Securing BGP ASAP: ASPA and other Post-ROV Defenses

Justin Furuness*
University of Connecticut

Cameron Morris*
University of Connecticut

Reynaldo Morillo
University of Connecticut

Arvind Kasiliya
University of Connecticut

Bing Wang
University of Connecticut

Amir Herzberg
University of Connecticut

***Abstract*—Before the adoption of Route Origin Validation (ROV), prefix and subprefix hijacks were the most effective and common attacks on BGP routing. Recent works show that ROV adoption is increasing rapidly; with sufficient ROV adoption, prefix and subprefix attacks become ineffective. We study this changing landscape and in particular the Autonomous System Provider Authorization (ASPA) proposal, which focuses on route leakage but also foils some other attacks.**

Using recent measurements of real-world ROV adoption, we evaluate its security impact. Our simulations show substantial impact: *already today*, prefix hijacks are less effective than forged-origin hijacks, and the effectiveness of subprefix hijacks is much reduced. Therefore, we expect attackers to move to forged-origin hijacks and other *post-ROV attacks*; we present a new, powerful post-ROV attack, *first-ASN-stripping*.

We present extensive evaluations of different post-ROV defenses and attacks. Our results show that ASPA significantly protects against post-ROV attacks, even in partial adoption. It dramatically improves upon the use of only ROV or of BGPsec, Path-End, OTC, and EdgeFilter. BGP-iSec has even better protection but requires public-key operations to export/import announcements. We also present ASPAwN, an extension that further improves ASPA's performance. Our results show that contrary to prior works [74], [95], ASPA is effective even when tier-1 ASes are not adopting, hence motivating ASPA adoption at edge and intermediate ASes. On the other hand, we find that against *accidental* route leaks, the simpler, standardized OTC mechanism is as effective as ASPA.

I. Introduction

For over three decades, Autonomous Systems (ASes) that constitute the Internet have relied on the Border Gateway Protocol (BGP) [69] to facilitate the exchange of routing information. BGP was developed without security in mind, leading to a plethora of routing attacks that have proven difficult to counteract. Extensive research and standardization efforts have been dedicated to securing BGP (see §X). Among them, *Resource Public Key Infrastructure (RPKI)* [17], [49], [96] is an IETF standardized approach against *prefix/subprefix hijacks*. In these hijacks, an attacker falsely originates a route to a prefix/subprefix that it is not authorized to announce. Specifically, the goal of RPKI is to provide precise, reliable, updated, and authenticated information on IP prefix origins. RPKI allows the owner of a prefix to identify authorized ASes that can announce the prefix using a signed *Route Origin Authorization (ROA)*. Using ROAs, routers can apply *Route Origin Validation (ROV)* to detect and discard BGP announcements from unauthorized origins. As a result, RPKI/ROV can effectively prevent prefix/subprefix hijacks.

The NIST RPKI monitor shows increasing adoption of ROAs [11], [65], currently covering nearly 50% of IPv4 address space [65]. In addition, recent measurements [41], [52], [54] show that the adoption of ROV has also been increasing rapidly. In particular, top-tier and upper-level ISPs have been enforcing ROV to filter invalid announcements, which can be particularly effective in preventing prefix/subprefix hijacks [27].

With the current estimated ROV enforcement, are prefix/subprefix hijacks still impactful, or attackers will turn to post-ROV attacks, i.e., attacks not blocked by ROV? We answer this question using extensive simulation over an empirical Internet topology. Our results show that, even with a conservative estimate of ROV enforcement, *already now* post-ROV attacks are more effective than prefix hijacks. Subprefix hijacks are still effective, although much less than before ROV (§VII); the improved *ROV++* [62] would have been much better against subprefix hijacks, but another defense is to simply announce only /24 prefixes, which are the most specific prefixes practically possible, preventing subprefix hijacks. As a result, we expect future attacks to usually be *post-ROV attacks*, i.e., use only ROA-valid announcements and hence evade ROV.

Two general forms of post-ROV attacks are *path manipulation* and *route leaks*. In path manipulation, an attacker forges or modifies the AS-Path for intercepting traffic, while in route leaks, an AS exports an announcement that conflicts with its supposed business model in order to attract traffic. *BGPsec* [53] is the IETF standardized protection against path manipulation attacks. Its deployment, however, faces formidable obstacles, including high computational requirements [9], [13], [47], [67], [89], [101] and limited benefits

*Equal contributors

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.240675
www.ndss-symposium.org

in partial adoption [15], [28], [30], [55], [63]. In addition, BGPsec is not designed to prevent route leaks.

A recently proposed extension to BGPsec, *BGP-iSec* [63], improves benefits in partial adoption and includes defenses against route leaks, but incurs similarly high computational costs. Another recent defense against route leaks is the *Only-To-Customer (OTC)* mechanism (RFC 9234 [5]). OTC, however, only addresses *unintentional* route leaks. As we will describe in §III, ASes adopting OTC add an OTC attribute to announcements sent to customers so that if those announcements are later leaked, they will be dropped when a provider sees them from a customer (since they are forwarded to a provider, violating the OTC attribute).

Autonomous System Provider Authorization (ASPA) [3], [84] is another recently proposed post-ROV defense, focused on route-leak prevention. ASPA works by having each AS publish and sign a list of its providers in the RPKI. Other ASes can then use this information to validate AS paths, which can protect against route leaks (both accidental and intentional) and some forms of path manipulations (see §III). While ASPA is still an Internet-Draft (I-D), it has gained increasing attention, with general availability in commercial off-the-shelf products estimated by 2026 [83].

ASPA appears to be a promising solution to many post-ROV attacks: it only involves a relatively simple extension to RPKI, incurs much less computational overhead than BGPsec, and addresses route leaks. Additionally, unlike OTC, ASPA provides detection and mitigation of improbable AS paths to some extent.

A. Contributions

• *Evaluation of ASPA adoption scenarios showing that ASPA is effective—even when tier-1 ASes do not adopt*. We simulate ASPA under various deployment scenarios and find that ASPA provides effective defense. We also show that ASPA is effective even when no tier-1 ASes are adopting, contrary to conclusions in [74], [95] (§VIII). We explain the reasons for these differences with prior works in §X. Our findings motivate adoption of ASPA at edge and intermediate ASes.

• *Evaluation of ASPA under the highly effective shortest-path export-all attack.* We show that ASPA is significantly less effective against the shortest-path export-all attack than the previously evaluated forged-origin hijack. For ASPA, shortest-path export-all attack becomes more effective than forged-origin hijack when the adoption rate approaches 20%, and the gap is even larger as adoption increases further (§VIII-B). Hence attackers will turn to shortest-path export-all attacks under higher adoption rates (beyond 20%), motivating the need for better defense strategies.

• *First-ASN-Stripping attack*. We present the first-ASN-stripping hijack (§V-B), and show that it is highly effective even when ROV is deployed (see Fig. 8(c)) unless it is prevented by neighboring ASes adopting the `enforce-first-AS` mechanism. We have notified major vendors where `enforce-first-as` is not default or only applicable globally, and received positive responses from them.

• *ASPA with Neighbors (ASPAwN).* We propose and evaluate ASPAwN, a simple, easy-to-deploy extension to ASPA, designed to address a specific security concern in ASPA, i.e., attacks by a provider AS (§IV). We find that this policy is highly effective at protecting the attacker's customer cone (see Fig. 8(b)). A related extension to ASPA called *AS Relationship Authorization (ASRA)* [85] was recently proposed to the IETF; ASRA was developed independently and in parallel to ASPAwN[1]. We believe that ASPAwN has equivalent or similar security properties as one ASRA algorithm (Algorithm B), which has stronger security than the other ASRA algorithm (Algorithm A).

• *Evaluation of the impact of current and future ROV adoption.* We aggregate all published ROV data sources in [14], [41], [52], [54], [72], [75], [78], and use this real-world ROV data to simulate the impact of prefix and subprefix hijacks compared to post-ROV attacks (§VII). Our results demonstrate that attackers will attract more traffic *today* using post-ROV attacks than using prefix hijacks, motivating the need for effective defenses against post-ROV attacks.

• *OTC is as effective as ASPA for preventing accidental route leaks.* We show that, both by simulations (§IX-A) and analytically (Appendix B), ASPA and OTC [5] are identical in effectiveness against *accidental* route leaks. OTC is already deployed in the wild, and we believe that it is easier to deploy than ASPA. Therefore, our results motivate rapid deployment of OTC attributes to prevent accidental route leaks.

• *Extensive evaluation of ASPA and comparison to other defenses.* We simulate the effectiveness of several defenses for BGP, under different adoption percentages, against a variety of path manipulation attacks (forged-origin hijacks, shortest-path export-all hijacks, edge and transit attackers, single and multiple attackers), route leaks (from multihomed and transit ASes), and the aforementioned first-ASN-stripping hijack (§VIII). The defenses include ASPA, ASPAwN, BGP-iSec, Path-End, EdgeFilter, OTC, and combined mechanisms. Our results show that ASPA is effective against both path manipulation and route leaks (malicious and accidental), even under partial adoption, and even when tier-1 ASes are not deploying, contrary to the claims in [74], [95].

• *New open-source software.* We extended the BGPy simulator [23] to include the various defenses and attacks that we evaluate in this paper. In addition, we create a tool to aggregate ROV enforcement measurements from data sources that are publicly available [14], [41], [52], [54], [72], [75], [78]. We open-sourced the BGPy extension and the tool [21], [22].

II. Attack and Routing Models

In this section, we describe attack and routing models. We discuss our ASPA adoption assumptions in §VI.

[1]While there were earlier presentations of ASRA in [25], we were not aware of it until after our submission.

A. Attack model

For all attack scenarios, we assume that the attacker has knowledge of the AS topology. We also assume that the attacker has knowledge regarding which ASes are adopting the defensive policies, which can be obtained using corresponding ROA information, ASPA records, etc. We assume that an attacker can eavesdrop on BGP announcements between other ASes using services like BGP route collectors [42], [73], [76] and looking glass servers. Once an attacker has performed a successful hijack, they can perform a variety of malicious activities such as eavesdropping through man-in-the-middle attacks, impersonating legitimate services for phishing purposes, denial of service attacks, etc.

Attackers are selected from edge ASes (no customers) or transit ASes (has customers) as defined by [70]. Tier-1 ASes are a clique of top-tier ASes that have no providers, and we use the CAIDA AS topology [10] to determine these ASes.

Attacker ASes. The attacker is assumed to control at least one malicious AS from which it can send arbitrary BGP announcements. We also simulate a case where an attacker controls multiple ASes, such as when a nation-state/country performs an attack (see §VIII-B).

B. Routing Model

As in other studies [15], [27], [28], [32], we model the Internet as an AS-graph and consider two types of relationships: customer-provider (customer pays its provider for the transit of traffic) and peer-peer (traffic is exchanged between the two ASes without monetary compensation). Other types of relationship (e.g., sibling) is not considered, similar to prior works [62], [63]. Specifically, we use the following routing assumptions that are commonly used in existing studies.

Valley-free routing. For a given AS-graph, we assume *valley-free* (Gao-Rexford) routing [24]. That is, for any prefix p, an AS forwards the best announcement that it receives from its customers to the neighboring ASes (all customers, and some or all of the providers and peers). If none was received from customers, then it forwards the best announcement that it receives from a peer (or from a provider if no announcement is received from a peer) only to its customers. Following the above policy, no valley will be formed in the routing path, and hence the name 'valley-free'. While this is a simplifying assumption and does not always hold in practice [1], [57], [59], [64], we adopt it for this work (as done in most existing studies), due to a lack of better models.

Path-selection policy. Each AS has a path-selection policy that selects the best path to use for each IP prefix. We assume an AS prefers paths from customers, then from peers, and lastly providers, i.e., 'local preference first', for economic considerations. Second, if two paths have the same relationship, e.g., both are from a customer, peer, or provider, then the AS prefers the shorter path. Otherwise, an AS breaks ties following certain policies (e.g., preferring the one whose next hop has the lowest AS number).

Export policy. Last, an AS uses an export policy that determines what routes (if any) to forward to a neighbor. We assume the widely used and simplifying *export-to-all* policy. That is, an AS sends the preferred announcements to all customers; and if the preferred announcement for a prefix was received from a customer, then it is sent to all neighbors, including providers and peers.

III. ASPA and Other Post-ROV Defenses

As mentioned earlier, ASPA, currently an Internet-Draft (I-D) [3], appears to be a promising approach against post-ROV attacks. In this section, we describe the mechanisms of ASPA and several other post-ROV defenses.

