

Understanding the Stealthy BGP Hijacking Risk in the ROV Era

Yihao Chen^{§†}, Qi Li^{‡†}, Ke Xu^{¶†}, Zhuotao Liu^{‡†}, Jianping Wu^{‡†},

[§]DCST & BNRist & State Key Laboratory of Internet Architecture, Tsinghua University

[‡]INSC & State Key Laboratory of Internet Architecture, Tsinghua University, [†]Zhongguancun Laboratory

[¶]DCST & State Key Laboratory of Internet Architecture, Tsinghua University

yh-chen21@mails.tsinghua.edu.cn, {qli01, xuke, zhuotaoliu}@tsinghua.edu.cn, jianping@cernet.edu.cn

Abstract—The partial deployment of Route Origin Validation (ROV) poses an unexpected security threat known as stealthy BGP hijacking, *i.e.*, a particularly elusive form of BGP hijacking where malicious routes divert traffic without reaching (and thus alerting) the victims. This risk remains largely unexplored, with neither documented real-world incidents nor systematic characterization available. To bridge this gap, we formalize stealthy BGP hijacking and propose heuristics to discover potential instances through routing table discrepancies. We conduct the first *empirical* study to track and profile stealthy BGP hijacking in the wild, contributing a curated real-world incident dataset and a long-term monitoring service. Inspired by the empirical insights, we further conduct an *analytical* study to exhaustively assess the risk. This requires accurate ROV deployment data, complete Internet-wide routes, and tailored analytical models. To address these challenges, we develop SHAMAN, a BGP route inference framework dedicated to assessing stealthy BGP hijacking risk. SHAMAN consolidates multiple sources to construct an accurate view of ROV deployment, infers complete Internet-wide routes through a highly efficient matrix-based approach, and facilitates statistical risk analysis via a “victim-target-hijacker” 3-tuple model. By reducing the time for generating Internet-scale routes from over three months to just 5.22 hours, SHAMAN enables systematic risk assessment across 8.3 billion generated routes under real-world ROV deployment. Our findings reveal a 14.1% overall success probability for stealthy BGP hijacking, with targeted attacks reaching 99.5% success in specific cases. Validation against our real-world dataset shows up to 95.9% incident-level accuracy, demonstrating the fidelity of our analytical results.

I. INTRODUCTION

The Border Gateway Protocol (BGP) has been known for its security vulnerabilities, with the infamous BGP hijacking being a major threat. To counter this threat, the community has proposed various security enhancements [1]–[7], among which the Resource Public Key Infrastructure (RPKI) and Route Origin Validation (ROV) show the promise in real-world deployment. ROV-enabled Autonomous Systems (ASes) can retrieve authorized registries managed by RPKI, known as Route Origin Authorizations (ROAs), based on which they can verify the correctness of prefix-origin associations contained in

received BGP announcements. As of March 2025, ROAs have covered 57.1% of globally routable IPv4 prefixes [8], yet the deployment of ROV is relatively limited, with only hundreds to thousands of ASes identified as ROV-enabled [9]–[11].

Presumably, ROV will remain in partial deployment for a relatively long time, which, in addition to offering incomplete protection as a consequence, results in an unexpected security threat, *i.e.*, *highly stealthy BGP hijacking that is effectively invisible from the victim on the control plane*. This new threat, which we refer to as ROV-related stealthy BGP hijacking, or for short, *stealthy hijacking*, occurs when an AS, despite being nominally protected by ROV-enabled ASes from receiving malicious routes, has its traffic silently diverted to a hijacker through legacy ASes along the data plane path. It is particularly insidious because the affected AS remains unaware of malicious routes throughout the hijacking, rendering common control-plane based protections ineffective in practice.

This highlights the unexpected downside of partial ROV deployment, yet the issue remains largely unexplored. No real-world stealthy hijacking incidents have been documented, and a systematic investigation into its prevalence and impacts is still missing. A recent study [12] takes a pioneering step towards mitigating stealthy hijacking via proactive rerouting and blackholing. Yet, its mitigation-oriented focus provides limited real-world evidence or heuristics for tracking and profiling the threat (see further discussions in §IX).

To bridge this gap, we seek insights from real-world observations. However, the lack of an established definition of stealthy hijacking makes it difficult to identify the threat. To this end, we formalize stealthy BGP hijacking and derive heuristics to discover hijacking instances based on routing table discrepancies observed across vantage points. Our rationale behind is to determine if any AS along legitimate routes can forward traffic to potential hijackers. Using routing tables from RouteViews vantage points, we conduct the first empirical study to track and profile stealthy hijacking in the wild (§IV). We capture 1,393 potential incidents over a two-month window in 2025, and analyze their impacts and causes extensively. We further validate these observations against a broad knowledge base including RPKI, IRR and WHOIS, which results in a curated dataset of 318 high-confidence incidents covering 2,178 routes. This dataset, along with our long-term monitoring service continuing to report real-world incidents, is

publicly available at <https://yhchen.cn/stealthy-bgp-hijacking>.

A key observation from our empirical study is that the visibility of stealthy hijacking is sensitive to vantage point selection and subject to continual change. This inspires us to further propose comprehensive and deterministic risk assessment through an analytical approach, which, however, faces three key challenges. First, determining the current state of ROV deployment is nontrivial. Unlike ROAs, ROV deployment is not publicly disclosed by default, and existing ROV measurements remain limited in coverage. Second, given the complexity of Internet topology, acquiring complete knowledge of Internet-wide routes to fully assess the risk is challenging. Third, no dedicated analytical model currently exists to characterize stealthy hijacking risk at a fine-grained level, yet such a model is essential for systematic risk assessment.

To address these challenges, we develop SHAMAN, a BGP route inference framework dedicated to assessing stealthy hijacking risk (§V). SHAMAN consolidates ROV measurements from multiple sources to ensure an accurate view of current ROV deployment. It utilizes a matrix-based approach to infer complete knowledge of legitimate AS-level routes and potential AS-to-hijacker routes under partial ROV deployment. This approach encodes essential routing information in compact matrices through a unique one-byte encoding scheme, and leverages highly optimized matrix operations for efficient route inference. Combined with a topology compression method, SHAMAN generates routes across *all* ASes within hours, reducing the otherwise months-long runtime of prior art [13]–[15]. It further applies a “victim-target-hijacker” three-tuple model to statistically characterize fine-grained hijacking instances, thereby supporting systematic role-based risk analysis.

