iROV: Breaking the Silence of RPKI with Interactive Validation

Younsoo Kim *KAIST*
Seungjin Baek *KAIST*
Weitong Li *Virginia Tech*
Taejoong Chung *Virginia Tech*
Min Suk Kang [†] *KAIST*

Abstract

While Route Origin Validation (ROV) effectively mitigates prefix hijacking, it drops BGP updates that are inconsistent with currently published ROAs without consulting the prefix owner. This silent-drop behavior creates a serious agility gap: when a resource owner must urgently redirect traffic during DDoS mitigation, the temporary route may deviate from the currently published ROAs and be dropped before a ROA update propagates. We propose iROV (interactive ROV), a lightweight extension to ROV, to address the agility gap by transforming validation from a static local decision into an interactive, owner-involved process. iROV enables routers to query prefix owners for "Just-in-Time" authorization of invalid routes. This interactivity also allows prefix owners to receive real-time visibility feedback on ROV-invalidation events, which helps them detect remote prefix-hijacking attempts or quickly discover their own benign misconfigurations. To mitigate availability threats introduced by this interactivity, iROV employs validity stapling to suppress redundant queries, and probabilistic querying to bound owner-side query load. Large-scale simulations and a prototype implementation demonstrate that iROV restores reachability for urgent route changes from 23% to nearly 90%, and delivers rapid feedback on anomalies with granular impact estimation, while incurring negligible overhead.

1 Introduction

The Resource Public Key Infrastructure (RPKI) has emerged as a fundamental defense against BGP prefix hijacking, enabling cryptographic validation of IP prefix ownership through Route Origin Authorizations (ROAs) [37]. Route Origin Validation (ROV), the enforcement arm of this ecosystem, drops BGP updates that are inconsistent with the information represented by the ROAs. With ROA-covered prefixes encompassing 60% of the global routing table [49] and ROV protecting a significant portion (about 30% [38, 39]) of Autonomous Systems (ASes), the infrastructure has successfully raised the bar for routing attacks.

However, we argue that the current ROV architecture suffers from a fundamental limitation; namely, it operates only as a *silent, unilateral enforcement* mechanism. When a validating router detects an invalid route, it discards the announcement based solely on its local ROA cache. This design is especially problematic in security-critical operational settings that require *urgent routing changes*. For example, during a distributed denial-of-service (DDoS) attack, a resource owner may need to redirect traffic to a scrubbing provider, causing the temporary route to deviate from the currently published ROAs [13, 31, 32, 65]. Under standard ROV, validating ASes cannot distinguish this owner-initiated mitigation maneuver from an unauthorized origin announcement; they simply drop the route as ROV-invalid. Updating ROAs through the RPKI repository can take up to tens of minutes [20], which is poorly matched to the time scale of emergency DDoS mitigation. As a result, ROV can inadvertently block the routing change that the legitimate resource owner critically needs to defend its network.

To address this agility gap, we propose iROV (*interactive* Route Origin Validation), a lightweight extension that transforms ROV from a static local decision into an owner-involved authorization process. Under iROV, a validating AS can query the legitimate resource owner AS for authoritative verification if it encounters an ROV-invalid BGP update, rather than silently and unilaterally discarding the update. If the owner cryptographically authorizes the route, the validating AS accepts and propagates the route as an owner-approved exception; otherwise, it falls back to the standard ROV behavior and drops the update. This "Just-in-Time" authorization allows legitimate emergency routing changes, such as DDoS redirection, to temporarily override stale ROA state while preserving ROV's integrity against unauthorized origin announcements. Figure 1 illustrates the high-level difference between the current silent-drop model and our proposed interactive owner-involved validation model.

[†]Corresponding author.

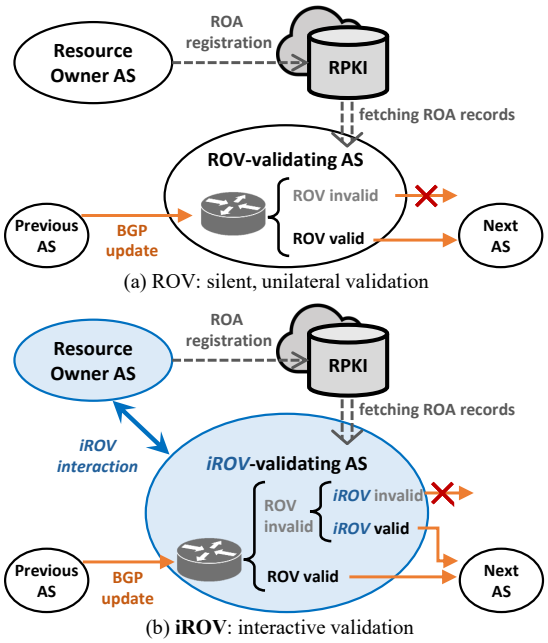

Figure 1: Comparison of (a) the unilateral silent drop model of ROV and (b) the interactive owner-involved validation model of iROV.

Beyond enabling emergency authorization, iROV also adds an owner-facing *visibility channel* that standard ROV lacks. Under the current silent-drop model, a resource owner does not learn where, when, or how often the route updates for its prefixes are rejected as ROV-invalid. With iROV, these rejection events can generate queries to the resource owner, turning ROV enforcement into an in-protocol feedback channel. This visibility provides two secondary benefits. First, when an unauthorized AS attempts to hijack the owner's prefix, the owner can observe the attack in real-time via iROV queries. The number of queries can be used to further assess the impact and spread of the attack, thus enabling informed mitigation decisions. Second, when the owner itself makes a benign configuration mistake in RPKI or BGP, such as an ASN typo or overly restrictive MaxLength setting, the resulting queries provide timely feedback that helps the owner discover and correct the mistake before the error persists for weeks or months.

The introduction of interactivity, however, raises concerns regarding new denial-of-service (DoS) attack vectors with query flooding. This is because iROV's dynamic query mechanism adds new network traffic and burden to the resource owner AS for processing incoming iROV queries. This is particularly critical because adversaries could generate many invalid BGP updates targeting a specific prefix, thereby tricking iROV-validating ASes into inadvertently flooding the target resource owner with iROV queries. iROV presents a two-fold solution to this. First, we introduce a validity-stapling mechanism, where an upstream iROV-validating AS attaches the owner's validation result to the BGP update. This allows downstream ASes to verify the validity locally without sending redundant queries. Second, we implement probabilistic querying, where routers query with a specific probability rather than deterministically. Our analysis demonstrates that this combination significantly reduces the risk of query flooding while maintaining high network reachability for owner-approved routes.

We evaluate the effectiveness of iROV across three dimensions. First, using large-scale BGP simulations over AS-relationship data, we quantify the reachability gains enabled by owner-authorized propagation. In our baseline setting, average reachability increases from 23.35% (no iROV) to near 90% when iROV is widely deployed across AS tiers. Second, leveraging the same simulation framework, we assess the resource owner's visibility into ROV invalid events (both benign mistakes and hijacking attempts). We show that the first query sent to the owner (i.e., the initial alert) typically originates from an AS within a few hops of the invalid route's originator, enabling rapid detection of both mistakes and malicious hijacks. Moreover, iROV provides accurate estimates of the impact of invalid updates and thus helps owners assess the severity of benign mistakes or ongoing attacks. Finally, we build a prototype by extending an open-source BIRD router and FORT Relying Party, and replay real-world BGP update traces for about five days, observing negligible router overhead as well as low RP CPU (about 3.7% on average) and memory (about 600 to 900 MB) usage.

In summary, we make the following contributions:

- We propose iROV, a lightweight extension to ROV that uses direct owner interaction to authorize ROV-invalid routes. iROV enables urgent owner-authorized routing changes, such as DDoS redirection, while also giving resource owners timely visibility into invalid-route events.
- We design two security mechanisms, validity-stapling and probabilistic querying, to secure iROV against query flooding attacks. These mechanisms resolve the inherent trade-off between protocol interactivity and service availability.
- We provide a large-scale simulation and a functional prototype implementation on BIRD and FORT, demonstrating its effectiveness and negligible performance overhead through extensive evaluation.

2 Background and Motivation

In this section, we begin by reviewing the studies on the existing RPKI ecosystem, particularly focusing on the adoption and operational aspects of Route Origin Validation (ROV) reported in measurement studies (§2.1). We then discuss the key challenge that is driven by the information propagation latency inherent to RPKI's architecture (§2.2), followed by other issues arising from the silent nature of standard ROV

(§2.3). Last, we present the case for introducing interactivity into ROV as an architectural solution to these challenges (§2.4).

2.1 Understanding ROV Invalids in the Wild

ROV is the de facto security enforcement mechanism against prefix hijacking attacks in the current Internet [15, 21, 54] and has drawn significant attention from the measurement study community. Several notable studies have measured the adoption of ROV enforcement [11,23,25,26,39,56], revealing that there still remains a significant portion of the Internet that does not enforce ROV. These studies have raised awareness about the importance of ROV and helped the community to adopt ROV more widely.

Additionally, several other lines of work have focused on understanding the BGP updates that are marked as invalid by ROV (or simply ROV invalids) [13, 27, 40, 58, 67]. The proportion of ROV invalids in the wild was once significant (e.g., over 60% of all ROA-covered prefixes as early as 2011) [13], but has decreased over time with adoption of ROV enforcement and increased awareness among network operators (e.g., correction of confusion regarding the MaxLength field) [13, 23]. Yet, unfortunately, it has notably stagnated at approximately 0.6% of all announced routes in recent years [39, 40].

More recent studies have investigated what causes these ROV invalids in the wild through large-scale measurements and analysis [13, 40, 58]. Main causes include: (i) benign misconfigurations by network operators and (ii) traffic engineering practices (such as redirection of traffic to DDoS mitigation services without updating ROAs accordingly [13, 58]), as well as (iii) invalids with unknown causes (about 3% of all invalids [40]), which may include actual hijacking attacks.

A few recent works have attempted to address the ROV invalids. SROV (Smart ROV) [27] infers whether an ROV invalid is benign or malicious based on heuristics, such as the lifespan of the route and the relationship between the originating AS and the announced prefix. More recently, LOV (Learning Origin Validation) [58] attempts to classify the benignity of ROV invalids using machine learning techniques. These approaches, however, force the validating AS to trust the inferences made by third-party entities, which may not always be accurate or reliable.

To that end, we delve deeper into the two architectural challenges of the current ROV enforcement model, which we discuss next.