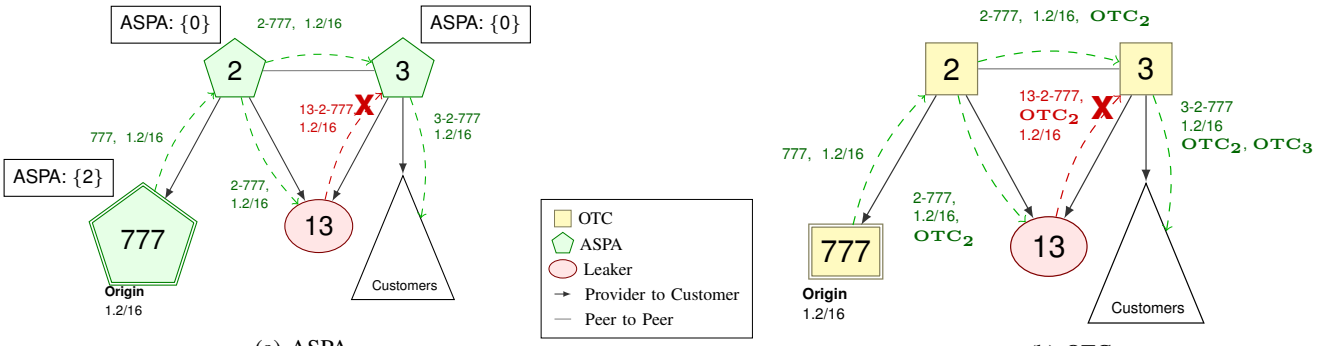
A. AS Provider Authorization (ASPA)

ASPA [3] is a security mechanism designed to provide protection against route leaks. An ASPA-adopting AS publishes a Set of Provider ASes (SPAS) to the RPKI that includes only the ASes authorized to propagate its routes upwards. Using these sets, an AS can verify that an AS path is valley-free and reject paths that appear to be route leaks. Specifically, the verification process ensures that each AS path contains at most one segment where the route propagates upstream to providers followed by at most one segment where it propagates downstream to customers; at the point where the route changes direction from upstream to downstream, it is allowed to be propagated laterally to a peer.

The operation of ASPA is illustrated in Fig. 1(a). In this example, ASes 777, 2, and 3 adopt ASPA. AS 777 is the legitimate origin for prefix 1.2/16. The SPAS of AS 777 includes AS 2, while ASes 2 and 3 have only AS 0 in them to indicate they have no providers. AS 13 leaks an announcement of 1.2/16 that it receives from one provider, AS 2, to another provider, AS 3. Using ASPA, AS 3 detects the route leak because AS 13 is not in the SPAS of AS 2.

In addition to foiling most (accidental or malicious) route leaks, ASPA makes it more difficult for an attacker to announce forged paths. This is because if an AS is not in the set of authorized providers for the previous AS, the route cannot be propagated upward to the attacker's providers and can only be sent downward to customers. For example, in Fig. 1(a), if AS 13 performs a forged-origin hijack with a path of 13-777, then it will be rejected by AS 3 because AS 13 is not a provider of AS 777 (and hence not in the SPAS of AS 777).

Note that ASPA can only validate whether a path is consistent with the published ASPA records. It cannot validate whether an announcement is actually sent along an AS-Path. As an example, in Fig. 1(a), if AS 777 has another provider, AS 4, which does not adopt ASPA, then AS 13 can announce a shortest-path export-all hijack (described in §V-A) with an AS-Path of 13-4-777 to AS 3, and AS 3 will regard it as a valid announcement since AS 4 is in AS 777's SPAS, and will not be able to detect the fake announcement. This is why it is so critical that ASPA be evaluated in the context of shortest-path export-all hijacks, since ASPA does not protect against forged AS paths which are consistent with ASPA records. For



(a) ASPA. (b) OTC.

Fig. 1: Example accidental route leaks by AS 13 with OTC and ASPA defenses. Both prevent accidental route leaks, but only ASPA can stop an intentional route leak, since an attacker can remove OTC attributes. Arrows pointing down indicate a provider to customer relationship where customers pay providers for traffic. Horizontal lines are peer-to-peer relationships where traffic flows freely.

protection against all kinds of path manipulation, [3] suggests using BGPsec; but, as we shall see in §VIII, BGPsec is ineffective under partial adoption.

B. Other Post-ROV Defenses

We next present several other post-ROV defenses from previous works; we later evaluate their performance compared with ASPA (and ASPAwN) and explore whether they are complementary to ASPA. These defenses include representatives of three common approaches: filter-based defenses, defenses against path manipulation, and defenses against route leaks (see related work in §X).

Path-End Validation [15]. Path-End authenticates the first two ASes (the origin and its immediate neighbors) on an AS path. It differs from ASPA in that instead of publishing only providers, an AS publishes the set of all of its neighbors without indicating the neighbor's relationship. While Path-End includes an extension to verify whether any link to/from an adopting AS on a path is consistent with the Path-End record for that AS, we only consider verification on the first two hops (i.e., the origin and its neighbor) since it is the main mechanism evaluated in [15]. In addition, we do not consider Path-End's prevention of route leaks since it is very limited (only applicable to stub ASes).

EdgeFilter. This technique is described in Section 9 of RFC 7454 [18]. It requires that an AS drops announcements coming from an edge AS if the edge AS has forged the AS-Path. This is possible because an announcement originating from an edge AS should only contain its own ASN. When a provider deploys this policy, all edge ASes connected to that provider will not be able to manipulate the AS path to perform any of the attacks described in this paper.

We believe that EdgeFilter is easier to deploy compared to prefix filtering, where an allowlist of prefixes is maintained, and announcements of prefixes outside of this allowlist are dropped. First, EdgeFilter incurs lower maintenance overhead and is less error-prone than prefix filtering. In EdgeFilter, the filter can be set up when an edge AS connects to a provider, and it is not changed unless due to special events (e.g., when the edge AS expands and gets customers), whereas with prefix filtering, the prefix allowlist can change often and failure to update it accurately can result in loss of traffic. Second, with the low overhead, the financial incentive for a provider to deploy EdgeFilter is higher than that with prefix filtering. We show that EdgeFilter can prevent route leak and path manipulation attacks from edge ASes at full adoption, and is generally comparable to other defenses under partial adoption; see §VIII.

While EdgeFilter is already recommended, it is not universally deployed. We do not have public data as to which ASes use this mechanism. However, from data sources such as [7], [58], we find that route leaks are often by edge ASes, and that even large transit ASes such as Verizon did not perform EdgeFilter as recently as 2019 [90].

Only To Customer (OTC) Attributes (RFC 9234 [5]). The OTC attribute can be added to BGP announcements to indicate that an announcement should only be sent to customers. If an AS receives an announcement containing an OTC attribute from an unexpected AS such as their customer, the AS can know that the announcement was accidentally leaked and drop it. The operation of OTC attributes is shown in Fig. 1(b). In this case, ASes 2 and 3 adopt OTC. AS 2 adds an OTC attribute $\mathbf{OTC_2}$ when sending the announcement to AS 13. AS 3 can detect the route leak from leaker AS 13 because it sees the $\mathbf{OTC_2}$ attribute added by AS 2.

OTC can only foil accidental route leaks, since for malicious route leaks, a leaker can simply remove the OTC attribute, making OTC ineffective. We see OTC attributes have been deployed in the wild in our investigations into the RIPE and RouteView RIB dumps [73], [76] using bgp-kit [99], but only by a few ASes that were not well connected.

BGPsec (RFC 8205 [53]). BGPsec is designed to authenticate the contents of the AS_PATH attribute in a BGP Update message. Preventing route leaks is not a design goal of BGPsec; it only aims to ensure the integrity of the AS path. BGPsec is not deployed in the wild, and we show in our

simulations that it has limited effectiveness in partial adoption scenarios, consistent with results in [15], [28], [30], [55], [63]. We assume that the BGPsec path preference is being placed after the AS-Path length preference (also known as security-third) since this is the preferred choice by most network operators (see survey [29]). As described in [63], BGPsec is vulnerable to downgrade attacks. Specifically, even if an entire network deployed BGPsec, an attacker could simply not attach BGPsec signatures to their announcement, and all ASes would treat it like a normal BGP announcement, only preferring signed announcements (after AS-Path length preference [29]). This severely limits BGPsec's effectiveness.

BGP-iSec [63]. BGP-iSec is a recent design that builds upon BGPsec, but improves its support for incremental deployment and adds defenses against route leakage. It introduces four mechanisms: *transitive signatures*, *protected OTC*, *UP attributes*, and *ProConID*. Transitive signatures extend BGPsec signatures to allow partial path verification. Protected OTC uses signed OTC attributes to prevent intentional route leaks. UP attributes use hash preimages to detect route leaks. ProConID allows adopting ASes to specify the nearest adopting ASes in their provider cone, preventing many shortest-path export-all hijacks, including route leaks. BGP-iSec has a similar computational overhead to that of BGPsec, and can benefit from techniques that address the high computation requirements of BGPsec [9], [66], [89], [101]. Due to the ProConID mechanism, BGP-iSec requires manual curation of the adopting ASes within the provider cone.

IV. ASPA with Neighbors (ASPAwN)

The ASPA I-D identifies a security concern, where a provider can announce to its ASPA-adopting customer a corrupt announcement, including a route leak or a forged-origin hijack, without this being detected (see Section 12 in [3] and Section 9.2 in the latest version (version 19) [4]). This is because ASPA records do not include information on customers or peers, only providers.

The ASPA I-D states that such attacks would be rare, and if they occur, legal ramifications should be used to prevent the attack from reoccurring. While rare, attacks from transit ASes do happen such as in [35], and settling security vulnerability exploitation with legal disputes, especially across borders, may be sub-optimal when compared to alternative secure solutions. Furthermore, non-adopting providers may propagate corrupted announcements that they receive from their customers.

We now present *ASPAwN*[2], an extension to ASPA that we designed to mitigate such vulnerabilities. ASPAwN extends both the ASPA record and the ASPA policy. In the ASPAwN record, the AS can also specify its non-provider neighbors, i.e., customers and peers. Let us now explain the ASPAwN verification policy.

The verification procedure of ASPAwN is applied as follows. Suppose an ASPAwN adopting AS, say AS x, receives an announcement. If the AS-path contains an ASPAwN adopting AS, say AS y, preceded or succeeded by another AS, say AS z, and z does not appear in the ASPA or ASPAwN records of AS y, then the path is invalid. If no such invalid pair of ASes, y, z, is found in the AS-path, then AS x uses the ASPA verification mechanism to further verify the path; if ASPA validation is also successful, then the announcement passed ASPAwN validation. For example, in Fig. 2, when the attacker announces 666-777 to its customers, the announcement will be dropped by customers adopting ASPAwN since AS 666 is *not* listed as a neighbor of AS 777.

The ASPAwN policy is especially helpful to protect the ASes in the attacker's customer cone (see §VIII-B). Additionally, as we shall see, even if only edge ASes announce their customers and peers, then attackers will be forced to announce an AS path of at least length 3, and ASPAwN will be at least as strong as Path-End. This matters because for edge and stub ASes, publishing a list of neighbors will be within their financial interests. Also, since they have no customers (and stub ASes will also have no peers), this will be easy and yet make the attacks more difficult, even if transit ASes do not partake in this optional extension.

V. Post-ROV Attacks

In this section, we present several well-known and one new post-ROV attacks; the new[3] attack is *first-ASN-stripping*. We further describe the actions of the various post-ROV defenses presented in §III against these attacks. All of these attacks are considered post-ROV attacks because ROV does not detect them, and thus ROV adoption has no effect on them.

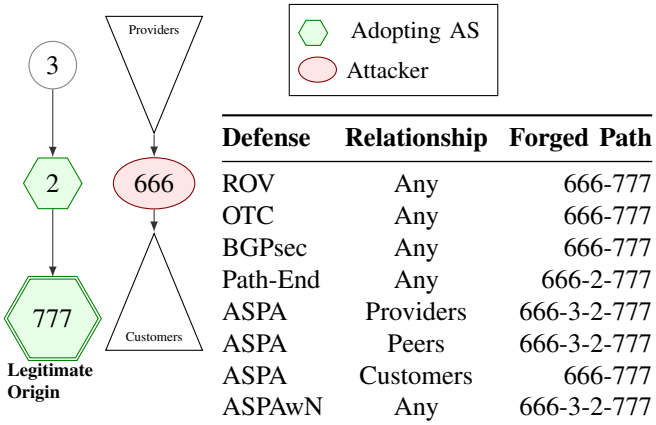
A. Well-known Attacks

Forged-origin Hijack. This attack is also referred to as *1-hop attack* [15], [77]. It is an *aggressive* path manipulation attack where an attacker sets the AS path to be the legitimate origin followed by that attacker's AS, and sends the announcement to all neighbors. Since the first AS in the AS path is the legitimate origin, the announcement is ROV-valid. In §III, we show that for an adopting origin, ASPA can foil this attack when the announcement is sent to a provider, while it fails when the announcement is sent to a customer or bilateral peer. In contrast, ASPAwN foils this attack in both cases. Path-End can foil this attack since an adopting AS declares a set of valid neighbors ("next hops").