Through SHAMAN, we conduct comprehensive analytical risk assessment with the current Internet topology, examining 8.3 billion Internet-wide routes across 77,600 ASes (§VI). Our analysis exhausts “victim-target-hijacker” instances that are vulnerable to stealthy hijacking, and characterizes their prevalence, distribution, and topological features, yielding seven key insights. We reveal that the current partial ROV deployment introduces a 14.1% overall success probability for stealthy hijacking, with targeted attacks reaching 99.5% success in specific cases. We also evaluate SHAMAN extensively (§VII). We show that it reduces the time for Internet-scale route generation from over three months to just 5.22 hours, achieving a 500-fold speedup over existing methods. Validation against our curated dataset demonstrates up to 95.9% incident-level accuracy, confirming the fidelity of our analytical results. Lastly, we conduct ablation experiments to justify SHAMAN’s design choice of consolidating multiple ROV measurement sources, and validate its robustness against input noise.

To summarize, our contributions are four-fold:

- We develop effective heuristics to discover stealthy BGP hijacking instances based on routing table discrepancies.
- We conduct the first empirical study to track stealthy BGP hijacking in the wild, establishing a curated real-world incident dataset and a long-term monitoring service.
- Motivated by empirical insights, we design SHAMAN, a

framework dedicated to systematic assessment of stealthy BGP hijacking risk, and evaluate it extensively.

- Through SHAMAN, we assess stealthy BGP hijacking risk in the current Internet thoroughly, deriving seven key insights while achieving 95.9% incident-level accuracy.

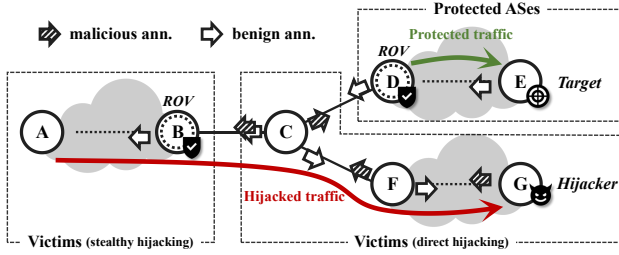
II. BACKGROUND

Interdomain Routing and BGP Hijacking. As of March 2025, the Internet comprises over 77,600 Autonomous Systems (ASes) in interdomain routing [16], each identified by a unique AS Number (ASN). These ASes interconnect via the Border Gateway Protocol (BGP), where each AS announces its IP prefixes to neighbors through route announcements. Upon receiving an announcement, an AS extracts the AS-level path destined for a specified prefix, updates its Routing Information Base (RIB), and performs best-route selection. The chosen route is then appended with the AS’s own ASN and further propagated to selected neighbors. Both best-route selection and propagation are influenced by the AS’s routing policies, *e.g.*, predefined route preferences based on business relationships.

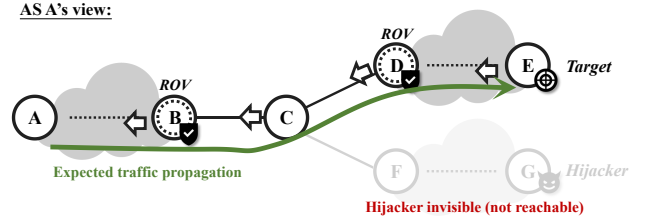
BGP lacks native security mechanisms and is vulnerable to misinformation injected by malicious actors. A consequent threat is BGP hijacking, where a malicious AS falsely claims to originate a prefix it does not own. If other ASes accept the forged announcement, they reroute traffic accordingly and divert it to the hijacker. BGP hijacking takes two forms: *exact-prefix hijacking*, where the hijacker announces the exact prefix owned by the legitimate origin, and *sub-prefix hijacking*, where a more specific sub-prefix is announced. The latter is typically more damaging, since BGP routers prioritize the longest-prefix match when forwarding traffic. Real-world BGP hijacking incidents have reported serious consequences, *e.g.*, redirecting cryptocurrency funds to attacker-controlled accounts [17].

RPKI and ROV for BGP Security. To counter BGP hijacking, the community proposed various security extensions [1], [2], [4]. Among them, Resource Public Key Infrastructure (RPKI) [6] and Route Origin Validation (ROV) [7] have gained significant real-world adoption. RPKI provides a cryptographic framework for prefix holders to publish Route Origin Authorizations (ROAs), which specify valid origin ASes and maximum allowable prefix lengths. Meanwhile, ROV refers to the operational practice by which ASes retrieve and verify these ROAs to validate BGP announcements, thereby ensuring that the announcing AS is authorized to advertise a specific prefix. Invalid BGP announcements are typically discarded.

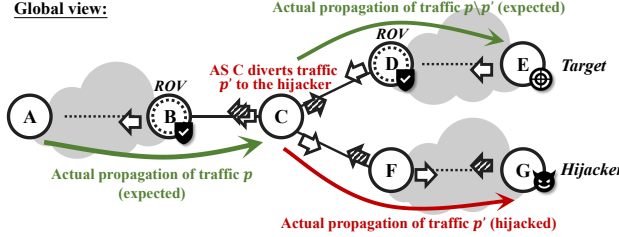
RPKI has gained great traction in recent years, but the actual state of ROV deployment remains nontrivial to know. Existing measurements of ROV deployment are mostly best-effort and limited in coverage, estimating anywhere from several hundred to over three thousand ROV-enabled ASes across the Internet [9]–[11], [18], [19]. This incomplete deployment greatly throttles ROV’s effectiveness, *e.g.*, even with 60% global ROV adoption, sub-prefix hijacking still succeeds 40% of the time [12]. More critically, partial ROV deployment introduces an unintended security risk, as we highlight next.



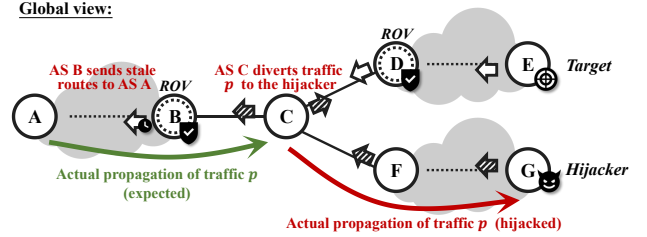
(a) Impact of partial ROV deployment on BGP hijacking.