2.2 Agility Gap in Route Authorization

The first challenge arises from the temporal gap between highly dynamic network operations and the rigid ROV enforcement model. Network operations are often real-time and urgent; operators may need to shift traffic immediately to mitigate a DDoS attack (e.g., "opaque transit" for DDoS scrubbing [40]) or bypass a network outage (e.g., "routing shifts" or "traffic engineering" [58]). In contrast, the RPKI repository ecosystem is designed for stability and consistency, with propagation delays often measured up to tens of minutes, as shown in a recent measurement study of the RPKI ecosystem [20].

In fact, DDoS mitigation service providers have expressed concerns about this agility gap. Providers, such as Lumen and Corero, have publicly stated that their DDoS mitigation effectiveness may be limited if the target prefix is ROA-protected and the customer has not authorized the provider's AS in the ROA [16, 42]. While these providers provide guidance to customers on updating ROAs for DDoS mitigation scenarios, they often face challenges convincing customers to make these changes, as many are either unwilling or lack the technical expertise; see discussions in the NANOG public forum regarding these operational challenges [43].

2.3 Lack of Visibility in Route Validation

Furthermore, the silent-drop nature of ROV enforcement offers no in-protocol mechanism for visibility into the route validation process to the resource owner AS. That is, resource owner ASes do not receive any feedback when routes to their prefixes are determined to be ROV-invalid and dropped by validating ASes. This lack of visibility creates two significant challenges for network operators.

Blind spots in hijack detection. First, the silent enforcement model of ROV hampers the detection and mitigation of actual hijacking attempts. When a malicious, unauthorized AS announces a prefix, ROV-validating ASes drop the announcement without contacting the resource owner. While this prevents the malicious message from propagating through the network, it also means that the legitimate resource owner, the victim of the hijack, remains *unaware* of the attack.

Understanding this lack of in-protocol attack visibility, network operators relied on global BGP monitoring projects [50, 57] to detect hijacking attempts [12, 36, 53, 60, 61, 68]. Yet, several recent studies show that sophisticated hijacks can evade global BGP monitoring systems and remain undetected or contain the visibility to a small subnet [7, 41, 47, 59]. These stealth attack techniques show that obtaining a precise view of hijacking attacks across all networks is impractical, making it impossible for victims to be aware of these sophisticated hijacks.

Persistence of configuration errors. Second, empirical studies show that a significant portion of ROV-invalid BGP updates are in fact *benign misconfigurations* [13, 27, 40, 67]. Common culprits include ASN typos or overly restrictive MaxLength settings in ROAs [13]. Sometimes routes are classified as invalid due to non-obvious complexities in network operations; for example, the announcement of leased prefixes may seem invalid to an ROV-performing router if the corre-

sponding ROAs are not updated properly [40].

When a validating AS drops a misconfigured route, it does so locally based on its cache. Consequently, the resource owner remains oblivious to the reachability loss until they realize it via external complaints or manual checks, often weeks or months later, as measured and reported in a recent study [40]. Complete lack of in-protocol mechanism for error reporting means that network operators must vigilantly monitor their own prefixes and manually investigate any anomalies, a process that is both error-prone and resource-intensive.

2.4 Case for Interactivity in ROV

The challenges described above stem from a single architectural constraint; i.e., the validating AS makes a drop decision silently based on cached ROA data without consulting the resource owner at enforcement time. We argue that RPKI can evolve from this static enforcement model to an *interactive, owner-involved authorization* model. iROV provides a necessary architectural evolution; namely, shifting from brittle, static enforcement to resilient and flexible *"Just-in-Time authorization."*

With iROV, a validating AS now learns the real-time intent of the target prefix's legitimate resource owner, addressing the three challenges discussed above: First, when an invalid BGP update is an honest mistake, the owner's intent (i.e., "this route should be valid") can be communicated back to the validating AS, enabling acceptance of the route and immediate action to correct the mistake. Second, when the invalid update is a hijacking attempt, the legitimate owner's intent (i.e., "this route should not be valid") can be conveyed to the validating AS, reinforcing the drop decision and facilitating rapid mitigation actions at the owner side. Third, when the invalid update is a transient and intentional operational maneuver (e.g., DDoS mitigation), the owner's intent (i.e., "this route is temporarily authorized") allows the route to be accepted without waiting for ROA propagation.

One may argue that rigorous monitoring or pre-registered contingency ROAs could mitigate these issues to some extent without protocol changes. However, relying solely on operator vigilance effectively shifts the burden of protocol rigidity onto humans, and pre-planning often fails to cover unpredictable emergency scenarios.

3 System Overview and Design Goals

We first present a high-level overview of iROV's architecture with emphasis on its main components (§3.1). Then, we define the threat model considered in this work, explicitly acknowledging the new risks introduced by interactivity in the new protocol (§3.2). Finally, we summarize the key design requirements that guide the rest of the design and implementation of iROV (§3.3).

3.1 System Components of iROV

Figure 1 provides a high-level view of iROV, and Figure 2 further illustrates the involved parties and their interactions. iROV involves three parties: an iROV-validating AS, a resource owner AS, and the existing RPKI infrastructure.

- *iROV-validating AS* is a network that enforces ROV and additionally supports iROV's interactive validation logic. It consists of two components. The first is a *BGP router* that performs standard ROV on received BGP updates and triggers iROV only when a BGP update is classified as ROV-invalid. The second is a *Relying Party* (RP) that already exists in the RPKI ecosystem and maintains validated ROA data for the router. To avoid burdening the router, iROV offloads external communication and cryptographic validation to the RP. Concretely, when the router encounters an ROV-invalid route, it queries the RP, and the RP performs the out-of-band interaction with the resource owner on the router's behalf.
- *Resource owner AS* is the legitimate holder of the prefix being validated. It operates a *respondent server* that answers iROV queries for its prefixes. The respondent server compares each query against the owner's local policy configuration, which we represent as a *whitelist* of authorized prefix–origin pairs for urgent maneuvers. Based on this whitelist, the owner returns an authoritative approval ("Yes") for an intended deviation or a denial ("No") for an unauthorized BGP update, enabling the validating AS to selectively permit or drop the route.
- *RPKI infrastructure* is intended to remain mostly unchanged in iROV. Resource owners continue to register ROAs as usual, and iROV-validating ASes continue to fetch and validate ROA objects through their RPs. iROV adds a contact information for the respondent server to the ROAs, enabling the RP to identify the correct server to query.

3.2 Threat Model

In this work, designing a practical extension to the current ROV framework, we consider an adversary model for typical routing manipulation attacks. First, we assume an adversary that can originate arbitrary BGP updates with direct or indirect control over one or more ASes or BGP speaking routers. Yet, the adversary does not possess the private keys of legitimate resource owners or compromise digital signatures bound to valid RPKI objects. We consider two primary attack vectors with distinct goals.

- *Prefix hijacking.* An adversary aims to attract traffic destined to another, legitimate resource owner AS by announcing unauthorized prefixes; that is, the attack compromises route origin integrity. This can be done by originating the legitimate owner's prefix or more specific sub-prefixes. This is the main attack vector that RPKI ROV is designed to

prevent; thus, any ROV validating AS must drop prefix-hijacking BGP updates.

- *Query flooding.* An adversary aims to disrupt the availability of the deployed iROV; thus, this attack is unique to iROV's interactive design. By announcing a large number of invalid BGP updates that target a resource owner AS, the adversary attempts to overwhelm the owner with iROV queries from many validating ASes.

We consider several threats as outside of the scope (or *non-goals*) of this work. First, we do not consider hijacking attacks that are fundamentally outside the protection scope of ROV. For example, AS path manipulation (e.g., adding or removing intermediate ASes while keeping the origin AS unchanged) is not prevented by ROV and is thus outside the scope of iROV. Second, we do not consider direct denial-of-service attacks against iROV respondent servers, RPKI repositories, or validating routers. Third, we do not consider active protocol interference by non-participating ASes as our main threat model.

3.3 Design Goals

Our design is guided by the following requirements, stated as testable goals that map directly to our security analysis and evaluation:

G1: *Preserve ROV's anti-prefix-hijack guarantees.* iROV must not relax ROV's core guarantee against prefix hijacking. In particular, iROV must not create a new opportunity for an adversary to turn a hijack into an accepted route.

G2: *Enable authoritative propagation with high reachability.* When a route is ROV-invalid but legitimately intended by the prefix owner, iROV should allow the owner to explicitly authorize it and thereby recover reachability that would otherwise be lost. This goal targets operational scenarios such as urgent, transient routing maneuvers.

G3: *Keep query overhead manageable under abuse.* Interactivity introduces a new availability risk, where adversaries can try to overwhelm a target owner by inducing many iROV queries. iROV should bound the owner-side processing burden.

G4: *Trigger fast first-time query.* When an invalid BGP update, especially the owner-unintended ones (e.g., benign misconfigurations or hijacks), appears in the network, iROV should provide timely feedback to the legitimate owner. Fast first-time query is critical for alerting owners about potential misconfigurations or hijacks.

G5: *Offer reliable risk estimation.* The feedback offered by iROV should be informative enough for owners to reason about the operational impact of the associated invalid BGP update. In particular, when it is a malicious hijacking attempt, iROV should help owners estimate the potential reachability impact due to the hijack, enabling them to plan mitigation or recovery actions accordingly.

G6: *Ensure negligible router and RP overhead.* iROV must remain practical for deployment in operational networks. It should minimize changes to the router's critical path, and keep computational cost for both the router and RP negligible.

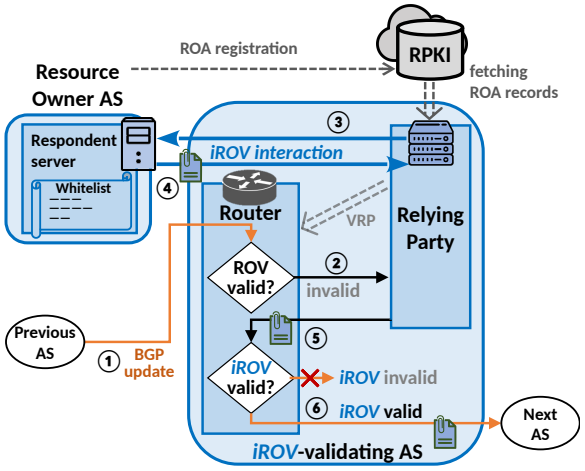

Figure 2: Overview of the iROV system architecture and the interactive validation workflow between the router, RP, and resource owner.