Shortest-Path Export-All. In this attack, an attacker exports, to all of its neighbors, the shortest AS path that will not be dropped by ASes adopting the defense mechanism(s) in use, including the defenses being evaluated (with different

[2]After submitting this paper, we learned of a closely related extension of ASPA called *ASRA (AS Relationship Authorization)* [25], [26], [85], with two verification algorithms, ASRA-Alg. A and ASRA-Alg. B. We believe that ASPAwN is simpler than these two algorithms, while has the same or very similar security benefits as ASRA-Alg. B, and significantly better security benefits than ASRA-Alg. A. Our work was developed independently and concurrently with ASRA.

[3]Some people may have been aware of or suspected the first-ASN-stripping attack, e.g., the ASPA I-D recommends defenses which may be designed to prevent it, but we did not find previous publication on this attack.



Defense	Relationship	Forged Path
ROV	Any	666-777
OTC	Any	666-777
BGPsec	Any	666-777
Path-End	Any	666-2-777
ASPA	Providers	666-3-2-777
ASPA	Peers	666-3-2-777
ASPA	Customers	666-777
ASPAwN	Any	666-3-2-777

Fig. 2: Example shortest-path export-all attacks against different defenses. If the receiving AS does not apply `enforce-first-AS`, then the attacker can further remove its own ASN (666) from the path; see §VIII-C.

adoption percentages). This effective attack has been evaluated in several works [33], [63]. When the defense is adopted by only a small percentage of the ASes, shortest-path export-all may be less effective than an aggressive attack that ignores the defense mechanism. Indeed, we found that before ASPA adoption reaches about 20%, forged-origin hijack is more effective. However, as adoption further increases, shortest-path export-all becomes more effective (see §VIII).

The AS path exported by the attacker in the shortest-path export-all may depend on the specific defense mechanism being evaluated. We demonstrate a few of these options in Fig. 2. In this example, AS 777 is the legitimate origin. It has a provider, AS 2, which has a provider, AS 3. For ASPA, since AS 777 and AS 2 adopt ASPA, while AS 3 does not adopt it, AS 666 will have to announce an AS path of 666-3-2-777 to its providers, faking as if it is a provider of AS 3, to avoid detection by any ASPA AS within the attacker's provider cone. When sending to peers and/or customers however, the attacker only needs to send an AS path of 666-777, since ASPA records do not contain information regarding peers or customers, so a forged-origin hijack is all that is needed to circumvent ROV (which we assume all ASes deploy for this work). This intuitively also motivates ASPAwN (see §IV), since the attacker will have to send an AS path of 666-3-2-777 to its peers and customers to evade detection.

Path-End, on the other hand, only validates the next hop from the origin. Therefore, the attacker just needs to use an AS path of length three, containing the legitimate origin (victim), the legitimate origin's provider, and its own ASN, to evade detection. In Fig. 2, the attacker merely needs to announce 666-2-777 to avoid detection by Path-End ASes.

For BGPsec, shortest-path export-all is the same as forged-origin hijack, since BGPsec allows announcements without signatures, e.g., if the announcement is passed via a non-adopting AS.

Route Leak. While a route leak can be announced maliciously, often route leaks are accidental misconfigurations [5]. As mentioned earlier, ASPA can foil route leaks (both accidental and malicious), while OTC can only foil accidental route leaks. Therefore, we compared (in §VIII) the effectiveness of ASPA and of OTC in preventing (only) *accidental* route leaks. We found that ASPA and OTC achieve identical performance for accidental route leaks for non-adopting leaking ASes; we prove this property in Appendix B.

BGPsec is not designed to prevent route leaks, while BGPiSec has multiple mechanisms to defend against route leaks. Path-End only has a very limited mechanism against route leak, which we do not consider in this work. EdgeFilter is effective in defeating route leaks for a leaker that is at the edge and the providers of the leaker adopt EdgeFilter.

B. First-ASN-Stripping Attack

When AS x sends an announcement to AS y, it *should* prepend its own ASN to the AS-path in the announcement. Correspondingly, when AS y receives an announcement from AS x, it *should* verify that x is the first ASN in the AS-Path received from x; this validation is referred to as `enforce-first-AS`. However, not all BGP routers perform `enforce-first-AS` validation; in fact, it is often not even the default (at least not by default; see Table I).

In the *first-ASN-stripping* attack, an attacker exports an announcement with an AS path that contains the legitimate origin ASN, *without prepending its own ASN*, when exporting to an AS that does not perform `enforce-first-AS` validation. The AS path may contain only the origin ASN, like in the regular prefix hijack, or it may contain the shortest AS path which suffices to prevent dropping by ASes adopting the evaluated defense, like in the shortest-path export-all attack. To the best of our knowledge, the first-ASN-stripping attack is novel, i.e., it has not been presented in prior publications.

The first-ASN-stripping hijack attack results in a shorter AS-Path, and hence a higher chance of attracting traffic. For instance, in Fig. 2, where for regular shortest-path export-all hijack, the attacker (AS 666) sets the AS path to 666-3-2-777 to its provider to bypass ASPA, the attacker will instead announce a shorter AS path, 3-2-777, with first-ASN-stripping hijack. Similarly, for a regular forged-origin hijack, AS 666 sets the AS path to 666-777, but with first-ASN-stripping hijack, it will simply use 777 as its AS path.

We believe that first-ASN-stripping hijack is often feasible, since the option to check for the neighbor AS is typically configurable in routers, as the `enforce-first-AS` option [2], [20], [46], [94]. ASes may turn this option off when they remove their own ASN from the AS-Path to shorten the path and attract more traffic [39]. They also do this by default when peering to transparent route servers. In fact, transparent route servers do not prepend their own ASN to the AS path (defined in [45]), so *this attack can be done by transparent route servers*, and may be feasible also to their customers. This is because when an AS is peered to a transparent route server,

TABLE I: Default behaviors and per-neighbor enforcement features of several popular router vendors for `enforce-first-AS`. Cisco does not support per-neighbor enforcement of the first ASN [12], meaning if the setting is disabled to connect to a route server, it is disabled for all other BGP neighbors.

Vendor	Enforce 1st AS by Default	Per-Neighbor Enforcement
Cisco	Y	N
Juniper	N	Y
Arista	Y	Y
BIRD	N	Y

they must turn off `enforce-first-AS` for the peering (see Section 2.2.2.1 in RFC 7947 [45]).

We have found that the `enforce-first-AS` is not enabled by default on all routers. Specifically, we have seen that this option is disabled by default on BIRD routers [19] and Juniper routers [39], while it is enabled by default on both Cisco and Arista routers. On Cisco routers, although it is enabled by default, it is a global setting, meaning that a router that is connected to a transparent route server must have this setting turned off for all of its other neighbors. The global scope of the `enforce-first-AS` setting has been an open issue for Cisco for over 10 years, last updated in 2023 [12]. It is also important to note that a motivated attacker could simply seek out a provider that would be vulnerable to this attack. While there is no readily available data for us to quantify how many ASes do not enforce this option, we know at a minimum that all Cisco routers connected to a transparent route server have this setting turned off [12], so all ASes connected to those routers could launch this attack. We present an upper bound of the impact of this attack in §VIII-C and summarize the vulnerabilities in Table I. We have notified all major vendors that do not have `enforce-first-AS` as the default option, as well as Cisco about how their open vulnerability can be exploited.

The vulnerability of not enforcing the first-AS is, to some extent, attributable to the requirement language in Section 6.3 of RFC 4271 [71]. It states "the local system MAY check whether the leftmost (with respect to the position of octets in the protocol message) AS in the AS_PATH attribute is equal to the autonomous system number of the peer that sent the message." This language allows ASes to not deploy the `enforce-first-AS` mechanism, and thus become vulnerable to first-ASN-stripping hijack. In the ASPA I-D, it is pointed out that the attacker may remove themselves from the AS-Path to attract more traffic, and the requirement is amended to say *MUST* instead of *MAY*. However, ASPA does not require that the `enforce-first-AS` option be used with route servers (Section 6 in ASPA I-D) [3].

VI. Simulation Setup

Our simulations extend BGPy [23], a thoroughly tested, open-source BGP simulator, used in previous works [62], [63]. We have extended BGPy to include all the attacks and defenses described in this paper, and open source our extensions at [21].

Our evaluation uses CAIDA's Internet-scale AS topology (April 2024) with relationships marked as peer-to-peer or customer-to-provider [10]. We assume ROV is deployed at all ASes. For each defense policy, we assess partial adoption scenarios, varying adoption from 1% to 99%, while the rest of the network runs BGP/ROV. Following prior works [62], [63], we categorize the ASes into tier-1, edge, and other transit ASes. For a certain adoption percentage of a defense policy, unless otherwise stated, we assume uniform random adoption across all ASes. Each setting is tested with 1000 trials, and results are presented with 95% confidence intervals.

In each simulation run, the victim (i.e., the legitimate origin of a prefix) is selected from the edge ASes (stub or multihomed ASes). We also assume that the victim/legitimate origin always announces a ROA for their prefix. The attacker (or attackers in the multi-attacker case) is an edge AS or transit AS. Attacker-victim pairs remain consistent across the percent adoptions, and attacker-victim pairs are selected randomly for each trial.

Since we attempt to evaluate the effects of the security policy, we assume the victim AS (i.e., the legitimate origin) always adopts the security policy. This is because if the victim did not adopt the security policy, then the security policies such as ASPA and Path-End would have no effect and we would be evaluating BGP rather than the desired security policy. Additionally, for simplicity, ASes that adopt ASPA in our simulations both filter using ASPA and publish ASPA records; similarly for ASPAwN.

For this work, we focus on *attacker success rate*, i.e., the percentage of ASes in the overall topology that route back to the attacker on the data plane. We investigate the data plane (the actual paths for the data packets) rather than the control plane (the existence of a prefix in the routing table) since control plane metrics can significantly underestimate the effectiveness of certain attacks [62].

While we also measured disconnections and successful victim connections, we observed that the disconnection rate is close to zero, and hence the victim's successful connection rate is the complement of the attacker's success rate, and thus, we did not include these results.

VII. The Need for Post-ROV Defenses

Recent measurements [14], [41], [52], [54], [72], [75], [78] show that ROV enforcement has increased significantly in the past several years, particularly in tier-1 and upper-level ISPs. We have aggregated these data sources in a tool that we open source to the community [22]. In this section, we use various data sources to evaluate the impact of ROV enforcement on prefix and subprefix hijacks, as well as post-ROV attacks. For the first time, we now find that *with current ROV adoption*, the post-ROV attacks that we evaluated are already more effective than prefix hijacks, i.e., post-ROV defenses such as ASPA are already needed.

A. Methodology

For the results of this section, i.e., the need for post-ROV defenses, we evaluate the defenses provided by the current

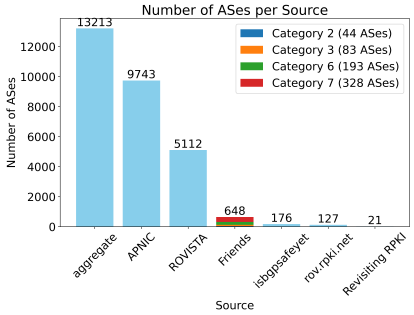


Fig. 3: ROV deployment based on measurements from existing studies [14], [41], [52], [54], [72], [75], [78]. Each bar represents the results from one existing study, where the number on top of the bar represents the number of ASes likely adopting ROV estimated from that study; the leftmost bar represents the aggregate from all the studies (it represents the number of unique ASes, and hence is lower than the total sum from the individual studies). For the study of [41] (marked as 'Friends'), we use stacked bars to show the numbers of ASes in the multiple categories defined there (corresponding to different levels of evidence of ROV enforcement).

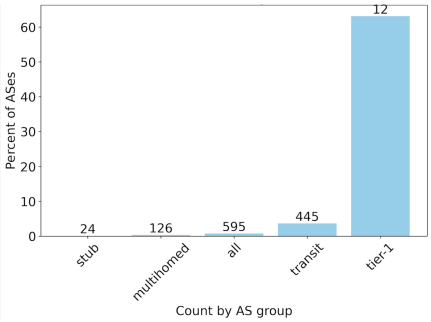


Fig. 4: ROV deployment status for the ASes in categories 3, 6, and 7 from [41]. Only the ASes in the CAIDA topology [10] are considered, and hence the total number of ASes is 595 (marked with 'all'), lower than the corresponding value in Fig. 3. For each type of ASes, the bar represents the percentage of ROV adopting ASes of all the ASes in that type in the CAIDA topology, where 'all' represents the aggregate of all AS types, including stub, multihomed, and transit ASes (tier-1 is a subset of transit). The number on top of each bar represents the number of ASes corresponding to the bar.