(b) Expected traffic propagation from AS A's view.



(c) Actual traffic propagation under sub-prefix hijacking.



(d) Actual traffic propagation under exact-prefix hijacking.

Fig. 1: Stealthy hijacking under partial ROV deployment.

III. PROBLEM STATEMENT

In this section, we articulate the unintended security risk of partial ROV deployment and define the relevant concepts.

A. The Unexpected Downside of ROV

ROV is proposed to prevent BGP hijacking. However, in case of partial deployment, its effectiveness can be undermined. Figure 1 illustrates a scenario where BGP hijacking succeeds despite ROV deployment. We abstract the Internet topology as a graph, where vertices represent ASes and edges represent interdomain links. In this scenario, the *hijacker* (AS G) launches an attack by announcing (sub-)prefixes owned by the *target* (AS E). The ROV-enabled ASes block propagation of malicious announcements, so only AS C and F may accept the bogus route to the hijacker¹ depending on routing policies.

We take AS A as an example to explain BGP hijacking in this scenario. As shown in Figure 1a, AS A only receives routes to the target since ROV-enabled AS B correctly drops the hijacker's malicious announcement. Consequently, AS A lacks a route to the hijacker in its routing table, limiting its control-plane visibility (see Figure 1b). AS A, unaware of the hijacker's presence, expects its traffic to reach the target correctly (green arrow). However, from a global perspective (see Figure 1c and 1d), AS A's traffic traverses legacy AS C en route to the target. If AS C accepts the hijacker's route (e.g., when the bogus route is preferred over the legitimate one or targets a sub-prefix), the traffic from AS A is actually forwarded to the hijacker. Note that AS A cannot observe the actual traffic forwarding unless it performs active data-plane probing or gains a broader view from external vantage points.

The above example showcases the unexpected downside of partial ROV deployment: *it prevents certain ASes from*

¹For brevity, we refer to the interdomain route (accepted by X) for a specific prefix announced by Y as "the route (from X) to Y".

observing bogus routes, contributing to the stealthiness of BGP hijacking attacks. We refer to such BGP hijacking, which is invisible to certain victims on the control plane, as *stealthy hijacking*. In general, an AS is at risk of stealthy hijacking when it has no route to the hijacker, but at least one legacy AS along the legitimate path accepts the bogus route; this applies to both exact-prefix and sub-prefix hijacking. Based on this principle, we formally define the risk in the next section.

B. Problem Formulation

Now, we formalize stealthy hijacking risk introduced by partial ROV deployment, starting from the Internet topology:

Definition 1 (Internet Topology). An Internet topology is a tuple $G = (V, E, V_{rov})$ where V is the set of all ASes in the Internet, E is the set of links between ASes in V , and $V_{rov} \subseteq V$ is the set of ASes that adopt ROV.

The Internet topology exhibits partial ROV deployment if and only if $V_{rov} \notin \{V, \emptyset\}$. We then define the AS-level route:

Definition 2 (AS-Level Route). Given $G = (V, E, V_{rov})$, let $u, v \in V$ and let p be a prefix announced by v . $R_G(u, v; p)$ denotes the AS-level route starting from u to reach the prefix p announced by v after Internet routing converges on G under existing routes, ROAs, and BGP policies. If it exists, $R_G(u, v; p) = (a_0, \dots, a_l)$, where $a_0 = u, a_l = v, a_0, \dots, a_l \in V, (a_i, a_{i+1}) \in E$ for $0 \leq i \leq l-1$; otherwise, $R_G(u, v; p) = ()$.

Based on them, we can define the stealthy hijacking risk:

Definition 3 (Stealthy Hijacking Risk). Given $G = (V, E, V_{rov})$, let $u, v, w \in V$ and let p be a prefix announced by v . The route $R_G(u, v; p)$ is at risk of stealthy hijacking by w if there exists a prefix p' , such that (i) $p' = p$ or p' is a sub-prefix of p , (ii) $R_G(u, w; p') = ()$, (iii) $R_G(u, v; p) \neq ()$, and (iv) there exists $a_i \in R_G(u, v; p)$, such that $a_i \neq u, a_i \neq v$, and $R_G(a_i, w; p') \neq ()$.

TABLE I: Tags for predefined incident behaviors.

Tag ¹	Definition ²	Data Source
Origin Relay	There exists M_2 such that $M_2 = O_1$.	Self-contained
Origin AS-Set	O_2 is in the form of AS-set.	Self-contained
Origin Related	O_1 and O_2 have a business relationship.	CAIDA [24]
Private ASN	The ASN of O_2 is reserved for private use.	IANA [25]
Similar Name	O_1 and O_2 have similar ³ organization names.	CAIDA [26]
Direct View	There exists M_1 such that $M_1 = V_2$.	Self-contained
Country Diff	O_1 and O_2 are located in different countries.	CAIDA [26]

¹ Tags in indicate route engineering practices, while tags in are informational.

² The notations are the same as in §IV-A.

³ Two strings are deemed similar if their fuzz partial ratio score is greater than 90.

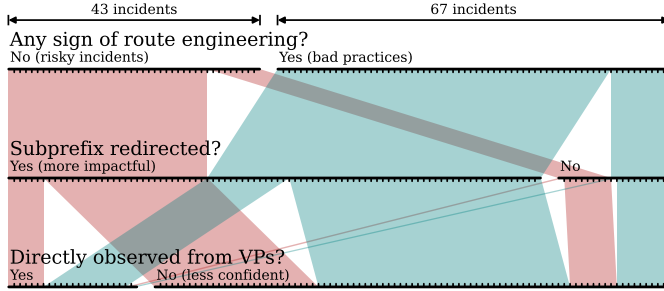


Fig. 2: Breakdown of the unique incident set.

Intuitively, given victim u , target v , and hijacker w , stealthy hijacking occurs when u cannot reach w , yet on u 's route to v , an AS a_i has a route to w . Thus, traffic originated from u follows the route towards v until it reaches a_i , who instead forwards the traffic to w . We call the “victim-target-hijacker” 3-tuple (u, v, w) a *stealthy hijacking instance*, and a_i the *risk-critical AS*. Each instance corresponds to a risk-critical AS; for example, in Figure 1, AS C is the risk-critical AS of the instance (A, E, G) . Note that each instance only reflects the risk, meaning that a potential stealthy hijacking attack *could* occur, but does not necessarily imply an actual occurrence. We base our risk analysis on the 3-tuple model, and refer to BGP hijacking that is not stealthy hijacking as *direct hijacking*.