4 iROV Protocol Design

This section details the design and operation of the iROV protocol. We first describe the interactive validation process where a validating AS communicates with a resource owner to resolve ambiguities regarding ROV-invalid routes in §4.1. Then, we detail the validity stapling mechanism, which leverages BGP transitive attributes to propagate authoritative decisions and suppress redundant queries downstream in §4.2. In §4.3, we explain probabilistic querying, which serves to further reduce the number of queries. iROV's failure handling policy is discussed in §4.4, and, finally, we propose some protocol standardizations for iROV in §4.5.

4.1 Interactive Validation Protocol

The core of iROV is an on-demand verification loop that triggers only when a router encounters a BGP update classified as *invalid* by standard ROV. As illustrated in Figure 2, this process involves three key components: the BGP router at the validating AS, the RP software, and the respondent server at the resource owner AS.

To maintain router efficiency, iROV offloads the heavy lifting of external communication to the RP. The router remains responsible for high-speed packet processing, while the RP handles the asynchronous web queries to the resource owner.

The protocol operation proceeds in six steps, corresponding to the labels in Figure 2:

① *Detection of ROV invalid*: The process begins when the *BGP router* at the iROV-validating AS receives a BGP update. The router performs standard ROV. If the route is deemed *invalid* (e.g., due to a prefix length mismatch or ASN mismatch), the iROV logic is triggered instead of immediately dropping the route.

② *Internal Router-RP Query*: The router suspends the decision for this specific prefix and issues a query to its local *RP*. This communication utilizes an extended version of the RPKI-to-Router (RTR) protocol, introducing a new Protocol Data Unit (PDU) type that carries the prefix, prefix length, and origin ASN of the suspect BGP update.

③ *External Inquiry*: Upon receiving the query, the RP sends an out-of-band query to the *respondent server* hosted by the resource owner. The address information of respondents are registered on RPKI (more discussion in §8). As the RP routinely syncs with RPKI repositories, it will naturally acquire respondent addresses as well, enabling it to contact respondent servers of all iROV adopting resource owners. To ensure accessibility and ease of deployment, the respondent server operates as a standard web server.

④ *Authoritative Verification*: The *respondent server* processes the incoming query against a local policy configuration. This server maintains a *whitelist* of prefixes or ASN pairs authorized for urgent maneuvers (e.g., DDoS mitigation routes). If the route matches an entry in the whitelist, the server generates a "Yes" response; otherwise, it generates a "No" response. The resulting iROV response consists of this verdict (Yes/No), timestamp, the original query (prefix and ASN), and an *RPKI Signed Checklist (RSC)* [62] object that attests the authenticity of the response. The total size of a BGP update that contains an iROV response stays well within the practical limit of 4,096 bytes, the maximum size of a BGP message. Details of the response format and size feasibility are provided in §5.3.

⑤ *Response Relay*: The signed iROV response is returned to the RP, which validates its RSC signature. The RP then relays the verdict and iROV response back to the BGP router via RTR.

⑥ *Enforcement and Stapling*: Finally, the router enforces the decision. If the response is "No", the router drops the BGP update, preserving standard ROV security. If the response is "Yes", the router overrides the ROV *invalid* state and accepts the route. Before forwarding the route to neighbors, the router *staples* the iROV response to the BGP update.

4.2 Validity Stapling and Propagation

A naive implementation of interactive validation could force every AS on the downstream to query the resource owner independently. This not only introduces significant latency and overhead but also risks overwhelming the resource owner's server with redundant requests for the same route validation. To mitigate this issue, iROV implements *Validity Stapling*, as illustrated in Figure 3.

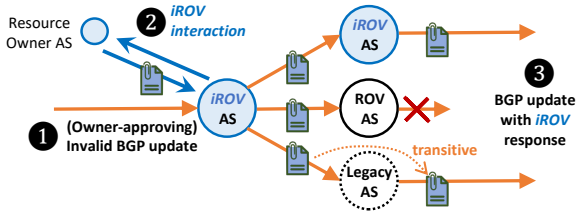

Figure 3: Illustration of the validity stapling mechanism, enabling the propagation of authorized iROV response to downstream ASes (including legacy BGP routers) via transitive attributes.

To achieve efficient propagation, iROV leverages the extensible design of BGP path attributes, which are categorized into well-known and optional [55]. Crucially, *optional transitive* attributes are designed to be preserved and forwarded by routers even if they do not recognize the specific attribute type. Thus, by encapsulating the iROV authorization within such an attribute, we ensure that the proof of validity survives as it traverses non-participating (legacy) nodes in the network.

Once an iROV-validating AS receives a BGP update of an invalid route (❶ in Figure 3) and successfully obtains a signed authorization from the resource owner (❷), it encapsulates this response into an optional transitive BGP path attribute. This *stapled* proof travels with the route BGP update, creating different propagation behaviors depending on the receiving AS's capability (❸):

- *iROV-enabled AS*: When a downstream iROV AS receives a BGP update with an iROV response, it enters a *read-stapled* mode. The router extracts the signed iROV response and validates it via the local RP without initiating a new query to the resource owner. By validating the iROV response locally, downstream ASes eliminate the latency of external interactions and shield the resource owner from redundant traffic.
- *ROV-only AS*: An AS that enforces strict ROV but lacks iROV logic will not recognize the stapled iROV response. It will assess the route solely against the cached ROA data, mark it *invalid*, and drop it.
- *Legacy AS*: ASes that have not deployed ROV (or iROV) do not validate route origins. Adhering to the transitive forwarding behavior described above [55], they forward the iROV response attribute to other routers although they do not recognize it. Previous studies [24, 48] showed that the vast majority of BGP routers behave according to the specification, with only about 2% of routers discarding unknown attributes. Consequently, legacy ASes act as *passive*

carriers, propagating the invalid route along with its iROV response to other parts of the network.

The reachability of an invalid update with the owner's authorization therefore depends on the ratio of iROV and legacy ASes versus rigid ROV-only ASes. Since iROV ASes validate and propagate the route, and legacy ASes blindly forward it, the route can successfully traverse substantial portions of the Internet despite the presence of invalid ROAs.

4.3 Probabilistic Querying

Validity stapling reduces redundant downstream queries once an upstream iROV AS has obtained an authorization. However, a resource owner can still receive many *first* queries when an invalid BGP update initially fans out to many iROV ASes. To further bound owner-side load, iROV supports *probabilistic querying*.

When an iROV AS receives an ROV-invalid BGP update that does not carry a valid iROV response, it queries the resource owner with probability $p \in [0,1]$. If it queries, it runs the interactive validation protocol and accepts the route only upon a verified positive authorization. If it does not query, it drops the BGP update immediately, matching the standard ROV behavior. The probability $p = 1$ makes iROV query for every invalid route, while $p = 0$ reduces iROV to the standard ROV in handling an invalid update. To avoid inconsistent behavior across repeated observations, an iROV AS makes a deterministic decision per (prefix, origin ASN) pair. It computes a hash-derived value $h \in [0,1)$ and queries only if $h < p$. This keeps the policy stable across time and across multiple neighbors.

A higher p improves the chance that an upstream AS quickly obtains an authorization and staples it, improving reachability for authorized invalids. A lower p reduces expected query load on the owner. Operators can use tier-aware settings, for example, higher p at a small number of high-degree ASes and lower p at the edge. Probabilistic querying limits how many ASes attempt initial contact, while stapling prevents repeated contact after an authorization is obtained.

While small values of p (e.g., 0.1) might appear to limit interactivity (effectively reverting to standard ROV behavior in the overall Internet), our evaluation in §7.3 demonstrates otherwise. We show that even with $p = 0.1$ across all Tier-2 and Tier-3 ASes, iROV still achieves high reachability when the legitimate resource owner authorizes invalid routes. This is because (1) some iROV-validating ASes still perform direct query and obtain valid iROV responses, and the resulting stapled responses propagate through iROV and legacy ASes, and (2) ROV-invalid updates still propagate through legacy ASes that forward invalid routes without validation, thus reaching many iROV ASes that may query the resource owner.

4.4 Failure Handling

iROV depends on timely owner responses. Our failure handling follows a simple rule. When anything fails, iROV *falls back* to standard ROV behavior. A validating AS starts a timer after triggering iROV for an ROV-invalid BGP update. If no verified positive authorization arrives before timeout, it drops the route. If an iROV response fails verification or does not match the queried contents (prefix, origin ASN), it is rejected and the route is dropped. A response carries a timestamp, which can be used to calculate its expiration time based on a configurable validity window. An iROV response that has expired will not be accepted. Hence, replay attacks are effectively prevented. If an iROV response is missing or stripped, the downstream iROV AS simply applies probabilistic querying as usual. To avoid repeated work under churn, an iROV AS caches recent results for a short TTL. Repeated observations of the same (prefix, origin ASN) pair reuse the cached verdict instead of re-triggering external queries.

In case of route or domain manipulation attacks (e.g., a legitimate resource owner's prefix is already hijacked or the respondent server's domain is poisoned), a validating AS would fail to obtain a valid iROV response and thus drop the route, effectively falling back to standard ROV behavior. Should there exist a need to contact the resource owner through alternative means (e.g., for reporting operational issues), one could consider contacting the email address listed in the iROV-extended ROA; refer to §8.

4.5 Required Protocol Standardization

iROV requires minimal, backward-compatible extensions to two existing standards: the ROA object and the RTR protocol.

ROA Object. First, to establish a trusted source for discovering respondent servers, we propose extending the ROA object. Currently, an ROA identifies authorized prefixes; we simply append the respondent server's network identity (e.g., IP address of domain name) to this object along with a contact information (e.g., email).

Standard ROA:

`{prefix, MaxLength, asn}`

Extended ROA (iROV):

`{prefix, MaxLength, asn, iROV-server, email}`

`iROV-server` and `email` fields are left optional for backward-compatibility.

This approach allows an RP to build a "prefix-to-iROV lookup table" in parallel with the Validated ROA Payload (VRP) when traversing the RPKI repository, removing the need for separate data preparation phases.

RTR protocol. Second, to offload cryptographic validation from routers, we utilize the RP as a liaison between the router and iROV respondent server. To this end, we propose augmenting the RTR protocol with a new PDU type; i.e., the iROV query/response PDU. Although the prototype imple-

mentation assigns distinct PDU type values for queries and responses (12 and 13, respectively), in practice, they share an identical PDU structure: prefix, prefix length, origin ASN, and a validity flag, along with a variable-length field containing the iROV response. This is because all fields, with the exception of the validity flag, are common to both message types.