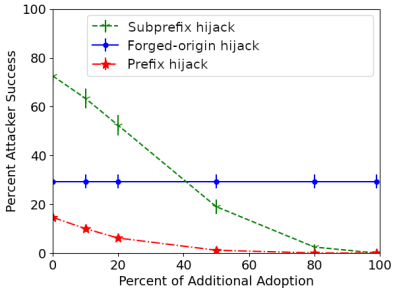


Fig. 5: Attacker success rate with current and future ROV deployment. The x-axis marks the percent of additional adoption relative to the current ROV deployment estimated from [41].

ROV enforcement against three types of attacks: *(i) prefix hijack*, where an attacker announces the same prefix as the legitimate origin of a valid announcement, *(ii) subprefix hijack*, where an attacker announces a subprefix of the prefix used by the legitimate origin of a valid announcement, and *(iii) forged-origin hijack*, where an attacker sets the AS-path to be the legitimate origin followed by itself to avoid being detected by ROV (also discussed in §V).

Subprefix hijacks are significantly more effective than prefix hijacks since there is no competing subprefix. However, for prefixes of length /24, an attacker can only use prefix hijack, since most ASes do not allow prefixes longer than /24 and so a subprefix attack would be dropped in such cases [91].

ROV foils both prefix and subprefix hijacks by identifying the attacker's announcement as invalid. On the other hand, a forged-origin hijack avoids ROV detection by appending the legitimate origin to the AS-Path. While this attack does have a longer AS-Path than a prefix hijack, it is unaffected by the percentage of ROV adoption (since it evades ROV detection) and thus becomes the most effective of the three attacks when ROV adoption exceeds a threshold.

B. Impact of ROV Adoption

We detail the number of ASes deploying ROV from existing studies in Fig. 3. While existing measurements all show significant ROV enforcement, they use different methodologies (see [41], [78]), and their results are not always comparable. For example, RoVista [54] assigns a probability to each AS, whereas [41] assigns each AS to a category.

To be conservative, we only look at the datasets individually (rather than merging them). In the following, we present the results using the data from [41], one of the latest studies on ROV enforcement. While the study in [41] classifies ASes into several categories, we choose only the categories with strong evidence of ROV enforcement, i.e., category 3 ("strong depreference of ROV invalid"), category 6 ("direct positive evidence of ROV"), and category 7 ("strong positive evidence of ROV"). Appendix A discusses the results from other datasets, which show similar trends.

Fig. 4 plots ROV deployment by AS type as defined in [70] using the above three categories of data from [41]. We see that 60% (i.e., 12 out of 19) tier-1 ASes adopt ROV, which can greatly reduce the impact of prefix and subprefix hijacks; other studies in [14], [54] show even more enforcing tier-1 ASes.

In Fig. 5, we present attacker success rate with the current and future ROV deployment under the three types of attacks, i.e., subprefix, prefix, and forged-origin hijacks. Specifically, 0% on the x-axis represents the current state of ROV deployment according to [41]; additional percentages of future ROV deployment (10%, 20%, 50%, 80%, and 99% more future ROV adopting ASes) are also plotted. We observe that, while subprefix hijack is the most effective attack under the current ROV deployment, at around 50% of additional ROV adoption, even subprefix hijack will be less effective than forged-origin hijack.

For /24 prefixes, an attacker cannot perform a subprefix hijack since ASes will drop prefixes that are more specific than /24. In this case, the attacker can use a prefix hijack or a forged-origin hijack, and as can be seen in Fig. 5, a forged-origin hijack is already more effective than prefix hijack.

Therefore, even under such conservative estimates, attackers will start using forged-origin hijacks to avoid ROV detection when attacking /24 prefixes. According to the NIST RPKI validator [65], there are currently 13 million prefix-origin pairs containing a /24 prefix. For subprefix hijacks to shorter prefixes (shorter than /24), deaggregating such prefixes to /24 prefixes is a recommended defense [77], which can also be hijacked by a forged-origin hijack.

This issue of forged-origin hijack is critical, because the percentage of ROV adoption does not affect the attacker success rate of a forged-origin hijack. Indeed, as shown in Fig. 5, this hijack attracts traffic from approximately 30% of ASes regardless of the ROV adoption. We must look to post-ROV defenses like ASPA to protect against these post-ROV attacks, and these defenses are needed *today* rather than in some far-off distant future.

VIII. Security Evaluation

In this section, we evaluate the effectiveness of the various post-ROV defenses (§III) against the post-ROV attacks (§V), namely path manipulations and route leaks.

A. Forged-origin Hijack

Contrary to prior works [74], [95], we find that, ASPA performs very well against forged-origin hijack under partial adoption, even when none of the tier-1 ASes adopts ASPA. In the following, we first present the results when the adoption of defense mechanisms is uniform random among the ASes (Fig. 6(a)), and then the results for ASPA under various adoption scenarios (Fig. 6(b)).

In Fig. 6(a), we plot the attacker success rate from all ASes with various defenses against an edge AS attacker. The results for BGP-iSec, BGPsec, EdgeFilter, ASPA, and Path-End while increasing the percentage of adoption are shown in the figure. The horizontal line marked with 'ROV' shows the results when ASes adopt ROV and no additional defense is used. This is a horizontal line since post-ROV attacks are not detected by ROV, and thus it serves as a baseline for the scenario when ASes are deploying no defense against post-ROV attacks.

We see that BGPsec provides the least impact among all post-ROV defenses, even under a high adoption rate, consistent with results in earlier studies [15], [28], [30], [55], [63] and described in §III.

ASPA performs slightly worse than Path-End and BGP-iSec. This is because ASPA ASes that receive the forged-origin hijack from a provider will not consider the hijack invalid (see one example in Fig. 12). BGP-iSec and Path-End perform identically and perform slightly better than ASPA. Typically BGP-iSec checks every hop along the AS-path, whereas Path-End checks only the origin and the next hop. However, since forged-origin hijacks only have one forged hop in the AS-Path, both BGP-iSec and Path-End perform identically. EdgeFilter can only drop the hijack announcement when all direct providers of the attacker adopt EdgeFilter (and drop the hijack announcement) and hence has worse performance than ASPA, Path-End, and BGP-iSec.

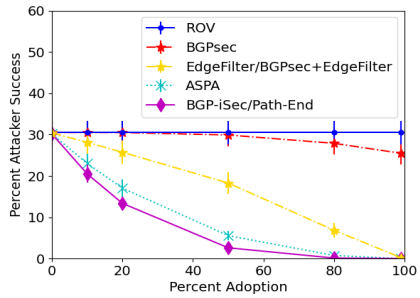
We also evaluated the impact of using EdgeFilter together with other policies. Adding EdgeFilter to ASPA does not lead to additional benefits, since EdgeFilter is only effective if all providers of an edge attacker adopt EdgeFilter, and for a forged-origin hijack, the effect will be the same if these providers adopt ASPA (they will also drop the announcement based on ASPA records). Fig. 6(a) shows that EdgeFilter and EdgeFilter+BGPsec are visually indistinguishable, due to the low impact of BGPsec under partial deployment. Similar trends can be seen for transit AS attackers in Appendix D.

Impact of ASPA in various adoption scenarios. We next evaluate ASPA in several scenarios to compare against prior works. Fig. 6(b) plots the results of three ASPA adoption scenarios: only edge ASes adopt ASPA, no tier-1 AS adopts ASPA, and tier-1 ASes adopt first (i.e., the rest of the adopters are chosen randomly from the non-tier-1 ASes). The results for uniform random adoption that we presented earlier are also presented in the figure for comparison. In contrast to prior works [74], [95], we see that ASPA is effective against forged-origin hijack, even when none of the tier-1 ASes adopts ASPA. As expected, tier-1 ASes adopting first leads to the lowest attacker success of all the scenarios. On the other hand, ASPA still is highly effective under random adoption and no tier-1 adoption. Therefore, our results motivate the adoption of ASPA at intermediate and edge ASes. The results when only edge ASes adopt ASPA are only slightly better than that of ROV, which is expected, since these adopting edge ASes receive hijack announcements from their providers most of the time, which ASPA does not protect against.

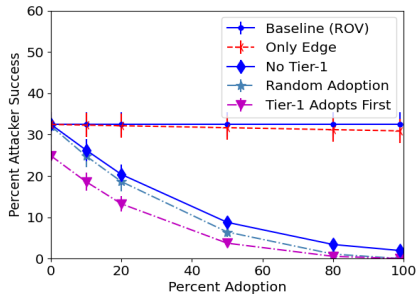
B. Shortest-Path Export-All Hijack

The shortest-path export-all attack was not evaluated against ASPA in [74], while [95] states that it was no more effective than a forged-origin hijack. Our results disagree with this and show that shortest-path export-all attack becomes significantly more effective than forged-origin hijack when adoption increases; it has some advantage already at 20% adoption, and at 50% the advantage is quite dramatic (comparing Fig. 6(a) and Fig. 7(a)). This is because when many ASes deploy ASPA, attacks that bypass ASPA (such as the shortest-path export-all hijack) outperform attacks that ASPA prevents (such as the forged-origin hijack).

For this attack, we consider edge and transit AS attackers. Since this is a more powerful attack than a forged-origin hijack when the adoption rate is high, we present results for both single and multiple (ten) edge attackers. For transit attackers, where ASPA has known vulnerability (see §IV), we further present the results of ASPAwN, our proposed extension to ASPA, to address this vulnerability. As mentioned in §V, for BGPsec, the forged AS-Path in the shortest-path export-all hijack is the same as the forged AS-Path in the forged-origin

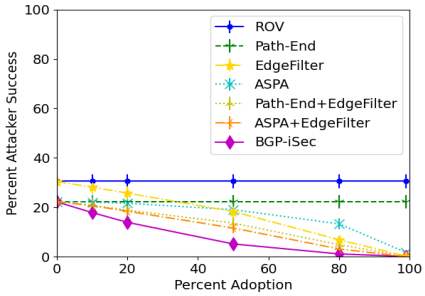


(a) Edge AS attacker, forged-origin hijack.

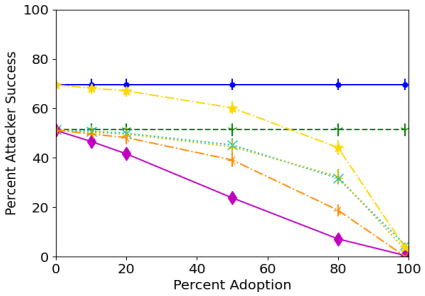


(b) Results under various ASPA adoption scenarios.

Fig. 6: Impact of different defenses against forged-origin hijack from an edge AS attacker. In (a), the results for BGP-iSec and Path-End are visually indistinguishable, and also visually indistinguishable from their results when combined with EdgeFilter. Similarly, the results of EdgeFilter (alone) are visually indistinguishable from EdgeFilter+BGPsec. In (b), in contrast to prior works [74], [95], we show ASPA is effective even when no tier-1 ASes adopt ASPA.



(a) Single edge AS attacker, shortest-path export-all.



(b) 10 edge AS attackers, shortest-path export-all.

Fig. 7: Results of various defenses against shortest-path export-all attacks from edge AS attackers. The legend for (b) is the same as that for (a) and is omitted.

hijack. We hence omit BGPsec from the following graphs to avoid cluttering them.

Single edge attacker. Fig. 7(a) presents the results of various defenses against shortest-path export-all with a single attacker at the edge of the network. For Path-End, the attacker will announce an AS path of length three to evade detection. As a result, the hijack rate is a flat line, independent of the adoption percentage. At a low adoption rate, this attack leads to longer AS paths, and hence a lower hijack rate than forged-origin hijack (which has an AS path of length two). For example, when only a single AS adopts Path-End, the hijack rate under shortest-path export-all is 11% lower compared to forged-origin hijack (see the leftmost point in Fig. 7(a) and that in Fig. 6(a).). As the adoption rate increases, the hijack rate under shortest-path export-all becomes higher than that of forged-origin hijack, i.e., an attacker will be more motivated to use shortest-path export-all rather than a forged-origin hijack.

We see from Fig. 7(a) that ASPA performs similarly to Path-End in early (<50%) adoption. This is because, in this case, for both defenses, an edge AS attacker will announce a path of length three to its providers in order to avoid detection. For a higher adoption rate, ASPA outperforms Path-End, since, unlike Path-End, increased ASPA adoption will eventually force an attacker to use a longer path. Specifically, if the legitimate origin and all of its providers have adopted ASPA, then the attacker must search the origin's providers' providers for a non-adopting AS, making the AS path at least length four. In this way, as ASPA adoption increases, the length of the AS path in the shortest-path export-all attack also increases and attacker success decreases, eventually reaching 0% at full adoption of ASPA when no more plausible paths exist that the attacker can use without detection. BGP-iSec outperforms ASPA+EdgeFilter slightly here due to its stronger security guarantees (see §III).