Notably, stealthy hijacking follows the established BGP hijacking attack model but manifests in a more subtle form under partial ROV deployment. Any stealthy hijacking due to partial ROV deployment would also be possible (though not necessarily stealthy) without ROV, because removing ROV only improves attacker reachability without making any benign route more preferred. Thus, regardless of ROV deployment, victims that are subject to hijacking would continue to prefer routes with risk-critical ASes. We emphasize that the shift from direct to stealthy hijacking is a byproduct of partial ROV deployment in the current Internet ecosystem, rather than a flaw in ROV itself. However, stealthy hijacking enables attacks that can evade existing control-plane defenses [20]–[22] and hinder post-attack forensics [23]. Understanding its real-world prevalence and associated risk is thus critical to BGP security.

IV. UNCOVERING STEALTHY HIJACKING IN THE WILD

In this section, we develop heuristics for stealthy hijacking discovery, and empirically investigate the threat in the wild.

TABLE II: Overall impact of the incidents.

Type	#Countries	#Prefixes	#Origins	#Routes	#VPs
Risky incidents	16	60	36	773	48
Bad practices	24	103	43	3,611	50
Total	31	156	73	4,278	50

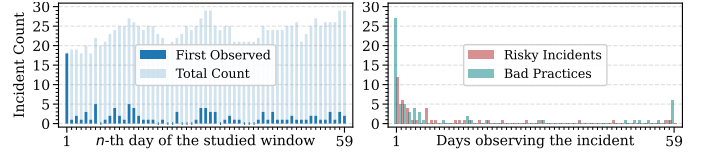


Fig. 3: Daily incident count (left) and incident duration (right).

A. Heuristics for Stealthy Hijacking Discovery

From Definition 3, we derive practical heuristics to discover stealthy hijacking instances based on routing table discrepancies. For clarity, we denote $p : V \cdots (M) \cdots O$ a route to prefix p , where V is the vantage point, O is the origin AS, and M (if present) represents an intermediate AS along the path. Given two routes $p_1 : V_1 \cdots (M_1) \cdots O_1$ and $p_2 : V_2 \cdots (M_2) \cdots O_2$, we examine the following conditions:

- 1) Conflict: p_2 equals or is a sub-prefix of p_1 , and $O_2 \neq O_1$.
- 2) Unauthorized: p_2/O_2 is RPKI-invalid while p_1/O_1 is valid.
- 3) Stealthiness: V_1 has no route to p_2 originated by O_2 .
- 4) Risk-critical AS: There exist M_1 and M_2 , with $M_1 = M_2$.
- 5) Risk-critical VP: There exists M_1 such that $M_1 = V_2$.

We first define the *loose heuristics*, which require conditions 1-4 to hold simultaneously to yield a “victim-target-hijacker” instance (V_1, O_1, O_2) , where M_1 (M_2) is the risk-critical AS. Since condition 4 infers the risk-critical AS from all intermediate ASes, the loose heuristics maximally utilize available route observations. However, the results may not be strictly reliable, as the risk-critical AS’s route to the hijacker is inferred from a route segment rather than directly observed. To improve reliability, we further propose the *strict heuristics*, replacing condition 4 with condition 5 to restrict risk-critical ASes to those that are also vantage points. This ensures strict adherence to Definition 3 and produces more reliable results.

The two heuristics offer a trade-off between breadth and confidence. In practice, we apply both in our empirical study to maximize discovery while accounting for different confidence levels. For analytical risk assessment, we rely solely on the strict heuristics, as our framework infers complete Internet-wide routes (elaborated later).

B. Real-World Observations and Insights

We now present our empirical study to track stealthy hijacking in the wild. Specifically, we analyze daily RIB snapshots taken at 12:00 UTC by the RouteViews [27] collectors *route-views2*, *amsix*, and *wide*, starting from January 1, 2025. These collectors are based in North America, Europe, and Asia, respectively. Each daily archive contains about 50 million BGP routes from over 370 vantage points. For each day’s snapshot, we apply the loose heuristics to discover potential

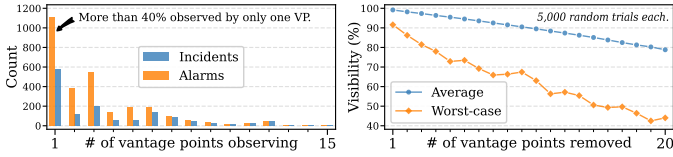


Fig. 4: Number of VPs observing each incident points (left), and overall incident visibility as VPs are randomly removed (right).

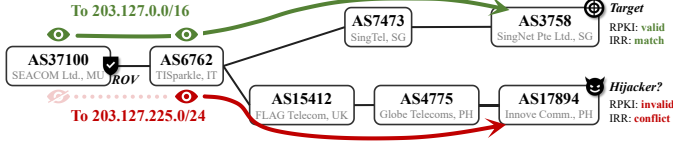


Fig. 5: A real-world stealthy hijacking incident.

stealthy hijacking instances. To ensure prefix-origin legitimacy, we cross-check RIPE NCC’s RPKI database [28], the RADb IRR database [29], and the five RIRs’ WHOIS databases [30], retaining only instances where the misbehaving origin is simultaneously RPKI-invalid, IRR-conflicting, and WHOIS-mismatching. To group related instances, we aggregate those affecting the same prefix into a single alarm, then merge alarms with the same misbehaving origin into a single incident.

To profile discovered incidents, we assign each incident with tags corresponding to predefined behaviors that either indicate route engineering or provide additional context. Table I details the tag definitions and data sources used. Incidents without any tags indicative of route engineering are considered particularly risky, while those linked to route engineering still exhibit stealthy hijacking but are more likely due to misconfigurations, such as improper route aggregation or private IP leasing without updating registries. Notably, incidents tagged *Direct View* follow the strict heuristics and are more reliable.