RTR iROV PDU:

```
{ prefix, length, asn, flags, iROV_Response }
```

An *iROV response* consists of a payload and an RSC object. The payload includes the query contents (prefix, origin ASN), a timestamp, and an indicator to signal the query's validity (i.e., `1` for "Yes" and `0` for "No"). The RSC object carries the hash of the payload, and is signed with an RPKI End-Entity certificate [62], identical to other RPKI signed objects such as ROA. An AS that receives an iROV response can therefore first validate the RSC object, ensure the payload hash's authenticity, then use the hash to ensure the payload's authenticity.

iROV Response:

```
{ Payload [prefix, origin, timestamp, validity],
  RSC Object
}
```

An RSC Object that carries one file hash, as created with the APNIC RSC Demo [4], has a size of 1519 bytes. Assuming that the payload consists of 1 byte validity flag, 4 byte origin ASN, 8 byte timestamp, 16 byte prefix (for IPv6) and 1 byte prefix length, the payload is 30 bytes in total. Thus, an iROV response comes to 1549 bytes in total size (1519 bytes for the public key, 30 bytes for the payload).

Given that the maximum BGP message size is 4096 bytes [55], and that observed average and maximum BGP message sizes are below 100 and 400 bytes respectively [19], BGP updates incorporating iROV responses remain within protocol limits. Furthermore, the operational impact of message bloat is bounded by how often iROV actually fires, and the overhead is not systemic; it is conditional on ROV-invalid routes that also carry explicit owner authorization. ROV-invalid announcements are a small fraction of observed routing (about 0.6% in recent measurements) [39], and iROV activates for only a subset of those cases (authorized exceptions). Even if each authorized exception carries an extra 1.5 kilobyte attribute, the aggregate control-plane footprint should remain small compared to baseline BGP churn.

5 Security Analysis

We provide an informal security analysis of iROV. We first focus on two threats that are directly introduced by interactivity. First, we show that iROV preserves the same origin-integrity guarantees as standard ROV against prefix hijacking (§5.1). Second, we argue that iROV's design mitigates the availability risk of new attack vectors, specifically query flooding, via probabilistic querying and validity stapling (§5.2). Then, later, we discuss a few other security risks under iROV deployment in §5.3.

5.1 Prevention of Prefix Hijacks

We claim that iROV does not create a new opportunity for adversaries to bypass ROV. Intuitively, iROV only relaxes the treatment of ROV-invalid BGP updates when the *legitimate prefix owner* explicitly approves them, and such approval is conveyed via a verifiable signature using RSC. Thus, unless the adversary can obtain the owner's signing key (which is outside our threat model), iROV cannot turn an otherwise-blocked hijack into an accepted route.

Claim. iROV does not expand the set of ROA-registered prefixes that a malicious AS can successfully hijack compared to standard ROV.

Setup. Let $\mathcal{P}$ be the set of prefixes that are ROA-registered in the RPKI. Prefixes outside $\mathcal{P}$ are treated identically by ROV and iROV, so we exclude them from this difference analysis. Let $\mathcal{H}_{\text{ROV}} \subseteq \mathcal{P}$ denote the set of prefixes that a certain adversary X can hijack under standard ROV.[1] Let $\mathcal{H}_{iROV} \subseteq \mathcal{P}$ denote the set of prefixes that the same adversary X can hijack in the network where all ROV ASes adopt iROV. Our goal is to show $\mathcal{H}_{iROV} \subseteq \mathcal{H}_{\text{ROV}}$.

Proof by contradiction. Assume there exists a prefix p such that $p \notin \mathcal{H}_{\text{ROV}}$ but $p \in \mathcal{H}_{iROV}$. The assumption $p \in \mathcal{H}_{iROV}$ means that some iROV-validating AS accepts a BGP update for p originated by the adversary. Under iROV, an ROV-invalid BGP update can be accepted only if the validating AS receives a *positive* owner authorization that is cryptographically verifiable. Because the adversary does not possess the resource owner's private key, it cannot forge such a valid iROV response. Therefore, the only remaining way for the adversary's BGP update to be accepted is that it is already ROV-valid, which implies that the adversary is authorized by a valid ROA for p. In that case, the same BGP update would also be accepted under standard ROV, contradicting $p \notin \mathcal{H}_{\text{ROV}}$. Hence, no such p exists, and we conclude $\mathcal{H}_{iROV} \subseteq \mathcal{H}_{\text{ROV}}$.

5.2 Resilience Against Query Flooding

We next analyze iROV's resilience against *query flooding*, an availability attack enabled by the introduced interactivity. The adversary controls one or more BGP-speaking routers (or ASes) and originates ROV-invalid updates for prefixes owned by a target AS that runs an iROV respondent server. The adversary's goal is to induce many iROV-validating ASes to query the target AS around the same time.

[1] We consider that the ROV adoption (at an arbitrary value) and the AS topology are fixed in this analysis.

Without any mitigation, a worst-case upper bound is that every iROV-adopting AS that observes the invalid BGP update queries the owner. Using the CAIDA AS-relationship dataset [10] (78,667 ASes as of January 2026) and assuming about 30% ROV adoption [38, 39], the number of plausible early iROV adopters is on the order of 23K ASes.

In expectation, probabilistic querying with probability p reduces the number of owner queries per BGP update from 23K to $p \times 23K$. For example, if $p = 0.1$ is applied uniformly across all iROV ASes, the above worst-case bound drops to 2.3K queries per invalid BGP update.

Once any upstream iROV AS obtains a signed authorization and staples it, downstream iROV ASes validate locally and do not re-query the owner. Therefore, the realized query load is typically lower than the 23K bound, and it is further reduced when higher-degree ASes adopt iROV more aggressively and use higher p than the edge. For example, if all Tier-1 and large Tier-2 ASes (e.g., with large customer cones) adopt iROV with $p = 1$ while smaller ASes use $p = 0.1$, the number of queries per invalid BGP update drops to a few thousand or even lower. The exact reduction depends on the adoption pattern and BGP update propagation paths, which we evaluate empirically in §7.3. Furthermore, we demonstrate that this differential interaction probability retains the increase in reachability in §7.2.

Moreover, adversaries may attempt to subvert the probabilistic querying and validity stapling measures to overwhelm a target AS by *repeatedly* inducing multiple waves of queries for the victim. For instance, an attacker can trivially craft numerous ROV-invalid BGP updates by iterating through all subprefixes of the target AS network. This exploit is easily mitigated by implementing a *per-owner AS rate limiting* on the iROV-validating ASes' side. A validating AS can keep track of how many queries have been sent recently to individual resource owner ASes. If too many queries were issued for a particular AS in a short period of time, the validating AS can stop inquiring the resource owner AS until some predetermined time has passed, discarding any relevant BGP updates in the meantime.

Last, when the resource owner AS is under a serious DoS attack, it may not be able to provide timely responses to iROV queries. For such extreme scenarios, iROV allows resource owners to deliver the iROV responses to the networks that would issue BGP updates on their behalf (e.g., DDoS scrubbing centers) via backup channels (e.g., email, web portal). Consequently, scrubbing centers can announce a BGP update with an attached iROV response in the first place, thus achieving high reachability without requiring real-time interaction with the resource owner AS.

5.3 Discussion on Other Deployment Risks

In addition to the two main threats, we list a few other potential security concerns one might have about iROV's deployment and discuss how iROV's design mitigates them.

- *Leakage of business relationships.* iROV queries act as a telemetry that transmit information about one AS's routing incident to another AS; thus, one may be concerned about potential leakage of business relationships between ASes. However, they contain only the absolute minimum data required: prefix, length, and origin ASN. Sensitive business relationships are never transmitted. Furthermore, validity stapling enables downstream ASes to validate without querying, minimizing redundant queries and preventing topology inference.
- *Revocation of iROV authorization.* iROV authorization is revoked by expiration rather than by an explicit active revocation protocol. As explained in §4.4, an iROV response carries an issuance timestamp and is valid only within a short configurable time window. This is suitable because iROV targets temporary emergency exceptions rather than long-lived policy state. In that sense, iROV is closer to a short-lived authorization model, analogous in spirit to short-lived certificates in TLS [66], where keeping the lifetime short reduces reliance on heavyweight revocation mechanisms.
- *Human errors in iROV server record.* iROV requires that the server location record (e.g., URL for an AS's respondent server) be registered manually, which may introduce human errors. While iROV does not eliminate the possibility for human errors in its record registration, it is designed to fail closed rather than fail open. If the server is unreachable or returns an invalid response, the validating AS falls back to standard ROV behavior. Even if the record incorrectly directs queries to an malicious respondent server masquerading as the real owner, the server cannot return a valid response unless it obtains the owner's private key. Thus, a mistakenly-issued, bad iROV record does not weaken the protection of RPKI; it only prevents the temporary exception from being used. Such human errors can also be reduced through internal testing before deployment.

6 Implementation

6.1 Operation Modes

There are two modes of operation for an iROV-validating AS that we evaluate: *initial-query* and *read-stapled* modes.

- *Initial-query mode.* This mode is triggered when the router receives a BGP update of an ROV invalid route that does not have an iROV response stapled to it. Upon receipt of the BGP update, it creates and sends an iROV query PDU via RTR to the RP. This query contains the prefix, prefix length, and origin ASN. The RP then sends an out-of-band query to a web server, which returns the response. The RP validates and relays this data back to the router.

- *Read-stapled mode.* This mode emulates the situation where the router receives a BGP update with an iROV response stapled to it. Upon receipt of the BGP update, the router loads the iROV query PDU with an iROV response. The router sends this PDU to the RP, requesting cryptographic validation. The RP returns the validation result to the router.

6.2 Implementation Overview

We modify the BIRD Routing Daemon [17] and the FORT Validator [34] to serve as the router and RP component of the iROV system, respectively. For the respondent, we create a simple HTTP server with Python Flask, which cryptographically signs and returns an iROV response.

- *BIRD.* Upon receipt of an invalid route, an entry point added inside BIRD's ROV function creates an iROV query request and puts it into a queue. This queue is routinely checked by an iROV inquiry procedure implemented in BIRD's RPKI hook. If the procedure sees that the queue is not empty, it dequeues one request and checks it against previous responses saved in cache. If there is a match, it discards the request and attempts to dequeue another item; otherwise, an iROV query RTR PDU is created and sent to the RP. A receipt handler is also added in the RPKI hook, which will extract the contents of an iROV response RTR PDU from the RP then save it to cache. The added source lines of code (LoC) in BIRD (version 3.1.2 [18]) is 646 lines, showing that iROV can be integrated with lightweight changes to existing BGP functionality in routers.
- *FORT.* We extend FORT to support the iROV's interactive operations with the router and the respondent. We also implement the cryptographic validation of iROV responses, namely, the RSC object component. As native support for RSC is not yet widespread among routing infrastructure projects, we implement a system that performs two passes of signature validation on SHA256 with RSA-2048, accurately emulating the processing overhead of RSC object validation [29, 62]. Likewise with BIRD, support for the iROV RTR PDUs have been added to FORT. This includes the implementation of the PDU sending and receiving handlers, as well as the cryptographic validation of iROV response. To handle the "initial-query" mode, HTTP request and fetching have also been implemented using the CURL library. This adds 392 lines of code to FORT (version 1.6.6 [35]).