In Fig. 7(a), as expected, EdgeFilter leads to a 0% attacker success rate at full adoption against an attacking edge AS. At 50% adoption, it performs in line with Path-End and ASPA, and afterwards slightly outperforms both ASPA and Path-End. This is because as the adoption increases, it is more likely that all providers of the attacker adopt EdgeFilter, and hence the hijack announcement will more likely be dropped. Note that EdgeFilter can be deployed *easily today* by most ASes. On the other hand, attackers may find some AS that does not adopt EdgeFilter and become its customer, circumventing EdgeFilter.

Last, when EdgeFilter is combined with Path-End or ASPA, both of these combined policies perform better than all other defenses, with EdgeFilter+ASPA performing slightly better

than EdgeFilter+Path-End above 50% adoption.

Multiple edge attackers. We next consider multiple edge attackers, e.g., larger-scale attacks that can be performed by nation-states. Fig. 7(b) plots the results of various defenses when using ten (10) attacking edge ASes that are randomly selected from the network. Compared to the results with a single edge attacker in Fig. 7(a), it is clear that 10 edge attackers lead to a significantly higher hijack rate. For example, with a single adopter (i.e., the leftmost point in Fig. 7(b)), the hijack rate is increased by around 26% with 10 attackers compared to having a single attacker for all policies that force an initial shortest path length of three (e.g., ASPA, Path-End); for ROV and EdgeFilter, the increase is 30%.

The effect of EdgeFilter defense against ten (10) attacking ASes is significantly reduced compared to its effect against a single edge attacker. This is expected, since the probability of having all the providers of the 10 attackers deploy EdgeFilter is significantly lower than that with a single attacker. For the other policies, the trend with 10 attackers is similar to that with a single attacker. On the other hand, the gap between ASPA and Path-End when there are 10 attackers is larger than that with a single attacker, indicating larger benefits of ASPA over Path-End with multiple edge attackers. The same is true when comparing BGP-iSec to ASPA, and the same is true for ASPA+EdgeFilter versus Path-End+EdgeFilter.

In the shortest-path export-all attack, the attacker must announce a long path to compete with the existing prefix from the legitimate origin in order to evade detection by the defense policy. Due to the length of this AS-Path, the attacker typically only wins the traffic from their provider cone, since announcements from customers are preferred over announcements from peers and providers. Multiple attackers will have significantly more ASes contained within their respective provider cones and thus will win significantly more traffic than a single attacker.

Transit attacker. While intuitively it is not common for transit ASes to launch hijacks (since it would erode trust and hurt their business model), such hijacks can happen [35]. We next consider shortest-path export-all by a transit AS, which is randomly selected among all transit ASes, excluding tier-1 ASes, since it is very unlikely that a tier-1 AS would perform such hijacks.

Fig. 8(a) and (b) present the results. EdgeFilter has no effect in this case, and hence the results of EdgeFilter and ROV coincide with each other. We again see that the hijack rate in Fig. 8(a) is higher than that with a single edge AS attacker (see Fig. 7(a)). In addition, ASPA has a similar performance as Path-End at a low adoption rate and then outperforms Path-End despite its vulnerability in protecting customers of the transit attacking AS. ASPAwN has a minimal effect on the overall internet since it is specifically focused on the attacker's customer cone, and thus this line is visually indistinguishable from ASPA. BGP-iSec outperforms ASPA by a significant margin due to its stronger security guarantees (discussed in §III).

Fig. 8(b) presents the results for the customer cone of the attacking transit AS. That is, the hijack rate is obtained only considering the ASes in the customer cone of the attacker. In this case, not surprisingly, ASPA has a higher hijack rate than Path-End. This is because ASPA does not protect against these types of path manipulations against customers (see §IV).

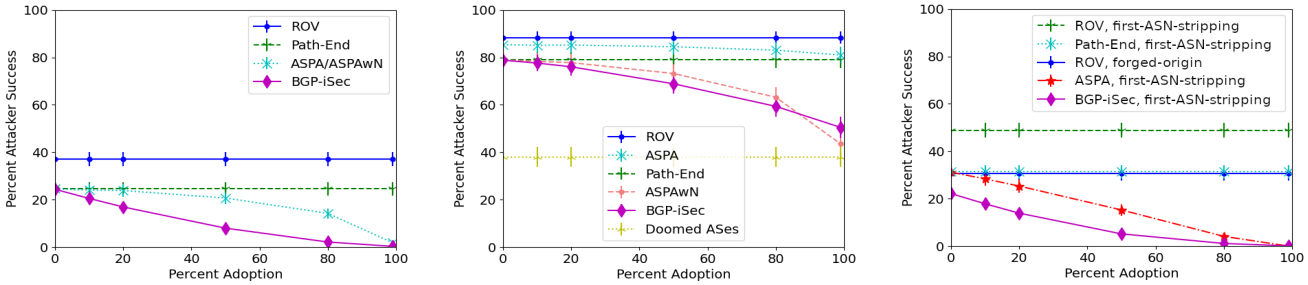
On the other hand, ASPAwN results in a significantly lower hijack rate than both Path-End and ASPA. Note that even when the adoption rate is 100%, the hijack rate under ASPAwN and BGP-iSec remains above 40%. This is unavoidable since it is due to 'doomed ASes', i.e., those customers of the transit AS that have no other ways of routing the traffic beyond the transit AS. BGP-iSec performs similarly to ASPAwN since it checks transitive signatures at every hop along the AS-Path. The above shows that ASPAwN is effective in resolving ASPA's vulnerability to attacks from transit ASes against their customer cones.

C. First-ASN-Stripping Attack

We now present the results under first-ASN-stripping attack from an edge attacker. While we believe that this attack can be launched by a large number of ASes and all transparent route servers (see §V-B and Table I), we do not have actual data on which ASes are vulnerable to first-ASN-stripping. For simplicity, we assume that *all* the ASes are vulnerable, which is clearly an overestimate, and hence our results present an upper bound on the impact of this attack. While we select the attacker randomly from edge ASes, it is important to note that a motivated attacker could simply seek out and become a customer of one of the ASes or IXPs that are vulnerable, i.e., those that do not apply `enforce-first-AS`.

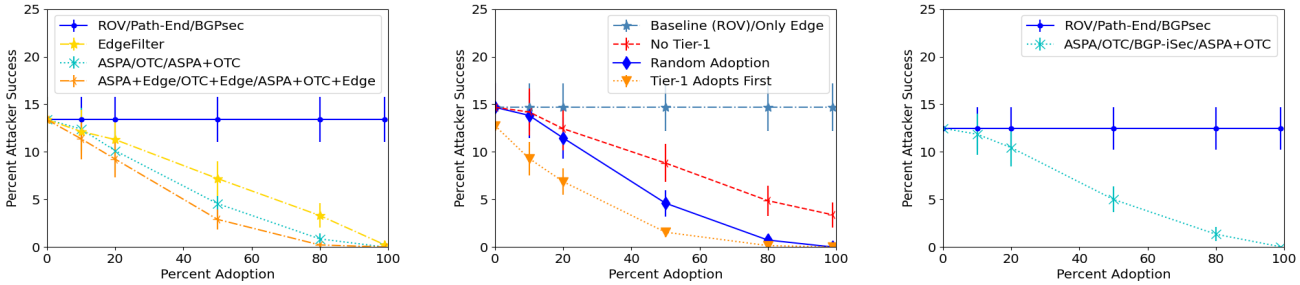
In the first-ASN-stripping attack, when the defense is only ROV (i.e., no other post-ROV defenses), to attack a legitimate origin, x, the attacker a will simply set the AS path to be x, instead of a-x (i.e., instead of using a forged-origin hijack); this suffices to evade ROV. Fig. 8(c) compares the results of these two cases (i.e., ROV with first-ASN-stripping and forged-origin hijack). We see that first-ASN-stripping hijack leads to a 20% higher hijack rate compared to forged-origin hijack when assuming no `enforce-first-AS`, i.e., first-ASN-stripping attack can be performed. The above results indicate that the first-ASN-stripping attack can have a significant impact on routing security.

We further evaluate the impact of the first-ASN-stripping attack with ASPA, BGP-iSec, and Path-End defenses. In all cases, we assume that the attacker does shortest-path export-all hijack, but removes itself from the AS path. For Path-End, this leads to a path length of two, the same path length as that under forged-origin hijack with ROV defense. We see in Fig. 8(c) that the hijack rate under Path-End is visually indistinguishable from that under forged-origin hijack with ROV. For ASPA and BGP-iSec, the hijack rate decreases with the adoption rate and eventually reaches zero at full adoption.



(a) From a transit AS (not tier-1) attacker. (b) Results for the attacker's customer cone. (c) Results under first-ASN-stripping hijack.

Fig. 8: Results under shortest-path export-all attacks from a single transit AS are shown in (a) and (b); In (a), the results of ASPAwN and ASPA overlap. Results under the first-ASN-stripping hijack from an edge AS are shown in (c), where we also show the results of ROV against forged-origin hijack for comparison.



(a) Multihomed AS leaker. (b) ASPA, multihomed AS leaker. (c) Transit AS (not tier-1) leaker.

Fig. 9: Results against accidental route leak from a single AS attacker. Both (a) and (b) consider a single multihomed leaker, while (c) considers a transit AS leaker.

D. Accidental Route Leaks

Preventing route leaks is stated as a major goal in the ASPA specifications. We next compare ASPA and OTC under *accidental* route leaks, since OTC can only prevent accidental route leaks, and most route leaks are believed to be unintentional [87]. As such, a leaker does not manipulate the AS path and path attributes when leaking a route (so the OTC attribute is not dropped). In addition, we assume the leaker does not adopt OTC or ASPA, and leaks a route due to misconfigurations. We only consider route leaks in the form of violating valley-free routing, i.e., leaking a route from a non-customer (provider or peer) to another non-customer.

In the following, we consider two scenarios with a single leaker: when the leaker is a multihomed AS, and when it is a transit AS. We do not simulate route leaks from a stub AS (i.e., has a single provider) since its provider should drop any route leaks due to the loop prevention mechanism in BGP. In our simulations, when selecting a leaker, the legitimate origin is chosen as an edge AS that is not within the leaker's customer cone. In that way, the leaker always receives the announcement that it is leaking from a peer or a provider and leaks the best announcement received in its local RIB to providers and peers.

Leak from a multihomed AS. Fig. 9(a) plots the results from a multihomed AS. Again, contrary to prior works [74], [95] we find that ASPA performs well under random adoption. Since Path-End and BGPsec do not defend against route leaks, their results coincide with that of ROV. Since BGP-iSec uses a signed OTC attribute, its results are equivalent to OTC. We also observe that the attack success rates under ASPA and OTC are identical. In fact, we prove that they provide equivalent protection in the settings we simulate in Appendix B. EdgeFilter performs slightly worse than ASPA and OTC. Additionally, OTC+EdgeFilter, ASPA+EdgeFilter, and ASPA+OTC+EdgeFilter are visually indistinguishable, all better than using ASPA or OTC alone. The benefit of using EdgeFilter with ASPA or OTC is that if the provider of the leaker that receives the leaked route adopts EdgeFilter, it will discard the leaked announcement, while this will not happen if this provider adopts ASPA or OTC (recall we assume that the leaker does not adopt ASPA or OTC).

ASPA adoption scenarios with a multihomed AS leaker. We now evaluate ASPA under different adoption scenarios. Fig. 9(b) plots the results with a multihomed AS leaker under four adoption scenarios: only edge ASes adopting, no tier-1 AS adopting, tier-1 AS adopting first, and uniform random adoption. Contrary to prior works [74], [95], we find that ASPA performs well even when none of the tier-1 ASes adopts ASPA. Of course, when tier-1 ASes adopt first, ASPA performs better than other scenarios, as expected. However, random adoption of ASPA also performs only about 5%

worse and reduces the hijacks considerably when compared to the baseline scenario (i.e., only using ROV). Even when no tier-1 ASes adopt ASPA, ASPA still performs well. While edge AS adoption is also impactful, as quantified by the random adopting ASes, they must be coupled with transit ASes adopting, or else they will be unable to detect the hijack, and thus the results are the same as the baseline results (when only using ROV).