Findings. Over a two-month window starting January 1, 2025, we capture 1,394 potential stealthy hijacking incidents in the wild. Deduplicating them over the timeline yields 110 unique incidents. Figure 2 provides a detailed breakdown of them: 43 cases, with no signs of route engineering, are particularly risky, while the rest are attributed to bad operational practices. Among them, 91 involve sub-prefix hijacking, which tends to have more serious impacts, and 22 are directly observed from vantage points, thus of the highest confidence. In total, these incidents involve 4,278 routes observed by 50 vantage points, affecting 156 prefixes and 73 origins across 31 countries, as summarized in Table II. Over time, we observe 18-29 incidents per day, with 0-5 newly discovered daily (except on Day 1), as shown in Figure 3 (left). In terms of duration, 76 incidents (69.1%) persist for 7 days or fewer, while 17 (12.7%) last over 30 days, including 14 deemed bad operational practices, as shown in Figure 3 (right). Moreover, Figure 4 (left) shows that most incidents are seen by three or fewer vantage points, with over 40% visible to only one. This indicates a strong dependence of stealthy hijacking visibility on vantage point selection. Figure 4 (right) further confirms this: randomly removing just 20 vantage points leads to a 22% average drop

in observable incidents, and up to 55% in the worst case.

Takeaway: Stealthy hijacking in the wild is mostly short-lived and targets sub-prefixes, with new cases emerging almost daily and some persisting long-term, likely due to overlooked misconfigurations. Its exposure is sensitive to vantage points.

Case Study. We present an example incident in Figure 5. This incident persists throughout our study and shows no signs of route engineering. Both vantage points, AS37100 (*SEACOM*) and AS6762 (*TISparkle*), observe the prefix 203.127.0.0/16 announced by its legitimate origin, AS3758 (*SingNet*). Meanwhile, the sub-prefix 203.127.225.0/24 is announced by an unauthorized origin, AS17894 (*Innovate Comm.*). The two origins are in different countries and have no established relationship. Since AS37100 has deployed ROV with a 100% filtering rate [11], it discards the bogus /24 route. Yet, traffic from AS37100 or its customers to the /24 prefix still experiences hijacking when transited through legacy AS6762, which accepts the bogus route (the red path). Examination of AS37100’s looking glass further confirms the incident (detailed in Appendix A). We reported it to AS4775 (*Globe Tel.*) on February 10, 2025, and received promise to investigate.

Resources. We curate a high-confidence dataset of 318 real-world stealthy hijacking incidents, covering 2,178 unique routes, by selecting incidents tagged *Direct View* while excluding those with *Similar Name*². This dataset serves as ground truth for broader research and is specifically used to validate our framework in §VII. Beyond this study, we continue to run stealthy hijacking discovery as a service, implement on-demand data-plane validation based on RIPE Atlas [32], and provide a feature-rich frontend to publish daily incident reports. Readers are encouraged to explore these resources at <https://yhchen.cn/stealthy-bgp-hijacking>.

We conclude this section by reiterating a key observation, *i.e.*, stealthy hijacking exposure is sensitive to vantage point selection. This highlights the need for a comprehensive view of global routing to enable deterministic and exhaustive risk assessment, motivating our analytical approach presented next.

V. THE SHAMAN FRAMEWORK

“In ancient times, a shaman guided through unseen perils.”

A. Overview

We present SHAMAN, a BGP route inference framework dedicated to analytical assessment of stealthy hijacking risk. As shown in Figure 6, SHAMAN takes AS relationships and ROV measurements from multiple sources as input. It then (i) reconstructs the Internet topology, (ii) performs matrix-based BGP route inference, and (iii) applies the strict heuristics to assess stealthy hijacking risk across the inferred routes.

The core rationale is to determine whether any AS along legitimate routes can forward traffic to potential hijackers. Achieving comprehensive risk assessment thus requires knowledge of both *all* routes to benign ASes and *all* routes to

²Incidents tagged *Similar Name* are excluded to remove cases that are likely caused by private interconnection between affiliated ASes [31].

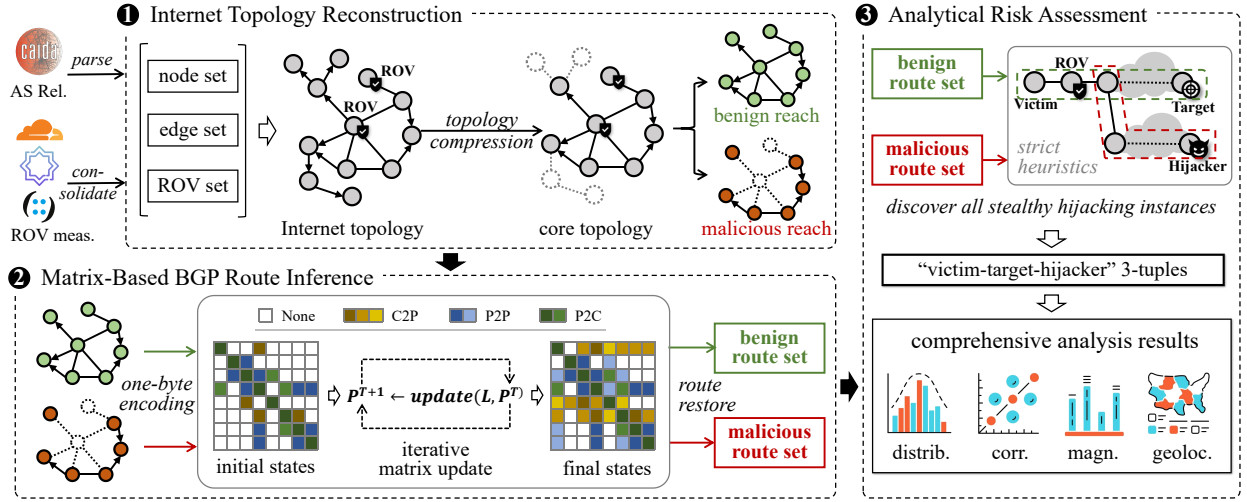


Fig. 6: Workflow of the SHAMAN framework.

potential hijackers, posing a primary challenge given the Internet scale. To address this, SHAMAN compresses the Internet topology during reconstruction and extracts the benign reach and malicious reach, *i.e.*, the respective sub-topologies traversable by benign and malicious announcements (see ❶). By converting them into matrices, where each cell encodes compact routing information, SHAMAN iteratively updates these matrices through highly optimized matrix operations to infer Internet-wide routes (see ❷). It then identifies risk-critical ASes by checking each benign route with the malicious route set (see ❸). We elaborate on each step below.