The compact codebase modifications in both BIRD and FORT demonstrate that iROV requires only lightweight instrumentation of existing infrastructure. In addition to the validation of the iROV's core functionalities between the three components, we also measure the performance (i.e., processing, and memory) overhead introduced by iROV to both the router and the RP. Details for the performance evaluation setup are given in Section 7.1.2, with results in Section 7.6.

6.3 Engineering Decisions

- **Cache:** The implementation of a cache, especially on the router side, is critical to the throughput of iROV. Real-life BGP traces contain many groups, where dozens of BGP updates with the same prefix and origin AS are recorded in rapid succession. Without a router-side cache, the RTR query cadence would have to be increased significantly to cope with frequent duplicate queries in a production environment.
- **Async:** To ensure the system remains lightweight and non-intrusive to the router's route update processing, iROV operates as a fully asynchronous add-on. That is, all iROV-related events taking place on the router (ROV, RTR) do not block one another. This design guarantees that the router's main execution thread remains unblocked, preserving uninterrupted route processing regardless of external query latency.
- **Variable polling:** The throughput of BGP updates fluctuates significantly, which means there cannot be a single optimal interval value when it comes to polling the iROV inquiry queue on the router. To maximize efficiency under idle conditions while preventing pileups during surges, polling frequency was implemented to be dynamic based on the number of items remaining on the queue. This adaptive mechanism contributes to the negligible CPU overhead observed in our evaluation.

7 Evaluation

We evaluate iROV across three complementary dimensions that mirror its core design goals: network reachability, operator-facing benefits, and system performance. We first outline our experimental methodology and datasets in §7.1: §7.1.1 details the simulation setup for §7.2–§7.5, and §7.1.2 explains the prototype testbed setup used in §7.6.

Then we describe the findings that support our design goals (note that **G1** is addressed analytically in §5.1). The simulation results in §7.2 and §7.3 confirm that iROV restores reachability for authorized routes (**G2**) while maintaining manageable query overhead (**G3**). §7.4 demonstrates the system's ability to provide timely alerts (**G4**) while §7.5 shows accurate damage estimation (**G5**). Our prototype benchmarks in §7.6 verify that iROV imposes negligible overhead on routers and RPs (**G6**). Finally, we present a quantitative cost-benefit analysis in §7.7 to contextualize the practical implications of our findings.

7.1 Methodology and Datasets

7.1.1 Simulation Setup

We conduct empirical studies using our simulation with the CAIDA AS Relationship dataset [10]. The BGP update prop-

agation is simulated to follow the commonly used valley-free principle; that is, an AS only propagates routes learned from its customers to all its neighbors, while routes learned from providers or peers are only propagated to its customers. Also, an AS has a simple local preference policy that prefers routes learned from customers over peers, and peers over providers. In case of multiple routes with the same local preference, the AS selects the route with the shortest AS path length. A tie in AS path length is broken by selecting the route with the lowest next-hop AS number. When a legacy (i.e., no ROV) AS receives an update with a stapled iROV response, it accepts and transitively forwards the iROV response attribute to the next hop with a 98% probability to reflect the real-world behavior [24, 48].

Our AS tier classification within the database is as follows: Tier-1 ASes (27), which include clique ASes and a few large ISPs (where customer cone size $>$ 4,000 according to CAIDA ASRank [9]); Tier-3 ASes (41K), with no customers and only providers; and the rest Tier-2 ASes (37K).

Based on this classification, we set different iROV and ROV adoption rates for each tier of ASes. Also, we assign different iROV interaction probabilities for each tier of ASes to reflect the varying likelihood of interaction with iROV adopters in real-world settings.

7.1.2 Prototype Setup

We create a testbed for the iROV prototype performance evaluation on a host running Linux 6.17.12 with an AMD Ryzen 9 9950X (SMT enabled, 12 cores disabled for a total of 4 cores, 8 threads; operating frequency unmodified). We use the Linux network namespace (`ip netns`) feature to create virtual ASes within this host to simulate an eBGP environment. We use ExaBGP (version 4.2.25 [51]) in one of the virtual ASes to replay real-life BGP update archives while respecting original BGP update timing. The archives were retrieved from a RouteViews collector (`route-views2.oregon-ix.net`) [5, 6], which were recorded from 2nd to 6th of December, 2025.

ExaBGP peers with two ASes: a "baseline" AS running unmodified BIRD and FORT software, and an iROV-validating AS that runs the modified counterparts that we implemented. By realistically replaying a real-world BGP trace and announcing the BGP updates to both baseline and iROV ASes simultaneously, we obtain a highly accurate comparison between the two routers in the testbed. This result is presented in §7.6.

7.2 Authoritative Propagation Reachability

We first evaluate the ability of iROV to restore reachability for ROV-invalid yet owner-authorized BGP updates. This scenario is particularly critical for operations such as emergency DoS mitigation, where a resource owner may authorize a route that contradicts existing ROAs.

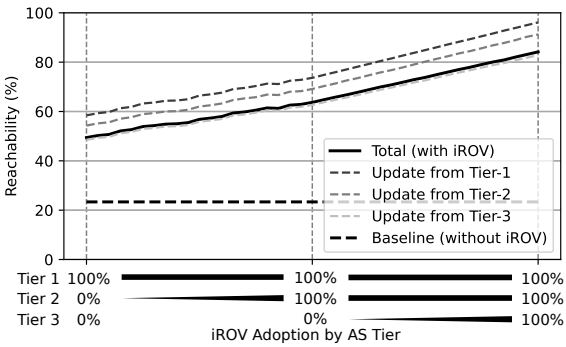

Figure 4: Restoration of global reachability for owner-authorized invalid routes under varying iROV adoption rates by AS tier. The adoption rate of iROV linearly increases (as denoted by the increasing thickness of the black bar) from 0% to 100% across the x-axis, first for Tier-2 ASes and then Tier-3 ASes. The iROV adoption rate for Tier-1 ASes are fixed to 100%.

We define the "Baseline" reachability as the percentage of ASes that receive and accept such routes when iROV is not deployed. In this experiment, we assume that 100% of Tier-1 ASes and 30% of Tier-2 and Tier-3 ASes have adopted standard ROV, roughly reflecting the current state of RPKI deployment [38]. We fix the iROV interaction probability at 100% for Tier-1 ASes, while setting it to 10% for Tier-2 and Tier-3 ASes to minimize overhead (see further discussion on this later in this section). We then vary the adoption rate of iROV (among ROV-enforcing ASes) by sweeping from 0% to 100% for Tier-2 and Tier-3 ASes. Note that we maintain 100% iROV adoption for Tier-1 ASes throughout, representing a deployment strategy that prioritizes core networks.

The x-axis of Figure 4 depicts these varying adoption scenarios across the three AS tiers. The y-axis denotes reachability, measured as the percentage of ASes in the Internet that accept the benign invalid route. The simulation is conducted across all possible resource owner ASes in the AS-level topology. Figure 4 highlights several key findings. First, the Baseline reachability is poor, resulting in slightly above 20%. This implies that if a benign invalid is announced for emergency DoS mitigation, only about 20% of attack traffic would be diverted to the scrubbing service, while the remaining 80% would continue to impact the target, thus failing at DoS mitigation. Second, we observe an immediate, substantial increase in reachability, more than doubling the baseline, solely from the adoption of iROV by Tier-1 ASes. This underscores the critical role of core ASes in the iROV ecosystem. As Tier-2 ASes adopt iROV, the majority of the Internet gains reachability to the invalid prefix. When Tier-3 ASes subsequently adopt the protocol, reachability extends to nearly all ASes. Given that 100% reachability is not strictly required for effective urgent routing changes, these results demonstrate that

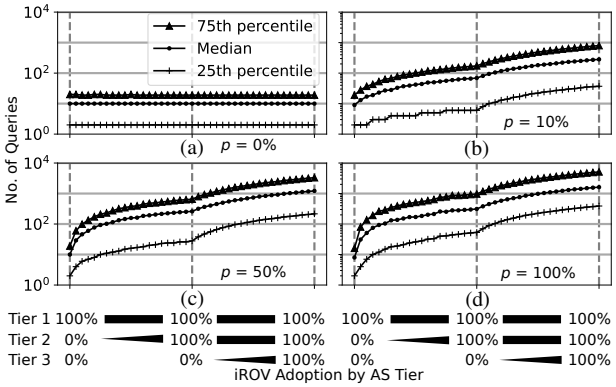

Figure 5: Total query overhead imposed on the resource owner per ROV-invalid route propagation, under varying interaction probabilities (p) for Tier-2 and Tier-3 ASes. X-axis indication is identical to Figure 4.

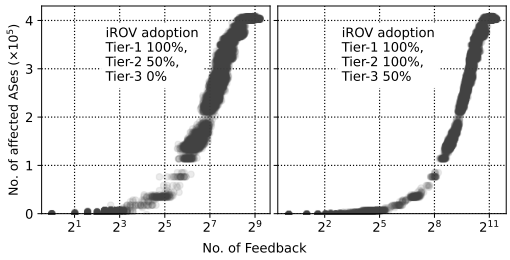

Figure 6: Correlation between the volume of iROV queries received by the legitimate resource owner and the total number of ASes affected by the invalid route.

partial deployment yields sufficient operational benefits.

7.3 Query Overhead for Resource Owners

Building on the reachability evaluations, we next evaluate whether iROV can sufficiently suppress the volume of queries sent to the legitimate resource owner. This metric is vital for determining whether the interactive mechanism introduces a vector for query-flooding DoS attacks. Using the same methodology for sweeping iROV adoption rates, we measure query overhead in terms of the total number of queries sent to the resource owner AS per invalid BGP update. In this evaluation, we vary the interaction probability (0%, 10%, 50%, 100%) for Tier-2 and Tier-3 ASes while maintaining full interaction (100%) for Tier-1 ASes, assuming full contribution from the core is feasible in the early stages.