Leak from a transit AS. For the accidental route leak from a transit AS, we compare the performance of ASPA, OTC, BGP-iSec, and ASPA+OTC. We do not consider EdgeFilter since the route leak is by a transit AS. Additionally, since it is not possible to leak an announcement to customers, the customer cones of transit ASes are not considered in the resulting metrics. The results are presented in Fig. 9(c). We see that once again, ASPA, OTC, BGP-iSec, and their combinations all provide identical results.

IX. Discussion

A. Defending against Route Leaks: ASPA vs. OTC

We have observed from our simulations that ASPA and OTC are equally effective at preventing accidental route leaks. In Appendix B, we present an analysis that shows that if adopting ASes would not perform (accidental) route leaks, then ASPA and OTC are indeed equally effective. Even without this assumption, our simulations show that ASPA performs only negligibly better than OTC (their results are visually indistinguishable). Hence, against accidental route leaks, adopting OTC already provides benefits equal to what will be provided by adopting ASPA.

However, ASPA has further advantages over OTC. Most significantly, ASPA protects against intentional route leaks, where OTC attributes will be stripped by the attacker, and have no impact. ASPA also prevents many path manipulation attacks as shown in this paper (see §IX-C). Additionally, the study in [40] shows that a small fraction of ASes may remove transitive attributes such as OTC attributes, foiling OTC. Furthermore, adding OTC attributes slightly increase the size of the announcements, whereas announcing an ASPA record does not change the size of announcements.

B. Adoption of EdgeFilter

As can be seen in Fig. 6, Fig. 7, and Fig. 9, the simple EdgeFilter mechanism is very effective at preventing numerous path manipulation attacks and misconfigurations. EdgeFilter is a recommended security practice [18] which is already deployed in multiple ASes, although it is not yet universally deployed (§III). It would be best if EdgeFilter is universally deployed, and in particular, deployed in conjunction with other security mechanism(s). Note that attackers may selectively use providers that do not perform EdgeFilter; further research is required to study the viability and impact of *chosen-location attackers*, i.e., attacks where the attacker can choose among multiple possible locations for its AS in the Internet. This is also very relevant for the first-ASN-stripping attack.

C. ASPA vs Alternatives against Path Manipulation Attacks

We compared, using simulations, ASPA, ASPAwN, Path-End, BGP-iSec, and BGPsec as defenses against (intentional) post-ROV attacks. BGPsec was standardized since 2017 [53] and yet is still not deployed. As can be seen in Fig. 6, when assuming BGPsec is deployed security third [29] (discussed in §III), BGPsec barely improves on the performance of BGP/ROV, and both ASPA and Path-End significantly outperform it.

While BGP-iSec significantly improves upon the security of BGPsec and is the most secure policy we tested against intentional hijacks, it has the same deployment challenges as BGPSec along with additional operational overhead from the ProConID mechanism. There have been works on optimizing BGPsec and reducing overhead from signature verification, and further research should apply these techniques and additional techniques to reduce the computational overhead of BGP-iSec.

On the other hand, BGP-iSec may have additional benefits in real-world scenarios, since, like previous works, our simulations assume that if an announcement is exported to a provider, then it is exported to *all* providers. Further research should use measurements to perform more realistic simulations using realistic AS export policies, possibly by measuring and identifying the export policies of the ASes on the Internet. In such simulations, we expect BGP-iSec to provide additional security benefits over ASPA, since ASPA only checks for plausible paths (based on AS relationships), whereas BGP-iSec requires attackers to abuse existing, real-world paths.

We presented ASPAwN, an extension of ASPA. ASPAwN provides advantages over ASPA, especially in the case of the attacker's customer cone, where it performs significantly better. We believe that ASPAwN has similar security properties as ASRA Algorithm B [85], which was developed independently and in parallel. Further research is required to confirm this and evaluate the different options among the two ASRA algorithms, ASPAwN, and other variants.

We also presented the first-ASN-stripping hijack, a powerful post-ROV attack. We recommend that routing software enables the `enforce-first-AS` option by default to prevent abuse by ASes, and this option should be set per neighbor, rather than globally. We note that this will not prevent transparent route servers from launching this attack, and similar to EdgeFilter, a motivated attacker may simply avoid providers that enable this `enforce-first-AS` option. Note that this further motivates ASPA since, as shown in Fig. 8(c), ASPA effectively defends against this attack.

X. Related Work

A. Contrasting Results to Prior Works

Two recent studies [74], [95] evaluated the effectiveness of ASPA using simulations. Compared to them, our work provides a more comprehensive evaluation, covering a wider range of attack scenarios (path manipulation, accidental route leaks, edge and transit attackers, as well as single vs multiple attackers). Most importantly, we believe that our findings may

help to avoid possible misleading conclusions that could be drawn from [74], [95]. Let us explain.

Both [74], [95] come to the conclusion that ASPA is ineffective when not deployed at tier-1 ASes; [95] also states that deploying ASPA in intermediate ASes has no benefit. The results of our extensive simulations, e.g., Fig. 6(b) and Fig. 9(b), contradict these conclusions, and support the statement in the ASPA I-D [3] that ASPA "offers significant benefits to early adopters".

Many of the differences between our results and those of [74], [95] seem to be due to differences in methodology or simulation approaches. For example, the study in [95] assumed that only the target/victim AS (and no other ASes) issues ASPAs, limiting ASPA's effect at intermediate ASes. They also only evaluated about 500 ASes in Japan and included a single tier-1 AS, instead of working with the full AS topology. We believe these limitations account for the differences in their conclusions and lead to overlooking the benefits of deploying ASPA in intermediate ASes.

The study in [95] also states that the shortest-path export-all attack (which they call ASPA-aware attacks) is equivalent to a forged-origin hijack. This difference likely also comes from the decision to have only the target/victim AS (and no other ASes) issue ASPAs. If only a single AS issues ASPAs, the shortest-path export-all hijack has almost the same AS-Path length as a forged-origin hijack, making their effectiveness similar. However, as can be seen in our results in §VIII, once ASPA adoption reaches 20%, shortest-path export-all attacks perform significantly better than forged-origin hijacks since shortest-path export-all attacks go undetected by ASPA.

In the simulations of [74], a random set of ASes are selected to adopt ASPA, which may or may not include the victim AS. When the victim AS does not adopt ASPA (not issue ASPA records), ASPA would have no effect. We believe this blunted the effectiveness of ASPA in [74] and led to the conclusion that ASPA is only effective when tier-1 ASes deploy ASPA first. Additionally, when comparing with the same simulation settings, [74] finds that accidental route leaks have a success rate of only 1% when no defenses are deployed, whereas our work and prior works [15], [63] find this to have an attacker success rate over 10%.

B. Other Related Work

Filtering-based defenses. Several works review the practices of prefix filtering of edge ASes [31], [33], [56]. These papers discuss at length how defensive filtering using allowlists of prefixes are both nearly as effective as cryptographic defensive protocols in partial deployment and are much simpler to deploy. However, these papers also note extensive limitations associated with prefix filtering, such as the significant burden of needing to maintain a prefix allowlist.

Defenses against path manipulations. Before ASPA, many techniques have been proposed to defend against path manipulations (see surveys [8], [38], [44], [61], [81]). Specifically, these defenses include S-BGP [50] soBGP [97], psBGP [68], pgBGP [48], IRV [34], SPV [43], and Listen and Whisper [92], many others that predate BGPsec, and even BGPsec extensions such as BGP-iSec [63]. In this paper, we particularly compare ASPA with several techniques, including BGPsec, BGP-iSec, and Path-End against path manipulations, since BGPsec is currently an IETF standardized approach, BGP-iSec is a recent extension, and Path-End can be regarded as a variant of ASPA.

Defenses against route leaks. Existing practice against route leaks includes using filtering rules at routers (e.g., [88]), Peerlock and Peerlock-lite [51], [60], [82]. These approaches often involve manual efforts and hence are not scalable. Other approaches have been proposed, some are cryptographic-based (e.g., [86], [93]), and some are based on inspecting route information logs to detect route leaks [36], [37], [80], [98], [100]. In this paper, we compare ASPA with OTC since OTC is recent IETF proposal and has seen deployment in practice.

Other ASPA related works. After our submission for publication, a preprint evaluating ASPA was accepted for publication which agrees with our conclusions that ASPA is effective at preventing forged-origin hijacks [6]. Even though ASPA is still an Internet-Draft, several ASes are already deploying ASPA [16] and ASPA has already prevented a route leak [79].

ASRA. As mentioned earlier, a recent proposal, ASRA [25], [85], extends ASPA and allows for the publication of other relationships such as customers and peers. We believe ASPAwN provides equivalent or similar benefits as ASRA-Alg. B.

XI. Conclusion

In this paper, using recent real-world ROV measurements, we have shown that post-ROV attacks are already stronger than prefix hijacks, and hence it is important to address post-ROV attacks even today. We then evaluated ASPA and several alternatives for defending against post-ROV attacks. Contrary to prior works [74], [95], we showed that ASPA *is effective* against both forged-origin and shortest-path export-all attacks under partial adoption, even when none of the tier-1 ASes adopts ASPA. Our findings motivate ASPA adoption at edge and intermediate ASes. On the other hand, our results show that ASPA is not more effective than OTC at preventing *unintentional* route leaks.

We further presented, ASPAwN, as an extension to ASPA to prevent ASes from hijacking their customers, and first-ASN-stripping hijack, a powerful post-ROV attack. Further work is required to evaluate the feasibility and impact of chosen-location attacks on BGP, as well as the usage and impact of selected-provider export policies.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive and insightful feedback, as well as our anonymous shepherd for their helpful guidance. We would also like to thank Sriram Kotikalapudi for his comments and suggestions on earlier drafts and the final version of this paper, and thank Joel Halpern, Nils Rodday, and Professor Naoto Yanai for their

feedback. This work was partially based on work supported by the National Science Foundation under Grant No. 2247810, and Amir Herzberg was also supported by an endowment from Comcast. The opinions expressed in the paper are those of the researchers and not of their university or funding sources.

REFERENCES

[1] R. Anwar, H. Niaz, D. Choffnes, I. Cunha, P. Gill, and E. Katz-Bassett, "Investigating Interdomain Routing Policies in the Wild," in *Proc. of ACM IMC*, Oct. 2015.

[2] Arista, "EOS 4.31.1F user manual," 2024. [Online]. Available: https://www.arista.com/en/um-eos/eos-border-gateway-protocol-bgp

[3] A. Azimov, E. Bogomazov, R. Bush, K. Patel, J. Snijders, and K. Sriram, "BGP AS_PATH Verification based on Autonomous System Provider Authorization (ASPA) Objects," 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/17/

[4] ——, "BGP AS_PATH Verification based on Autonomous System Provider Authorization (ASPA) Objects," 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/19/

[5] A. Azimov, E. Bogomazov, R. Bush, K. Patel, and K. Sriram, "Route Leak Prevention and Detection Using Roles in UPDATE and OPEN Messages," RFC 9234 (Proposed Standard), RFC Editor, Fremont, CA, USA, May 2022. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9234.txt

[6] S. Barrett, C. Idom, G. Villafuerte, A. Byers, and B. Gulmezoglu, "Ain't How You Deploy: An Analysis of BGP Security Policies Performance Against Various Attack Scenarios with Differing Deployment Strategies," in *Proceedings of the 11th International Symposium on Networks, Computers, and Communications (ISNCC'24)*, 08 2024.

[7] BGPStream, "BGPMON's BGP Stream incident alert service," https://bgpstream.com, 2018.

[8] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford, "A survey of BGP security issues and solutions," *Proceedings of the IEEE*, vol. 98, no. 1, pp. 100–122, 2010.

[9] K. Butler, P. McDaniel, and W. Aiello, "Optimizing BGP security by exploiting path stability," in *Proceedings of ACM Conference on Computer and Communications Security (CCS)*. New York, NY, USA: ACM, 2006, pp. 298–310. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180442

[10] CAIDA, "CAIDA Serial 2 Data Set," CAIDA, Apr. 2022. [Online]. Available: https://publicdata.caida.org/datasets/as-relationships/serial-2/

[11] T. Chung, E. Aben, T. Bruijnzeels, B. Chandrasekaran, D. Choffnes, D. Levin, B. Maggs, A. Mislove, R. van Rijswijk-Deij, J. Rula, and N. Sullivan, "RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins," in *Proc. of IMC*. ACM, 2019.

[12] Cisco, "Support for enforce-first-as per-neighbor," 2013, accessed: 2024-07-10. [Online]. Available: https://bst.cloudapps.cisco.com/bugsearch/bug/CSCui91001

[13] Cisco Systems, "Comments of CISCO systems, inc. to FCC PS docket 22-90, in the matter of secure internet routing," April 2022.