B. Internet Topology Reconstruction

To reconstruct the Internet topology $G = (V, E, V_{rov})$, we use the CAIDA AS relationship dataset [24] to obtain the set of ASes (V) and AS-to-AS links (E). Each link $(a_i, a_j, r) \in E$, where $a_i, a_j \in V$, is associated with a relationship type $r \in \{C2P, P2P, P2C\}$. To determine the set of ROV-enabled ASes (V_{rov}), we consolidate measurements from three sources: APNIC [11], RoVista [9], and Cloudflare [10]. APNIC and RoVista report ROV-filtering rates per covered AS, and we consider an AS to be ROV-enabled if its filtering rate is at least 80%, a confidence threshold adopted in prior work [18] and also proved practical in our own experiments. Cloudflare, meanwhile, directly provides a list of ROV-enabled ASes. To maximize coverage, we include in V_{rov} any AS identified as ROV-enabled by at least one of these sources.

We next compress the resulting topology. We notice that, under certain conditions, the routing table of a single-homed AS can be directly derived from its upstream transit AS, which exchanges all routes with it. As such, there is no need to include these single-homed ASes in the computationally intensive route inference process; instead, we remove them to form a *core topology* for route inference, and later recover their routing tables from their respective upstreams. This preserves the integrity of inferred routes while reducing the number of vertices by 36.3%, greatly improving inference efficiency. We defer details of the compression method in Appendix B.

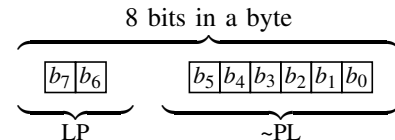
From the core topology, we further derive the benign reach and the malicious reach. The benign reach, where benign announcements can propagate freely, is identical to the core topology, while the malicious reach, denoting the restricted area where malicious announcements can propagate as ROV-enabled ASes block them, is obtained by removing all ROV-enabled ASes from the core topology. Route inference is later performed on these reaches to obtain legitimate AS-level routes and potential AS-to-hijacker routes, respectively.

C. Matrix-Based BGP Route Inference

We leverage highly optimized matrix operations for efficient BGP route inference, addressing the limitations of existing methods [13]–[15], which incur prohibitive overhead in generating complete BGP routes at the Internet scale.

Inference Criteria. We follow the same criteria commonly used in prior works [12], [33], [34], which are derived from the Gao-Rexford model [35] and capture typical BGP control-plane behaviors. Specifically, the best-route selection follows a three-step process: (i) prefer routes with the highest local preference, (ii) select routes with the shortest AS_path, and (iii) break ties randomly. By default, local preference is set to reflect business incentives, *i.e.*, a route received from a customer is preferred over one from a peer, which is preferred over one from a provider. Additionally, the valley-free constraint [35] is enforced in route propagation, *i.e.*, routes learned from a peer or provider are only forwarded to customers.

One-Byte Route Priority Encoding. Best-route selection is computationally intensive. To boost the process, we propose a one-byte route priority encoding that enables fast comparison and efficient update. Specifically, the priority of a route during selection is determined by two key properties, *i.e.*, local preference and path length. We use one byte to represent both:



where the LP field (the two most significant bits) encodes the exact value of local preference, and the \sim PL field (the six least significant bits) encodes the *bitwise complement* value of path length. The bit values of LP and PL are defined as follows:

$$\text{LP} = \begin{cases} (11)_2, & \text{for routes received from customers} \\ (10)_2, & \text{for routes received from peers} \\ (01)_2, & \text{for routes received from providers} \\ (00)_2, & \text{for unreachable origins} \end{cases} \quad (1)$$

$$\text{PL} = \begin{cases} l \text{ in 6-bit form,} & \text{for path length of } l \\ (000000)_2, & \text{for unreachable origins} \end{cases} \quad (2)$$

For example, a route received from a customer with a path length of 5 has $\text{LP} = (11)_2$ and $\text{PL} = (000101)_2$, resulting in a priority byte of $(11\ 111010)_2$. If the origin is unreachable or the route is yet to know, we use $(00\ 111111)_2$ as a placeholder for future update. Note that the 6-bit \sim PL field can represent a path length up to 63, so any longer route will be dropped. In our practice, no route exceeds this limit.

Compared with traditional data structures, our one-byte encoding maximizes memory utilization and enables fast priority comparison, *i.e.*, a greater byte value indicates a more preferable route. As a result, best-route selection becomes a simple task of finding the maximum value in a byte array. Moreover, its update during route propagation involves only basic arithmetic, *e.g.*, by subtracting one per hop to update the path length field (\sim PL). Such properties, as detailed later, help break down the complex iterations of BGP route inference into a series of highly optimized matrix operations.

Route Priority Update. We follow an iterative paradigm for BGP route inference. It starts with an initial state where no AS knows a route to any other. As each AS announces its presence, route announcements propagate hop by hop across the Internet topology. Each round of one-hop propagation, along with corresponding routing table updates, is an *iteration*.

We now describe how to update the one-byte encoding during an iteration. Consider a route to origin AS a_j in AS a_k 's routing table: after T iterations, we denote its priority byte by p_{kj}^T . In the next iteration, this route is forwarded one hop further. Suppose AS a_i is to receive this route, then the priority byte of the *received* route, denoted by \hat{p}_{ikj}^{T+1} , is determined by the a_i -to- a_k relationship type and the value of p_{kj}^T . If the a_i -to- a_k relationship exists and the propagation satisfies the valley-free constraint, \hat{p}_{ikj}^{T+1} is set accordingly: its LP field inherits the local preference value for the a_i -to- a_k relationship type (*i.e.*, the LP value of p_{ik}^l), and its \sim PL field equals p_{kj}^T 's \sim PL value minus one due to the increased path length. However, if the a_i -to- a_k relationship does not exist or the propagation violates the valley-free constraint, a_i is then not eligible to receive the route. In this case, we assign the placeholder value to \hat{p}_{ikj}^{T+1} .

Intuitively, this process involves two steps: (i) compute the LP field of \hat{p}_{ikj}^{T+1} , and (ii) assign the \sim PL field accordingly. For the first step, we introduce a custom operator \odot :

$$\text{LP}[\hat{p}_{ikj}^{T+1}] = \text{LP}[p_{ik}^l] \odot \text{LP}[p_{kj}^T], \quad (3)$$

where $\text{LP}[*]$ denotes the LP field of the corresponding priority

byte. The \odot operator maps all scenarios described above to corresponding outcomes; its truth table is shown in Table III.