The results are presented in Figure 5, revealing several important trends. First, Figure 5(a) depicts the Tier-1-only scenario, where overhead remains negligible because the relatively small number of Tier-1 ASes generates very few queries. Figure 5(b) illustrates the low probability setting ($p = 10\%$) for both Tier-2 and Tier-3 ASes. The results are highly favorable, with the query count remaining below 1,000 in most cases. Notably, the 1,000 queries will not be issued instantaneously by every validating AS, but rather issued throughout multiple seconds as the BGP update propagates through the Internet. Assuming an average propagation delay of 12 seconds [22] between the announcing AS and validating ASes, a respondent server needs to handle a load of approximately 83 queries per second, which is trivial for a modern web server [64]. In Figure 5(c) and (d), as we increase the interaction probability to 50% and 100% respectively, the query overhead increases linearly.

Considering the overall data from Figure 4 and Figure 5, we conclude that setting $p = 10\%$ for Tier-2 and Tier-3 ASes, combined with $p = 100\%$ for Tier-1 ASes, represents a nice balance between reachability and query overhead. Consequently, it serves as a viable sweet spot for integrating iROV into the current ROV deployment landscape.

7.4 Time-to-First Feedback

A critical operational requirement for iROV is providing timely feedback to resource owners regarding benign mistakes. We measure the propagation distance, in AS hops, that a benign invalid BGP update travels before triggering the first iROV query. We utilize the same simulation parameters as the overhead analysis in §7.2; that is, 100% adoption with always-on query for Tier-1 ASes and a conservative 10% interaction probability for Tier-2 and Tier-3 ASes for varying adoption rates.

Our empirical analysis demonstrates that iROV ensures rapid feedback even under partial deployment and probabilistic querying. Across all adoption ranges, the distance to the nearest querying node is at most 5 hops. The distribution is heavily skewed toward proximity; that is, the 99th percentile is 3 hops, while both the 75th percentile and the median are 2 hops. This indicates that in the majority of cases, a misconfiguration is flagged by a neighbor or a near-neighbor within minutes of announcement even under conservative settings (e.g., a 30-second per-hop delay following the suggested default MRAI timer value [55]).

7.5 Accurate Estimation of Damage

Beyond simple alerting, operators require actionable intelligence to prioritize their response to misconfigurations or possible attacks. A feedback mechanism that generates alerts without indicating severity risks being treated as operational noise. We evaluate whether the volume of iROV queries received by a resource owner can serve as a reliable side-channel for estimating the scope of the routing impact.

We posit that the aggregate number of iROV queries correlates with the number of networks that have accepted the invalid route. To test this, we analyzed the relationship be-

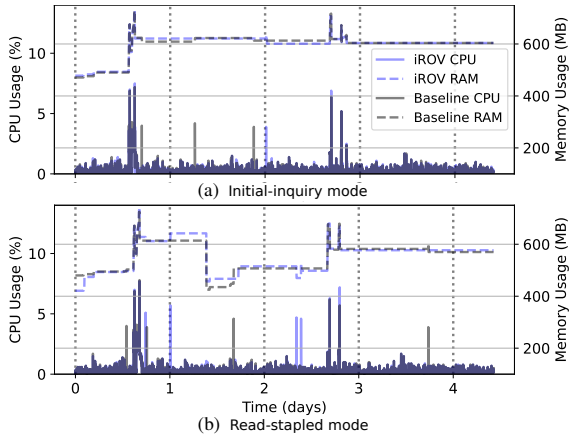

Figure 7: CPU and memory usage comparison between the iROV prototype and the baseline BIRD/FORT implementation during real-world BGP trace replay.

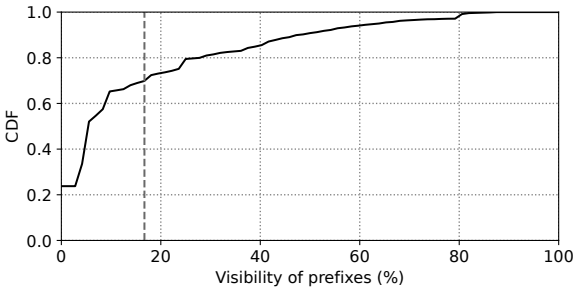

Figure 8: Visibility of invalid BGP updates as measured using RouteViews and RIPE RIS collectors. Average visibility, as indicated by the vertical dotted line, is 16.7%.

tween the volume of feedback and the number of "polluted" legacy ASes (non-validating ASes that accepted the invalid route) in our simulation. Once again, we fix the interaction probability of tier-1 ASes to 100% and others to 10%.

Figure 6 reveals a distinct positive correlation between the number of feedback queries and the number of affected legacy BGP ASes. As the invalid route propagates further, reaching more legacy networks, it encounters more iROV-enabled nodes, resulting in a higher query volume. Crucially, this correlation remains robust even at modest adoption levels; for instance, when 50% of Tier-2 ASes adopt iROV, the feedback count still accurately reflects the scale of the propagation.

This characteristic empowers network administrators to use iROV feedback as a quantitative risk analysis tool. Rather than treating all alerts equally, an administrator can distinguish between a contained configuration error (low query volume) and a widespread routing leak (high query volume). This visibility allows operators to make informed decisions regarding the urgency of remediation, effectively prioritizing their operational efforts based on the estimated damage.

7.6 Prototype Performance

We finally evaluate the system performance to verify that iROV incurs modest overhead on critical infrastructure. Using the testbed described in §7.1.2, we replayed real-world BGP traces to measure the CPU and memory consumption of both the router (BIRD) and the RP (FORT) under two distinct operation modes: *initial-query* (where the router must trigger new verifications) and *read-stapled* (where the router verifies attached proofs).

Figure 7 illustrates the resource usage over the duration of the experiment. Our results demonstrate that the overhead introduced by iROV is negligible for the BGP router. This efficiency stems from our asynchronous design choices detailed in §6; namely, by offloading the cryptographic verification and web queries to the RP, the router's main thread remains unblocked, preserving high-speed packet processing capabilities. Consequently, we observed no significant deviation in CPU utilization between the baseline BIRD router and the iROV-enabled router, even during bursts of invalid updates.

The overhead on the RP is similarly low. Throughout the experiment, the RP CPU usage averaged at approximately 3.7%, peaking only momentarily during signature verification batch processing. Memory usage for the RP remained stable between 600 MB and 900 MB. Collectively, these results confirm that iROV can be deployed on existing hardware without degrading the performance of the control plane.

7.7 Quantitative Cost-Benefit Analysis

While the effectiveness evaluations (§7.2–§7.5) demonstrate the clear benefits of iROV, they rely on simulation results, rather than real-world measurements, due to the difficulty of deploying iROV in the wild for testing purposes. To complement the simulation-based evaluations, we present our best-effort quantitative cost-benefit analysis based on real-world measurement data.

First of all, iROV incurs minimal cost to adopting ASes. The respondent server is expected to handle up to approximately 83 queries per second for an invalid BGP update as analyzed in §7.3. Even assuming a high intensity scenario of 100 distinct invalid BGP updates being announced every second, the number of queries could only amount to 8,300 per second. This rate of requests can readily be handled by one core of a commodity CPU [30][2], and by extension, modern entry-grade servers with an MSRP of under $2K [3]. Furthermore, respondent servers are simple web servers that can be incorporated

[2]Although creating an iROV response incurs cryptographic processing overhead on the server initially, the overhead is amortized by the fact that the response can be cached for further identical queries. Therefore, we consider the workload of a respondent server to be comparable to that of plain HTTP request handling.

into existing hardware such as HTTP-enabled RPs; it does not need dedicated, separate hardware. We further discuss the remaining, small yet non-negligible operational costs of running iROV in §8.

For quantitative analysis of the benefits, we first conduct a real-world measurement of how ROV-invalid BGP updates propagate across the Internet, which is the main problem that iROV aims to address. Although this does not directly measure the improvement of reachability through iROV (as we show with simulations in §7.2), it clearly measures the current, significantly limited route reachability (or visibility) due to the lack of iROV deployment in the real world. We aggregate the RIB dumps for May 1st, 2026 from all RouteViews and RIPE RIS BGP collectors [2, 6], then generate a database of ROV-invalid prefixes observed in the dumps using the Python ROV package [1]. The database consists of tuples containing the prefix, prefix length, and originating AS. This database is then compared once more with the RIBs, and each tuples' visibility is measured by dividing the number of RIBs that contain the tuple by the number of all RIBs. The result, as depicted in Figure 8, shows that ROV-invalid prefixes have a low visibility across the Internet. On average, ROV-invalid BGP updates are only visible in 16.7% of global public BGP collectors, roughly in line with our simulation results for baseline as indicated in Figure 4. This confirms that the current, low reachability of ROV-invalid BGP updates can be significantly be improved via iROV.

In addition, there are a few independent measurement study results that quantitatively demonstrate the severity of the limited route reachability that iROV can significantly improve. For example, previous studies show that 82% of hijack incidents are likely misclassification of benign invalids as hijacks [40]. Furthermore, about half of DDoS scrubber announcements lack valid ROAs [32].

8 Discussion

In this section, we discuss several important aspects of iROV regarding its large-scale deployment and operation.

iROV and alternative approaches. For the various utilities that iROV provides, some possible alternative methods exist that are worth noting. We briefly discuss them and highlight iROV's advantage and necessity.

- *Preregistration:* For urgent authorization of routes (e.g., DDoS scrubbers), one could argue that preregistration, along with reducing the delay for ROA propagation, could render iROV unnecessary. However, preregistration is only applicable to services that operators foresee using. This method is inherently fragile in that, by definition, any unexpected maneuvers cannot be covered. This fragility can only be prevented with our paradigm shift to real-time, on-demand authorization.
- *Speeding up ROA propagation:* In order for urgent authorizations to propagate swiftly on-demand, without introducing iROV, the delay present in the propagation of ROAs must be reduced. However, this is not trivial. For instance, the ecosystem does not define a specific value for RP refresh intervals. The default frequency value, which operators do not change often [33], differs by implementation. Furthermore, the ROA issuance interval and setup is different by RIRs [20]. To remove this discord in data propagation timing, and thus speed up ROA propagation, would require colossal changes to RPKI across the board.

iROV need not always compete with existing methods; regarding RPKI visibility, iROV works well to complement alternatives:

- *Troubleshooting workflows:* Access networks may already be employing troubleshooting workflows for control plane anomalies and incidents (including those related to RPKI), involving manual inspection to diagnose and rectify issues. However, this workflow is hindered by the fact that RPKI does not directly provide any insight in case of errors. By providing visibility into anomalies in the form of automated inter-AS queries, iROV can assist said workflows.
- *Detection systems:* Similarly, iROV can complement existing inference-based BGP hijacking detection systems, which may misclassify up to 82% of benign misconfigurations as attacks [40]. iROV removes false positives through explicit owner authorization. Furthermore, public route monitors [50, 57] have blind spots that sophisticated hijacks can exploit [7, 41, 47, 59]. iROV operates in-band across the routing infrastructure, providing direct, automated incident reporting without these blind spots.

