study does not raise ethical concerns and relies exclusively on the following publicly available datasets: (1) Validated ASPA Payloads from the RIPE NCC RPKI Archive as the underlying basis of the entire work, and from RPKIviews to evaluate sub-daily changes in the content of ASPA objects; (2) IPv4 and IPv6 RISwhois dumps of March 31st, 2026, to associate each Customer AS with its announced IP prefixes; (3) The March 2026 datasets of AS Paths, customer cones, and inferred AS Relationships provided by CAIDA; (4) Route collectors data from RIPE RIS and RouteViews; (5) AS Relationships inferred by BGPKIT between March 1st and 5th, 2026.

The only two data sources that lack historical access are the RISwhois dumps and the AS Relationships inferred by BGPKIT; we release them along with our code to enable the full reproducibility of our work.

## D Reproducibility

The code and data used for this paper are available on GitHub: <https://github.com/mauro-farina/first-look-aspa>.