TABLE III: Truth table of $\text{LP}[p_{ik}^l] \odot \text{LP}[p_{kj}^T]$. The 00 outputs in indicate that the input route does not exist or is not yet known, while those in result from valley-free violations.

$\text{LP}[p_{ik}^l] \backslash \text{LP}[p_{kj}^T]$	11 (P2C)	10 (P2P)	01 (C2P)	00 (None)
11 (P2C)	11	00	00	00
10 (P2P)	10	00	00	00
01 (C2P)	01	01	01	00
00 (None)	00	00	00	00

The \odot operator can be decomposed into basic arithmetic operations. Let $(ab)_2$, $(cd)_2$, and $(ef)_2$ denote the two bits in $\text{LP}[p_{ik}^l]$, $\text{LP}[p_{kj}^T]$, and $\text{LP}[p_{ik}^l] \odot \text{LP}[p_{kj}^T]$, respectively. Then, Table III can be expressed by minimal boolean expressions:

$$e = acd = a \cdot cd + 0 \cdot 0, \quad (4)$$

$$f = \bar{a}bd + \bar{a}bc + bcd = b \cdot cd + \bar{a}b \cdot (c + d). \quad (5)$$

We can thus implement \odot efficiently using shift and bitwise logic operations, which are vectorizable over matrices.

With \hat{p}_{ikj}^{T+1} 's LP field computed, its \sim PL field is determined:

$$\sim\text{PL}[\hat{p}_{ikj}^{T+1}] = \begin{cases} \sim\text{PL}[p_{kj}^T] - 1, & \text{if } \text{LP}[\hat{p}_{ikj}^{T+1}] \neq (00)_2 \\ (111111)_2, & \text{otherwise} \end{cases} \quad (6)$$

where $\text{LP}[*]$ and $\sim\text{PL}[*]$ denote the respective fields. Note that this two-branch computation can be further reformulated as a branchless expression without affecting correctness:

$$\sim\text{PL}[\hat{p}_{ikj}^{T+1}] = \sim\text{PL}[p_{kj}^T] - \text{PL}[p_{ik}^l], \quad (7)$$

where $\text{PL}[*]$ denotes the *exact value* of path length. We provide the proof of equivalence between Equations (6) and (7) in Appendix C. Since Equation (7) is independent of the LP field computation and is more amenable to vectorization, we implement Equation (7) in practice.

As \hat{p}_{ikj}^{T+1} indicates an updated route that a_i receives from a_k , the best-route selection at a_i involves comparing \hat{p}_{ikj}^{T+1} with p_{ij}^T and prefers the one with the higher value. The priority byte of the final selected route at the end of iteration $T+1$ is thus determined by considering all \hat{p}_{ikj}^{T+1} values across k :

$$p_{ij}^{T+1} = \max \{ p_{ij}^T, \hat{p}_{i0j}^{T+1}, \dots, \hat{p}_{i(n-1)j}^{T+1} \}, \quad (8)$$

where p_{ij}^T is known from iteration T and $\hat{p}_{ikj}^{T+1}, k = 0, \dots, n-1$ are computed via Equations (3)-(7). We further prove that there always exists some k such that $\hat{p}_{ikj}^{T+1} \geq p_{ij}^T$ (see Appendix D). Thus, the best-route selection process can be simplified as:

$$p_{ij}^{T+1} = \max \{ \hat{p}_{ikj}^{T+1} \}_{k=0}^{n-1}. \quad (9)$$

Based on Equation (9), we can update p_{ij}^T to p_{ij}^{T+1} for *each* pair of i and j . To accomplish this efficiently, we design a matrix representation scheme to organize all route priority bytes, and extend byte-wise operations to equivalent matrix-wise ones that are highly optimized for batch processing.

Matrix Representation. Extending the variable p_{ij}^T across all (i, j) pairs naturally forms an $n \times n$ matrix, where the cell at

i -th row and j -th column contains the corresponding byte p_{ij}^T . We thus define a state matrix P^T to collectively maintain all route priority bytes and update it recursively in place:

$$P^{T+1} = \text{UPDATE}(P^I, P^T), \quad (10)$$

where T starts at 0, and the generating function UPDATE performs the per-iteration computation on all route priority bytes as defined by Equations (3) to (9). The initial state P^0 is constructed by setting all diagonal cells to $(11\ 11111)_2$ and all off-diagonal cells to $(00\ 11111)_2$, since each AS has no external reachability knowledge other than their own existence at the start of route inference. Note that the diagonal cells indicate conceptually self-pointing routes, by which an AS can reach itself without traversing any other ASes. As such, these routes have the highest local preference and a path length of zero, and are always preferred over looped routes.

Besides self-pointing routes, one-hop routes from each AS to its neighbors form another prior knowledge. P^I captures these routes and is referenced during each iteration. Therefore, we treat P^I as a constant and pre-compute it using a separate matrix L (short for “Link”). Equation (10) is then altered:

$$P^{T+1} = \text{UPDATE}(L, P^T), \quad (11)$$

where L is initialized based on the given $G = (V, E, V_{rov})$:

$$L : l_{ij} = \begin{cases} (11\ 11111)_2, & \text{for } i = j \\ (11\ 11110)_2, & \text{for } (a_i, a_j, \text{P2C}) \in E \\ (10\ 11110)_2, & \text{for } (a_i, a_j, \text{P2P}) \in E \\ (01\ 11110)_2, & \text{for } (a_i, a_j, \text{C2P}) \in E \\ (00\ 11111)_2, & \text{otherwise} \end{cases} \quad (12)$$

The initialization of P^0 and L follows the one-byte encoding. It can also be verified that $L = P^I = \text{UPDATE}(L, P^0)$ by evaluating the first iteration. In this matrix form, the update of priority bytes can be naturally extended to matrix-wise operations. Additional technical details are provided in Appendix G.