Transitive attributes. iROV is designed to operate over the existing BGP infrastructure without protocol modification. We leverage the optional transitive BGP path attribute to "staple" and propagate iROV responses. Legacy routers that do not support ROV or iROV are required by the BGP specification to forward these attributes transparently [55]. Previous studies confirm that widespread adherence to this specification exists, with less than 2% of routers discarding unrecognized attributes [24, 48]. As shown in Figure 4, even accounting for this deviation from the standard, iROV significantly restores reachability for authorized updates compared to the baseline.

Incentives for deployment. The operators of legitimate resource owner ASes have three compelling reasons to deploy iROV. First, it provides *timely feedback* on benign misconfigurations. Our simulations show that even with low adoption, an AS typically receives the first iROV inquiry within 2–3 hops,

allowing for rapid error correction. Second, it offers *granular attack visibility*. Not only does iROV provide direct feedback to the resource owner, but it also enables operators to assess the impact of malicious route propagation. This allows operators to perform risk analysis, distinguishing between minor leaks and widespread hijacking attempts without relying on third-party monitoring. Third, it enables *urgent operational agility*. iROV facilitates the temporary authorization of routing deviations, such as redirecting traffic to a DDoS scrubbing center. Unlike standard ROV, which might rigidly block such protective maneuvers due to ROA mismatches, iROV allows the owner to explicitly and immediately authorize the deviation.

Validating ASes (e.g., transit networks) may also have direct incentives to adopt iROV. For instance, by permitting owner-approved routes through iROV, transit ASes can potentially serve more traffic and increase their revenue. While these transit-side incentives are arguably less direct than those for resource owners, it should be noted that even the successful adoption of standard ROV was not primarily driven by direct incentives for transit ASes; rather, it was achieved through the coordinated development of Best Current Practices [8, 15, 21], advocacy by network operator communities [44–46, 63], and policy initiatives [28, 54].

iROV configuration. We envision that the global operational parameters for iROV are to be determined prior to or during deployment through evaluation and community consensus. For example, the interaction probability p for each iROV-validating AS should be established in advance to ensure reliable global operation and utility; specifically, features such as the accurate estimation of affected ASes based on query volume (see §7.5) rely on a predictable probability distribution across the network hierarchy. In this paper, we provide a concrete example configuration: $p = 1.0$ for Tier-1 ASes and $p = 0.1$ for all other ASes. To promote harmonized community-driven deployment, we recommend that such parameter profiles be formalized within Best Current Practices (BCP) or industry frameworks like MANRS [21]. Finally, as a long-term goal following global deployment, we envision the development of a public monitoring platform capable of tracking iROV adoption rates and their configured interaction probabilities in the wild.

Limitations. First, we recognize that our simulation uses a simplified AS-level routing model based on standard valley-free propagation and local-preference assumptions, and therefore does not capture the full heterogeneity of real-life control plane policies. However, our intent is not to claim exact Internet-wide deployment outcomes, but to approximate iROV's efficacy using a widely used inter-AS routing abstraction. Second, iROV requires that an AS operate a respondent server, and our current design assumes that adopting ASes possess an RP inside its trusted internal network, which may not be the case for some ROV enforcing ASes (e.g., ASes using various trusted VRP providers [14, 52] in lieu of a local RP). However, as discussed in §7.7, the capability requirements for a respondent server is not high, and centralized VRP providers can migrate their current VRP information distribution semantics to iROV response processing (i.e., publishing validation results for any validity staples and ROV-invalid prefix detected real-time in the control plane).

Operational costs. iROV is designed to minimize overhead and deployment intrusiveness. Yet, some operational costs are inevitable for the operators of iROV-adopting ASes:

- **Respondent server maintenance:** While the capital cost of deploying an iROV respondent server is marginal, the operational cost of maintaining the server may not be negligible. An iROV respondent server should be operated and maintained, possibly with redundancy in mind for high availability. The location records of respondent servers also need to be kept up-to-date. Procedures to test server record correctness may also be implemented.

- **RSC signing workflow:** A valid RSC object must be created and signed for an iROV response. As such, a workflow to ensure secure management of related cryptographic assets (e.g., End-Entity certificates) may be necessary.

- **iROV feedback parsing:** While iROV provides visibility into RPKI, and helps ASes detect and rectify routing issues, manual effort is still needed to parse and act on this visibility (e.g., read iROV queries, determine relevance/severity, triage cause of issue). The operators may also need to create a workflow that responds to threat signals (e.g., prefix is being hijacked) or protocol abuse (e.g., respondent server is being swamped with queries).

Accurately quantifying these operational costs is difficult because they depend heavily on the specific operational environment and practices of each AS, and is therefore out of scope for this work.

9 Conclusion

iROV redefines the relationship between resource owners and validators from silent filtering to active collaboration. While we do not argue that interactivity is the only path to resolving the operational and security challenges of ROV, our work demonstrates that this lightweight and practical extension effectively addresses these long-standing challenges, offering a robust upgrade to this de facto global routing security mechanism. Crucially, iROV also shifts the burden of error detection from human operator vigilance to built-in automated protocol mechanisms.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-25404706, Development of Technologies and International Cooperation for Establishing a Trusted Global Ecosystem in 6G NTN).

A Ethical Considerations

Stakeholder analysis. We identify three stakeholders in regard to this research.

- Resource owner ASes: ASes which receive feedback from Validating ASes and issue route authorizations via iROV. These ASes will be able to better monitor and understand the extent of routing incidents that are relevant to them. The execution and publication of this research do not impact them negatively.
- Validating ASes: ASes which use iROV to possibly overrule the ROV invalid decision for a route, and may inquire pertinent resource owner ASes. These ASes will be able to accept routes that are ROV-invalid yet benign. This enables them to potentially select better routes and thereby receive higher revenue. The execution and publication of this research do not impact them negatively.
- Internet users: Individuals who transmit and receive data via either resource owner ASes, validating ASes, or both. Users will have a lower chance of experiencing performance or connectivity degradation incurred by stale or incorrect ROAs. The execution and publication of this research do not impact the users negatively.

Decision to execute and publish. We decided to execute and publish this study as it holds great potential to improve RPKI and ROV by enabling timely propagation of information and interactivity. This research, in its entire process including execution and publication, does not impact any stakeholders in a substantially negative way while having great potential benefits.

Ethics in execution of research. All evaluations were conducted via simulation or deployment in a closed testbed environment. All datasets used are public data we clearly cited in the paper and software used for the evaluation are open source. Therefore, we consider our work to have no ethical concerns.

B Open Science

Our research artifacts include five components that are used to evaluate the performance of our proposed protocol: The iROV-validating BGP router, Relying Party, HTTP server, MRT playback BGP router runner, and resource usage recorder. The artifacts also include our BGP announcement propagation simulation script and a CAIDA AS relationship dataset, along with ROV-invalid visibility measurement tools.

The iROV-validating BGP router is a modified version of the BIRD Routing Daemon. The Relying Party is a modified version of the FORT Validator. The HTTP server is a Python script that uses Python Flask and cryptography packages to serve iROV responses. The MRT playback BGP router runner consists of an ExaBGP configuration file, a Python script for ExaBGP helper process, and another Python script that runs and checks the status of the ExaBGP instance. Finally, the resource usage recorder is a Python script that uses the Python pidstat package to collect CPU and memory usage of processes.

These artifacts are available at:
https://github.com/NetSP-KAIST/iROV or
https://zenodo.org/records/20399386?token=eyJhbGciOiJIUzUxMiJ9.eyJpZCI6IjcwZmZhZDE3LTA5YTEtNDM3My04MTg2LThjNGY3NDM1YzA4ZCIsImRhdGEiOnt9LCJyYW5kb20iOiJhZjRlYzZhNjYxNWNkZjM2ZjM2NTk3OGE1NWE4OTBjMSJ9.VSFddW2eHCjVeE6f2ZfQypYMUUec_4zsmhPrlOn7rVt0LKAwoqrfYlg31sl5FC8Ns7nFRw3qY1DcDPzg4HmyOg

References

[1] Python route-origin-validator. https://github.com/InternetHealthReport/route-origin-validator/. Accessed: 2026-05-22.

[2] RIPE RIS Route Collectors. https://ris.ripe.net/docs/route-collectors/#peer-meta-data. Accessed: 2026-05-18.

[3] HPE ProLiant MicroServer Gen11 Retail. https://www.hp.com/us-en/shop/pdp/hpe-smart-choice-proliant-microserver-gen11-ultra-micro-tower-server-1-x-intel-pentium-gold-g7400-37-ghz-16-gb-ram-1-tb-h, 2024. Accessed: 2026-05-21.

[4] APNIC. rpki-rsc-demo. https://github.com/APNIC-net/rpki-rsc-demo. Accessed 2026-03-16.

[5] Network Setup Resource Center at the University of Oregon. route-views2 Collector Archive. https://archive.routeviews.org/route-views2/bgpdata/. Accessed 2026-01-27.

[6] Network Setup Resource Center at the University of Oregon. RouteViews. https://www.routeviews.org/routeviews/about/. Accessed 2026-01-27.

[7] Henry Birge-Lee, Maria Apostolaki, and Jennifer Rexford. Global BGP Attacks that Evade Route. In *Passive*

and Active Measurement: 26th International Conference, PAM 2025. Springer Nature, 2025.

[8] R. Bush. BCP 185: Origin Validation Operation Based on the Resource Public Key Infrastructure (RPKI), 2014.

[9] CAIDA. AS Rank. https://asrank.caida.org/. Accessed 2026-01-27.

[10] CAIDA. AS Relationships Dataset. https://www.caida.org/catalog/datasets/as-relationships/. Accessed 2026-01-27.

[11] Wenqi Chen, Zhiliang Wang, Dongqi Han, Chenxin Duan, Xia Yin, Jiahai Yang, and Xingang Shi. ROV-MI: Large-Scale, Accurate and Efficient Measurement of ROV Deployment. In *Network and Distributed System Security (NDSS) Symposium 2022*, 2022.

[12] Ying-Ju Chi, Ricardo Oliveira, and Lixia Zhang. Cyclops: The AS-level Connectivity Observatory. *ACM SIGCOMM Computer Communication Review*, 38(5):5–16, 2008.