[14] Cloudflare, "Is BGP Safe Yet?" 2024. [Online]. Available: https://isbgpsafeyet.com/

[15] A. Cohen, Y. Gilad, A. Herzberg, and M. Schapira, "Jumpstarting BGP Security with Path-End Validation," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 342–355. [Online]. Available: https://doi.org/10.1145/2934872.2934883

[16] R. C. Console, "ASPA (Autonomous System Provider Authorization)," 2024, accessed: 2024-10-28. [Online]. Available: https://console.rpki-client.org/aspa.html

[17] R. de Boer and J. de Koning, "BGP Origin Validation (RPKI)," Univeristy of Amsterdam, Systems and Network Engineering Group, Tech. Rep., July 2013.

[18] J. Durand, I. Pepelnjak, and G. Doering, "BGP Operations and Security," RFC 7454 (Best Current Practice), RFC Editor, Fremont, CA, USA, Feb. 2015. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7454.txt

[19] O. Filip, "enforce-first-as disable - BIRD-users mailing list discussion," https://bird.network.cz/pipermail/bird-users/2019-February/013100.html, Feb 2019, accessed: 2024-03-20.

[20] O. Filip, M. Mares, M. Matejka, and O. Zajicek, *BIRD User Guide*, BIRD Internet Routing Daemon, 2024, accessed: 2024-03-20. [Online]. Available: https://bird.network.cz/?get_doc&f=bird.html&v=20

[21] J. Furuness, "ASPA Evaluation Github Repository," https://github.com/jfuruness/aspa_eval, 2024.

[22] ——, "ROV Collector Github Repository," https://github.com/jfuruness/aspa_eval, 2024.

[23] J. Furuness, C. Morris, R. Morillo, A. Herzberg, and B. Wang, "BGPy: The BGP Python Security Simulator," in *Proceedings of the 16th Cyber Security Experimentation and Test Workshop*, ser. CSET '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 41–56. [Online]. Available: https://doi.org/10.1145/3607505.3607509

[24] L. Gao and J. Rexford, "Stable Internet Routing without Global Coordination," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 681–692, 2001.

[25] N. Geng, "Autonomous system relationship authorization (asra)," APNIC 56 Routing Security SIG, Tech. Rep., September 2023. [Online]. Available: https://conference.apnic.net/56/assets/files/APJS642/autonomous-system-re_1694481446.pdf

[26] N. Geng, K. Sriram, and M. Huang, "A Profile for Autonomous System Relationship Authorization (ASRA)," Internet Engineering Task Force, Internet-Draft draft-geng-sidrops-asra-profile-00, October 2024, work in Progress. [Online]. Available: https://www.ietf.org/id/draft-geng-sidrops-asra-profile-00.html

[27] Y. Gilad, A. Cohen, A. Herzberg, M. Schapira, and H. Shulman, "Are We There Yet? On RPKI's Deployment and Security," in *NDSS*. San Diego, CA: The Internet Society, 2017, pp. 1–14.

[28] P. Gill, M. Schapira, and S. Goldberg, "Let the market drive deployment: A strategy for transitioning to BGP security," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 14–25, 2011.

[29] ——, "A survey of interdomain routing policies," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 1, p. 28–34, dec 2014. [Online]. Available: https://doi.org/10.1145/2567561.2567566

[30] S. Goldberg, "Why is it taking so long to secure Internet routing?" *Queue*, vol. 12, no. 8, p. 20, 2014.

[31] ——, "Why is it taking so long to secure internet routing?" *Communications of the ACM*, vol. 57, pp. 56–63, 09 2014.

[32] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford, "How secure are secure interdomain routing protocols?" *Computer Networks*, vol. 70, pp. 260–287, 2014.

[33] ——, "How secure are secure interdomain routing protocols," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 87–98, aug 2010. [Online]. Available: https://doi.org/10.1145/1851275.1851195

[34] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. D. McDaniel, and A. D. Rubin, "Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing," in *NDSS*. The Internet Society, 2003. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/5.pdf

[35] D. Goodin, "Russian-controlled telecom hijacks financial services' internet traffic," Apr 2017. [Online]. Available: https://arstechnica.com/information-technology/2017/04/russian-controlled-telecom-hijacks-financial-services-internet-traffic/

[36] A. Gurney, A. Haeberlen, W. Zhou, M. Sherr, and B. Loo, "Having your cake and eating it too: routing security with privacy protections," in *ACM Workshop on Hot Topics in Networks(HotNets)*, 2011.

[37] A. Haeberlen, I. Avramopoulos, J. Rexford, and P. Druschel, "Netreview: detecting when interdomain routing goes wrong," in *USENIX Symposium on Networked Systems Design and Implementation (NDSI)*, 2009.

[38] A. Herzberg, M. Hollick, and A. Perrig, "Secure Routing for Future Communication Networks (Dagstuhl Seminar 15102)," *Dagstuhl Reports*, vol. 5, no. 3, pp. 28–40, 2015. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/5267

[39] L. Hill, "Enforcing First AS in BGP," 2023. [Online]. Available: https://lkhill.com/enforce-first-as/

[40] T. Hlavacek, I. Cunha, Y. Gilad, A. Herzberg, E. Katz-Bassett, M. Schapira, and H. Shulman, "DISCO: Sidestepping RPKI's deployment barriers," in *Proceedings of the 2020 Network and Distributed System Security (NDSS) Symposium*. San Diego, CA: NDSS, February 2020, pp. 1–14.

[41] T. Hlavacek, H. Shulman, N. Vogel, and M. Waidner, "Keep your friends close, but your routeservers closer: Insights into rpki validation in the internet," in *Proceedings of the 32nd USENIX Conference on Security Symposium*, ser. SEC '23. USA: USENIX Association, 2023.

[42] P. C. House, "Packet clearing house (pch)," 2024, accessed: 2024-03-20. [Online]. Available: https://www.pch.net/

[43] Y.-C. Hu, A. Perrig, and M. A. Sirbu, "SPV: secure path vector routing for securing BGP," in *SIGCOMM*, 2004.

[44] G. Huston, M. Rossi, and G. Armitage, "Securing BGP: A literature survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 199–222, 2011.

[45] E. Jasinska, N. Hilliard, R. Raszuk, and N. Bakker, "Internet Exchange BGP Route Server," RFC 7947 (Proposed Standard), RFC Editor, Fremont, CA, USA, Sep. 2016. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7947.txt

[46] Juniper, "Junos CLI reference," 2023. [Online]. Available: https://www.juniper.net/documentation/us/en/software/junos/cli-reference/topics/ref/statement/enforce-first-as-edit-protocols.html

[47] Juniper Networks, "Comments of Juniper Networks re: secure internet routing (FCC PS docket 22-90)," April 2022.

[48] J. Karlin, S. Forrest, and J. Rexford, "Autonomous security for autonomous systems," *Computer Networks*, Oct. 2008.

[49] S. Kent and K.seo, "An Infrastructure to Support Secure Internet Routing," Internet Requests for Comments, The Internet society, RFC 6480, February 2012. [Online]. Available: http://tools.ietf.org/html/rfc6480

[50] S. T. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000.

[51] M. Kowalski and W. Mazurczyk, "Toward the mutual routing security in wide area networks: A scoping review of current threats and countermeasures," *Computer Networks*, vol. 230, p. 109778, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623002232

[52] A. Labs, "Rpki," 2023. [Online]. Available: https://stats.labs.apnic.net/rpki

[53] M. Lepinski (Ed.) and K. Sriram (Ed.), "BGPsec Protocol Specification," RFC 8205 (Proposed Standard), RFC Editor, Fremont, CA, USA, Sep. 2017, updated by RFC 8206. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8205.txt

[54] W. Li, Z. Lin, M. I. A. Khan, E. Aben, R. Fontugne, A. Phokeer, and T. Chung, "RoVista: Measuring and Understanding the Route Origin Validation (ROV) in RPKI," in *Proceedings of the ACM Internet Measurement Conference (IMC'23)*. Montreal, Canada: ACM, October 2023, pp. 1–14.

[55] R. Lychev, S. Goldberg, and M. Schapira, "BGP security in partial deployment: Is the juice worth the squeeze?" *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 171–182, 2013.

[56] R. Lychev, M. Schapira, and S. Goldberg, "Rethinking security for internet routing," *Communications of the ACM*, vol. 59, pp. 48–57, 09 2016.

[57] H. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane Nano: Path Prediction for Peer-to-Peer Applications," in *Proce of NSDI*, 2009.

[58] J. Mauch, "https://puck.nether.net/bgp/leakinfo.cgi," 2024. [Online]. Available: https://puck.nether.net/bgp/leakinfo.cgi

[59] R. Mazloum, M. Buob, J. Auge, B. Baynat, D. Rossi, and T. Friedman, "Violation of Interdomain Routing Assumptions," in *Proc. of Passive and Active Measurement Conference (PAM)*, March 2014.

[60] T. McDaniel, J. M. Smith, and M. Schuchard, "Peerlock: Flexsealing BGP," https://arxiv.org/abs/2006.06576, June 2020.

[61] A. Mitseva, A. Panchenko, and T. Engel, "The state of affairs in BGP security: A survey of attacks and defenses," *Computer Communications*, vol. 124, pp. 45–60, June 2018.

[62] R. Morillo, J. Furuness, C. Morris, J. Breslin, A. Herzberg, and B. Wang, "ROV++: Improved deployable defense against BGP hijacking," in *Network and Distributed System Security (NDSS) Symposium*. Virtual Event: NDSS, 2021, pp. 1–14.

[63] C. Morris, S. Secondo, A. Herzberg, and B. Wang, "BGP-iSec: improved security against Post-ROV routing attacks," in *Network and Distributed System Security (NDSS) Symposium*. San Diego, CA: NDSS, 2024, pp. 1–14.

[64] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig, "Building an AS-topology model that captures route diversity," in *Proc. of SIGCOMM*, 2006.

[65] National Institute of Standards and Technology (NIST), "NIST RPKI Monitor, version 2.0," https://rpki-monitor.antd.nist.gov/, 2024, accessed February 2024.

[66] D. Nicol, S. Smith, and M. Zhao, "Efficient security for BGP route announcements," Dartmouth College, Computer Science Department, Tech. Rep. TR2003-440, 01 2003.

[67] D. M. Nicol, S. W. Smith, and M. Zhao, "Evaluation of efficient security for BGP route announcements using parallel simulation," *Simulation Modelling Practice and Theory*, vol. 12, no. 3, pp. 187 – 216, 2004, modeling and Simulation of Distributed Systems and Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X04000383

[68] P. C. v. Oorschot, T. Wan, and E. Kranakis, "On interdomain routing security and pretty secure BGP (psbgp)," *ACM Transactions on Information and System Security (TISSEC)*, vol. 10, no. 3, pp. 1–41, 2007.

[69] C. Perkins, P. Calhoun, and J. Bharatia, "Mobile IPv4 Challenge/Response Extensions (Revised)," RFC 4721 (Proposed Standard), RFC Editor, Fremont, CA, USA, Jan. 2007. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4721.txt

[70] Y. Rekhter and P. Gross, "Application of the Border Gateway Protocol in the Internet," RFC 1772 (Draft Standard), RFC Editor, Fremont, CA, USA, Mar. 1995. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1772.txt

[71] Y. Rekhter (Ed.), T. Li (Ed.), and S. Hares (Ed.), "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), RFC Editor, Fremont, CA, USA, Jan. 2006, updated by RFCs 6286, 6608, 6793, 7606, 7607, 7705, 8212, 8654, 9072. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4271.txt

[72] A. Reuter, R. Bush, I. Cunha, E. Katz-Bassett, T. C. Schmidt, and M. W"ahlisch, "Towards a rigorous methodology for measuring adoption of rpki route validation and filtering," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 1, pp. 19–27, 2018.

[73] RIPE, "RIPE NCC. Routing Information Service (RIS)," https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris, 2024.

[74] N. Rodday, G. D. Rodosek, A. Pras, and R. van Rijswik-Deij, "Exploring the benefit of path plausibility algorithms in BGP," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*. Seoul, South Korea: IEEE, 2024, pp. 1–14.

[75] N. Rodday, Ítalo S. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt, and M. Wählisch, "Revisiting rpki route origin validation on the data plane," in *5th Network Traffic Measurement and Analysis Conference, TMA 2021, Virtual Event, September 14-15, 2021*, V. Bajpai, H. Haddadi, and O. Hohlfeld, Eds. Virtual Event: IFIP, 2021, pp. 1–14. [Online]. Available: http://dl.ifip.org/db/conf/tma/tma2021/tma2021-paper11.pdf

[76] RouteViews, "University of Oregon Route Views Project," http://www.routeviews.org/routeviews/, 2018.

[77] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti, "ARTEMIS: Neutralizing BGP hijacking within a minute," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2471–2486, 2018.