[13] Taejoong Chung, Emile Aben, Tim Bruijnzeels, Balakrishnan Chandrasekaran, David Choffnes, Dave Levin, Bruce M Maggs, Alan Mislove, Roland van Rijswijk-Deij, John Rula, and Nick Sullivan. RPKI is Coming of Age: A Longitudinal Study of RPKI Deployment and Invalid Route Origins. In *Proceedings of the Internet Measurement Conference*, pages 406–419, 2019.

[14] CloudFlare. CloudFlare RPKI. https://rpki.cloudflare.com/. Accessed 2026-01-28.

[15] Communications Security, Reliability, and Interoperability Council at the Federal Communications Commission. Final Report – BGP Security Best Practices, March 2013. https://transition.fcc.gov/bureaus/pshs/advisory/csric3/CSRIC_III_WG4_Report_March_%202013.pdf. Accessed 2026-02-01.

[16] Corero. RPKI and Cloud DDoS Protection. https://www.corero.com/rpki-and-cloud-ddos-protection/. Accessed 2026-01-31.

[17] CZ.NIC. BIRD Internet Routing Daemon. https://bird.nic.cz/. Accessed 2026-01-27.

[18] CZ.NIC. Source code for BIRD 3.1.2. https://gitlab.nic.cz/labs/bird/-/tree/v3.1.2. Accessed 2026-01-27.

[19] IETF Datatracker. Decoupling BGPsec Documents and Extended Messages draft. https://datatracker.ietf.org/meeting/98/materials/slides-98-sidrops-decoupling-bgpsec-documents-and-extended-messages-draft-00. Accessed 2026-01-31.

[20] Romain Fontugne, Amreesh Phokeer, Cristel Pelsser, Kevin Vermeulen, and Randy Bush. RPKI Time-of-Flight: Tracking Delays in the Management, Control, and Data Planes. In *International Conference on Passive and Active Network Measurement*, pages 429–457. Springer, 2023.

[21] Mutually Agreed Norms for Routing Security. Actions. https://manrs.org/netops/actions/. Accessed 2026-02-01.

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

[23] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are We There Yet? On RPKI's Deployment and Security. *Cryptology ePrint Archive*, 2016.

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

[25] Tomas Hlavacek, Amir Herzberg, Haya Shulman, and Michael Waidner. Practical Experience: Methodologies for Measuring Route Origin Validation. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 634–641. IEEE, 2018.

[26] Tomas Hlavacek, Haya Shulman, Niklas Vogel, and Michael Waidner. Keep your friends close, but your routeservers closer: Insights into {RPKI} validation in the internet. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4841–4858, 2023.

[27] Tomas Hlavacek, Haya Shulman, and Michael Waidner. Smart RPKI Validation: Avoiding Errors and Preventing Hijacks. In *European Symposium on Research in Computer Security*, pages 509–530. Springer, 2022.

[28] The White House. National Cybersecurity Strategy Implementation Plan Version 2. https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/05/NCSIP-Version-2-FINAL-May-2024.pdf. Accessed 2026-02-01.

[29] G. Huston. RFC 7935: The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure, 2016.

[30] Keon Jang, Sangjin Han, Seungyeop Han, Sue Moon, and KyoungSoo Park. {SSLShader}: Cheap {SSL} acceleration with commodity processors. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*, 2011.

[31] Mattijs Jonker, Anna Sperotto, Roland van Rijswijk-Deij, Ramin Sadre, and Aiko Pras. Measuring the Adoption of DDoS Protection Services. In *Proceedings of the 2016 Internet Measurement Conference*, pages 279–285, 2016.

[32] Shyam Krishna Khadka, Suzan Bayhan, Ralph Holz, and Cristian EW Hesselman. Detecting and Characterizing DDoS Scrubbing from Global BGP Routing: Insights from Five Leading Scrubbers. In *Passive and Active Measurement Conference, PAM 2026*. Springer, 2025.

[33] John Kristoff, Randy Bush, Chris Kanich, George Michaelson, Amreesh Phokeer, Thomas C. Schmidt, and Matthias Wählisch. On measuring rpki relying parties. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, page 484–491, New York, NY, USA, 2020. Association for Computing Machinery.

[34] LACNIC and NIC.MX. FORT Validator. https://fortproject.net/en/validator. Accessed 2026-01-27.

[35] LACNIC and NIC.MX. Source code for FORT 1.6.6. https://github.com/NICMx/FORT-validator/releases/tag/1.6.6. Accessed 2026-01-27.

[36] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. PHAS: A Prefix Hijack Alert System. In *USENIX Security symposium*, 2006.

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

[38] W. Li and T. Chung. RoVista. https://rovista.netsecurelab.org/analytics. Accessed 2025-08-04.

[39] Weitong Li, Zhexiao Lin, Md Ishtiaq Ashiq, Emile Aben, Romain Fontugne, Amreesh Phokeer, and Taejoong Chung. RoVista: Measuring and Analyzing the Route Origin Validation (ROV) in RPKI. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 73–88, 2023.

[40] Weitong Li, Tao Wan, and Taejoong Chung. Demystifying RPKI-Invalid Prefixes: Hidden Causes and Security Risks. In *Network and Distributed System Security (NDSS) Symposium 2026*, 2026.

[41] Weitong Li, Yongzhe Xu, and Taejoong Chung. The Threat Landscape of IP Leasing in the RPKI Era. In *Proc. of IEEE Symposium on Security and Privacy*, 2026.

[42] Lumen. Lumen® DDoS Mitigation Services – Frequently asked questions about RPKI, December 2021. https://assets.lumen.com/is/content/Lumen/rpki-ddos-frequently-asked-questions. Accessed 2026-01-31.

[43] SecLists.org NANOG mailing list archive. It can be challenging to advise DDoS mitigation subscribers on their RPKI-ROA needs. https://seclists.org/nanog/2024/Oct/70. Accessed 2026-01-31.

[44] SecLists.org NANOG mailing list archive. Re: BCP38 For BGP Customers. https://seclists.org/nanog/2022/Nov/39. Accessed 2026-02-01.

[45] SecLists.org NANOG mailing list archive. Re: CloudFlare issues? https://seclists.org/nanog/2019/Jun/322. Accessed 2026-02-01.

[46] SecLists.org NANOG mailing list archive. RPKI's 2024 Year in Review. https://seclists.org/nanog/2025/Jan/126. Accessed 2026-02-01.

[47] Alexandros Milolidakis, Tobias Bühler, Kunyu Wang, Marco Chiesa, Laurent Vanbever, and Stefano Vissicchio. On the Effectiveness of BGP Hijackers That Evade Public Route Collectors. *IEEE Access*, 11:31092–31124, 2023.

[48] Cameron Morris, Amir Herzberg, Bing Wang, and Samuel Secondo. Bgp-iSEC: Improved Security of Internet Routing Against Post-ROV Attacks. In *Network and Distributed System Security (NDSS) Symposium 2024*. Internet Society, 2024.

[49] National Institute of Standards and Technology. NIST RPKI Monitor. https://rpki-monitor.antd.nist.gov/ROV. Accessed 2026-01-23.

[50] RIPE NCC. Routing Information Service Live (RIS Live). https://ris-live.ripe.net/. Accessed 2026-01-28.

[51] Exa Networks. ExaBGP. https://github.com/Exa-Networks/exabgp/releases/tag/4.2.25. Accessed 2026-01-27.

[52] NTT. NTT RPKI VRP. https://rpki.gin.ntt.net/api/export.json. Accessed 2026-01-28.

[53] NTT-GIN. BGPalerter. https://github.com/nttgin/BGPalerter. Accessed 2026-01-28.

[54] The White House Office of the National Cyber Director. Roadmap to Enhancing Internet Routing Security. https://bidenwhitehouse.archives.gov/wp-content/uploads/2024/09/Roadmap-to-Enhancing-Internet-Routing-Security.pdf. Accessed 2026-02-01.

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

[56] Nils Rodday, Ítalo Cunha, Randy Bush, Ethan Katz-Bassett, Gabi D Rodosek, Thomas C Schmidt, and Matthias Wählisch. Revisiting RPKI Route Origin Validation on the Data Plane. In *Proc. of Network Traffic Measurement and Analysis Conference (TMA), IFIP*, 2021.

[57] RouteViews. RouteViews FAQ: How do I access data in real-time? https://www.routeviews.org/routeviews/faq/#how-do-i-access-real-time-routeviews-data. Accessed 2026-01-28.

[58] Haya Schulmann and Shujie Zhao. Learning to Identify Conflicts in RPKI. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, pages 1490–1505, 2025.

[59] Haya Schulmann and Shujie Zhao. Stealth BGP Hijacks with uRPF Filtering. In *19th USENIX WOOT Conference on Offensive Technologies (WOOT 25)*, pages 129–138, 2025.

[60] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. ARTEMIS: Neutralizing BGP Hijacking Within a Minute. *IEEE/ACM transactions on networking*, 26(6):2471–2486, 2018.

[61] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu. Detecting Prefix Hijackings in the Internet with Argus. In *Proceedings of the 2012 Internet Measurement Conference*, pages 15–28, 2012.

[62] Job Snijders, Tom Harrison, and Ben Maddison. A Profile for RPKI Signed Checklists (RSCs). RFC 9323 (Proposed Standard), November 2022.

[63] Terry Sweetser. BCP 185 is a 'must do'. https://blog.apnic.net/2023/10/25/bcp-185-is-a-must-do/. Accessed 2026-02-01.

[64] TechEmpower. Round 23 results - techempower framework benchmarks. https://www.techempower.com/benchmarks/#section=data-r23&test=db, 2025. Accessed: 2026-02-05.

[65] Cecilia Testart, Philipp Richter, Alistair King, Alberto Dainotti, and David Clark. Profiling BGP Serial Hijackers: Capturing Persistent Misbehavior in the Global Routing Table. In *Proceedings of the Internet Measurement Conference*, pages 420–434, 2019.

[66] Emin Topalovic, Brennan Saeta, Lin-Shung Huang, Collin Jackson, and Dan Boneh. Towards short-lived certificates. Web, 2012.

[67] Wenjie Xu, Deliang Chang, and Xing Li. On the classification and false alarm of invalid prefixes in RPKI based BGP route origin validation. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 654–658. IEEE, 2019.

[68] Zheng Zhang, Ying Zhang, Y Charlie Hu, Z Morley Mao, and Randy Bush. iSPY: Detecting IP Prefix Hijacking On My Own. In *Proceedings of the ACM SIGCOMM 2008 conference on Data Communication*, pages 327–338, 2008.