[78] H. Shulman, N. Vogel, and M. Waidner, "Poster: Insights into global deployment of rpki validation," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 3467–3469. [Online]. Available: https://doi.org/10.1145/3548606.3563523

[79] A. Siddiqui, "Unpacking the first route leak prevented by aspa," 2023, accessed: 2024-10-28. [Online]. Available: https://manrs.org/2023/02/unpacking-the-first-route-leak-prevented-by-aspa/

[80] M. Siddiqui, D. Montero, S.-G. R., and M. Yannuzzi, "Self-reliant detection of route leaks in inter-domain routing," *Computer Networks*, vol. 82, pp. 135–155, 2015.

[81] M. S. Siddiqui, D. Montero, R. Serral-Gracia, X. Masip-Bruin, and M. Yannuzzi, "A survey on the recent efforts of the Internet Standardization Body for securing inter-domain routing," *Computer Networks*, vol. 80, pp. 1–26, April 2015.

[82] J. Snijders, "NTT peer locking," http://instituut.net/~job/peerlock_manual.pdf, 2016.

[83] ——, "Estimating the timeline for aspa deployment," 2023, accessed: 2024-04-18. [Online]. Available: https://manrs.org/2023/05/estimating-the-timeline-for-aspa-deployment/

[84] K. Sriram, "ASPA-based BGP AS_PATH Verification and Route Leaks Solution," 2023, presented at NANOG 89, San Diego, USA, October 2023. Accessed: 2024-12-04. [Online]. Available:

https://storage.googleapis.com/site-media-prod/meetings/NANOG89/4809/20231017_Sriram_Aspa-Based_Bgp_As_Path_v1.pdf
[85] K. Sriram, N. Geng, and A. Herzberg, "Autonomous System Relationship Authorization (ASRA) as an Extension to ASPA for Enhanced AS Path Verification," Internet Engineering Task Force, Internet-Draft draft-sriram-sidrops-asra-verification, October 2024, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-sriram-sidrops-asra-verification/
[86] K. Sriram and D. Montgomery, "Enhancement to BGPsec for protection against route leaks," 2014, draft-sriram-route-leak-protection-00.
[87] K. Sriram, D. Montgomery, D. McPherson, E. Osterweil, and B. Dickson, "Problem Definition and Classification of BGP Route Leaks," RFC 7908 (Informational), RFC Editor, Fremont, CA, USA, Jun. 2016. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7908.txt
[88] K. Sriram and D. Montgomery, "Resilient Interdomain Traffic Exchange: BGP Security and DDoS Mitigation," https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-189.pdf, 2019.
[89] V. K. Sriram and D. Montgomery, "Design and analysis of optimization algorithms to minimize cryptographic processing in BGP security protocols," *Computer communications*, vol. 106, pp. 75–85, 2017.
[90] T. Strickx, "How Verizon and a BGP Optimizer Knocked Large Parts of the Internet Offline Today," 2019. [Online]. Available: https://blog.cloudflare.com/how-verizon-and-a-bgp-optimizer-knocked-large-parts-of-the-internet-offline-today
[91] S. Strowes, "BGP Even-More Specifics in 2017," *RIPE Labs*, 2017. [Online]. Available: https://labs.ripe.net/author/stephen_strowes/bgp-even-more-specifics-in-2017/
[92] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz, "Listen and whisper: Security mechanisms for BGP," in *Proc. of NSDI*, 2004.
[93] S. Sundaresan, R. Lychev, and V. Valancius, "Preventing attacks on BGP policies: One bit is enough," Georgia Institute of Technology, Tech. Rep. GT-CS-11-07, 2011.
[94] C. Systems, "BGP commands on Cisco IOS XR software," 2011. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/routers/xr12000/software/xr12k_r4-1/routing/command/reference/routing_cr41xr12k_chapter1.html
[95] N. Umeda, T. Kimura, and N. Yanai, "The juice is worth the squeeze: Analysis of autonomous system provider authorization in partial deployment," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 269–306, 2023.
[96] M. Wählisch, O. Maennel, and T. C. Schmidt, "Towards detecting BGP route hijacking using the RPKI," in *Proc. of ACM SIGCOMM*, 2012, pp. 103–104. [Online]. Available: http://doi.acm.org/10.1145/2342356.2342381
[97] R. White, "Securing BGP through secure origin BGP (sobgp)," *Business Communications Review*, vol. 33, no. 5, pp. 47–47, 2003.
[98] M. Zhang, V. Giotsas, and C. Martinho, "How we detect route leaks and our new cloudflare radar route leak service," 2022, accessed: 2024-11-21. [Online]. Available: https://blog.cloudflare.com/route-leak-detection-with-cloudflare-radar/
[99] M. Zhang and J. Meggitt, "BGPKIT Parser: Mrt/bgp data parser written in rust," https://github.com/bgpkit/bgpkit-parser, 2024.
[100] M. Zhao, W. Zhou, A. Gurney, A. Haeberlen, M. Sherr, and B. Loo, "Private and verifiable interdomain routing decisions," in *ACM SIGCOMM*, 2012.
[101] M. Zhao, S. W. Smith, and D. M. Nicol, "Aggregated path authentication for efficient BGP security," in *Proc. of CCS*, 2005.

Appendix A
Other ROV Deployment Results

In addition to the dataset in [41], we used measurements from several other datasets. Results from all the datasets show similar trends: most of the tier-1 ASes adopt ROV, and forged-origin hijack is already more effective than prefix hijack. We next only present one additional result, obtained using the dataset from [14]; the results for other datasets can be reproduced using [22]. Fig. 10 plots the percentage of ROV adoption for each type of ASes, and Fig. 11 shows the attacker success rate under the current and future ROV adoption based on the dataset from [14].

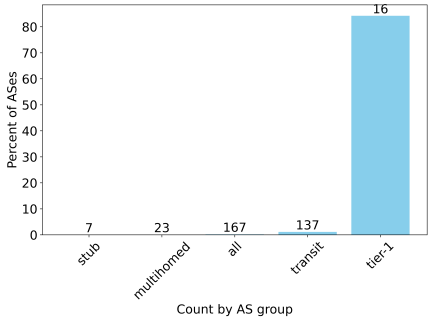


Fig. 10: ROV deployment status for the different types of ASes (obtained using measurements in [14]).

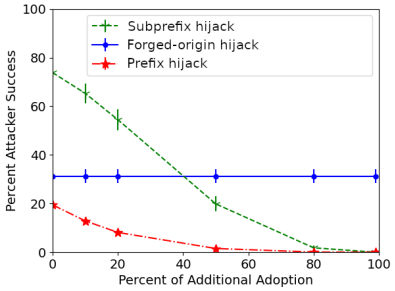


Fig. 11: Attacker success rate with current and future ROV deployment (based on measurements from [14])).

Appendix B
Proof that OTC is equivalent to ASPA for Accidental Route Leaks

Definition 1 (Adopting AS)**.** *We say an AS is adopting ASPA if they both publish their set of provider ASes (SPAS) to the RPKI and also validate paths of incoming announcements based on other ASPA records. An AS that is adopting OTC both adds OTC attributes when appropriate and drop any announcements that OTC attributes indicate as leaks.*

Definition 2 (Accidental Route Leak)**.** *For the purposes of this analysis, we consider an accidental route leak to be any violation of valley-free routing where no path manipulation has occurred. In other words, when an AS exports an announcement received from one of its providers or peers to another provider or peer, it does not modify any attributes, including the AS path.*

Definition 3 (Visible Route Leak)**.** *We consider a route leak to be* OTC-Visible *if it has an OTC attribute and is propagated to a provider or peer. We consider a route leak to be* ASPA-Visible *if the ASPA verification procedure can detect the leak.*

Theorem 1. *ASPA and OTC provide equivalent protection against accidental route leaks from non-adopting ASes, assumming that adopting ASes do not (accidentally) leak routes.*

Proof. Consider an accidental route leak that would be ASPA-Visible but not OTC-Visible with the same set of

adopting ASes. We look at the following two directions where this might be detected.

Case 1: Upstream (received from customers or peers). An adopting AS can identify a route leak traveling upstream if it has previously been announced by an adopting AS to a non-provider. Both OTC and ASPA detect this trivially.

Case 2: Downstream (received from providers). An AS that adopts OTC will not validate paths traveling downstream, i.e., received from a provider, however, an ASPA AS will. For downstream detection to occur, an adopting AS would have to receive an AS path from a provider where it can observe an upward segment of the path followed by a downward segment and another upward segment. Any observable upward segment after the downward segment would require an adopting AS to leak the announcement, while, based on our assumptions, it would not do so because it is adopting. Since all ASes that would leak an announcement must be non-adopting, then this means the route leak would not be visible at all. □

Justifying the non-leaking adopters assumption. In the proof above, we assume that adopting ASes are not accidentally leaking routes. We make this assumption because an AS that adopts route leak prevention mechanisms has already devoted substantial efforts towards preventing route leaks. The likelihood of an AS adopting such mechanisms accidentally leaking routes should be much lower than ASes that have not gone through the efforts to adopt route leak prevention mechanisms.

Of course, the proof does not hold true when adopting ASes are also leaking accidentally. However, we have simulated this case, and found that the results from ASes deploying ASPA are nearly identical to those of ASes deploying OTC attributes (with less than 0.1% difference in attacker success rate). The results are visually indistinguishable when compared to Fig. 9(a) and Fig. 9(b), so we omit them. We do however open source our scripts to generate these results in [21].

Appendix C
ASPA Origin Hijack protection from providers

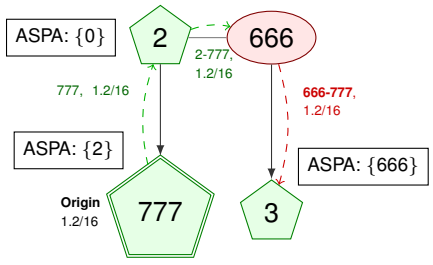


Fig. 12: Despite all non-attacker ASes adopt ASPA, an attacker (AS 666) can still announce a forged-origin hijack to its own customers without being detected. ASPAwN prevents this attack because AS 666 is not listed as a neighbor of AS 777. In this example, the attacker can win traffic from AS 3 regardless by simply forwarding the valid path; we show another example in Fig. 13.

An example of how attackers can bypass ASPA when announcing to customers (motivating ASPAwN) is shown in Fig. 12. In this example, all ASes except for the attacker (AS 666) adopt ASPA. In this case, the attacker can make a forged-origin hijack (with prefix 1.2/16 and AS-path 666-777) to its customer, AS 3, without being detected by AS 3.

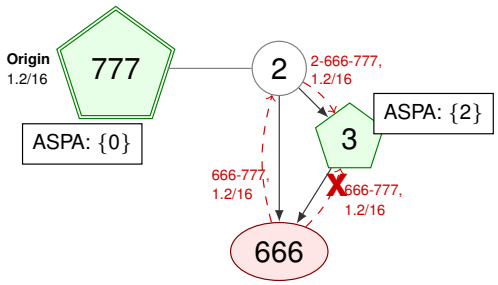


Fig. 13: Diagram showing an odd case where ASPA adopting AS 3 receives a forged-origin hijack from their customer (666) and rejects it, but then accepts the same forged-origin hijack from their non-ASPA provider (AS 2).

Another example that motivates ASPAwN is in Fig. 13. In this case, AS 666 attempts to hijack AS 3 with a forged-origin hijack. AS 3 receives an AS path of 2-666-777 from its customers, and rejects the AS path. However, AS 3 *also* receives a hijack for the same prefix, *also including the attacker AS* from its providers, and it *accepts the hijack*. This is because for the downstream verification of ASPA, AS 3 receives the AS path of 2-666- 777. It is possible that all of those ASes are a contiguous chain of customers from AS 777. It is subpar for ASPA AS 3 to reject the hijack from its customers, but then accept the same hijack from its providers. This behavior can easily be mitigated using ASPAwN. If AS 777 announced its neighbors as a set (in this case, only AS 2), then whether the hijack comes from a customer or a provider, AS 3 could reject this hijack, since AS 666 is not in AS 777's set of valid neighbors.

Appendix D
Forged-origin Hijack from a Transit AS

Fig. 14 plots the results for various defenses under forged-origin hijack from a transit AS attacker (not tier-1). The results show similar trend as Fig. 6 (for an edge AS attacker).

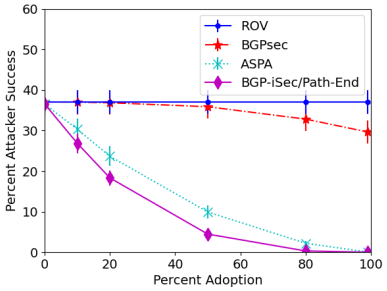


Fig. 14: Results under forged-origin hijack from a transit AS attacker (not tier-1).