Route Set Generation. Through matrix operations, we iteratively update P^T until no changes occur. During this process, we also record the next-hop AS for each selected route. Specifically, we maintain an $n \times n$ matrix N^T , where the cell at the i -th row and the j -th column stores the index of the next-hop AS for a_i to reach a_j after T iterations. N^T is updated alongside P^T . As iteration T yields the maximum priority byte for p_{ij}^{T+1} according to Equation (9), we simultaneously record the index k of the selected byte in the corresponding cell of N^{T+1} . Once inference completes, the full AS paths can be restored by recursively tracing the next-hop AS using N^T (detailed in Appendix E). In this way, we generate the complete set of AS-level routes across the Internet topology: On the benign reach, we obtain the benign route set, and on the malicious reach, we obtain the malicious route set.

D. Analytical Risk Assessment

Given the benign and malicious route sets, we systematically discover all potential stealthy hijacking instances. According to our strict heuristics, stealthy hijacking becomes possible only if the victim does not receive any malicious

route to the hijacker (otherwise it is direct hijacking), while at least one intermediate AS on the benign victim-to-target route accepts a malicious route to the hijacker. Therefore, we iterate through all benign routes, treating the vantage point AS as the victim and the origin AS as the target. We then check all ASes to identify potential hijackers that satisfy these conditions:

- The hijacker is neither the victim nor the target.
- No victim-to-hijacker route exists in the malicious set.
- There is an AS on the victim-to-target route for which the route to the hijacker exists in the malicious route set.

By exhaustively examining all benign routes, we obtain a complete set of stealthy hijacking instances that *could* occur in the current Internet. Based on these instances, we can assess the risk in a statistical manner, which is presented next.

VI. ASSESSING THE STEALTHY HIJACKING RISK

In this section, we assess the stealthy hijacking risk posed by the current ROV deployment through SHAMAN.

A. Framework Setup

We implement SHAMAN in Python 3.10 and use Numba to compile matrix operations to low-level C code with GPU acceleration. It performs on a Linux platform with an Intel Xeon E5-2650 CPU and an NVIDIA GeForce RTX 2080Ti GPU. For Internet topology reconstruction, we use the CAIDA AS relationship dataset [24] released on March 1, 2025, which contains 77,600 ASes and 709,737 AS relationships. The three ROV measurements are collected on the same date. APNIC [11] reports 43,042 ASes, of which 2,575 exhibit an ROV filtering ratio of at least 80%. RoVista [9] covers 32,486 ASes, with 6,655 meeting the same threshold. In addition, Cloudflare lists 165 ASes with full ROV filtering. Figure 7 summarizes their statistics. Collectively, we identify 7,275 ASes recognized as ROV-enabled by at least one source. This set forms our final input of ROV-enabled ASes to SHAMAN.

Using the AS relationship data and ROV measurements, we reconstruct the Internet topology. After compression, the resulting core topology contains 49,403 ASes (a 36.3% reduction) and 681,540 AS relationships (a 3.97% reduction). From this core topology, we derive both the benign reach and the malicious reach, and perform matrix-based BGP route inference on each. We run the inference for up to 20 iterations to ensure convergence. This generates the benign route set with 5,963,253,322 routes and the malicious route set with 2,399,622,350 routes. Based on these inferred routes, we discover the complete set of potential stealthy hijacking instances and conduct further assessment.

B. Risk Assessment Results

We now examine the stealthy hijacking risk in the current Internet based on SHAMAN’s results, gaining insights into its prevalence, distribution, and underlying topological features.

Overall Risk Level. We first assess the overall stealthy hijacking risk posed by the current partial ROV deployment. For

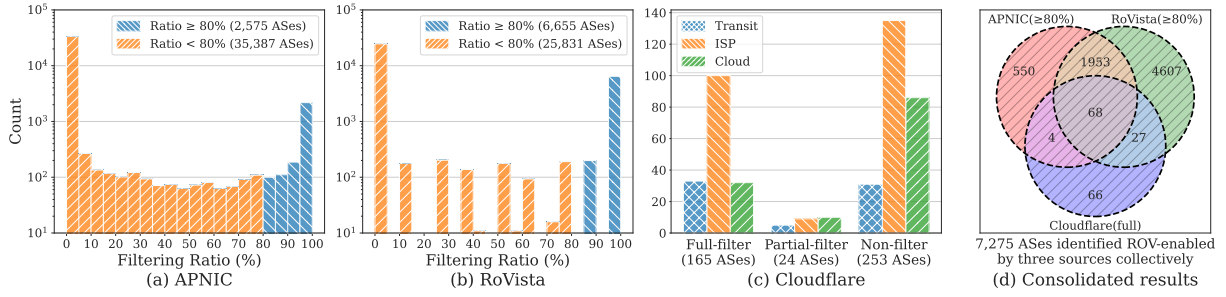


Fig. 7: The three ROV measurements and the consolidated results.

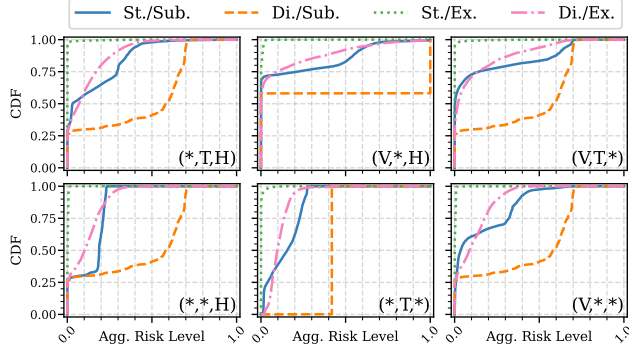


Fig. 8: CDF of aggregated risk levels.

any type of BGP hijacking (stealthy or direct, exact- or sub-prefix), we define its overall risk level as the statistical success probability across random “victim-target-hijacker” instances.

Our assessment with the 7,275 ROV-enabled ASes reveals an overall risk level of 0.141 for sub-prefix and 0.002 for exact-prefix stealthy hijacking. This suggests that a random stealthy hijacking attempt has over 14% probability of success. In comparison, with no ROV, both risks are 0. This stark contrast (0.141 versus 0) highlights the substantial increase in stealthy hijacking risk *exclusively* introduced by the current partial ROV deployment. Meanwhile, we observe the positive effect of ROV in reducing direct hijacking risk. With the same deployment, sub-prefix and exact-prefix direct hijacking risks drop to 0.419 and 0.106, with reductions of 57.5% and 70.1%, respectively, compared to a no-ROV scenario. These opposing trends, *i.e.*, rising stealthy risk and declining direct risk, reflect the double-edged effect of ROV in partial deployment.

Takeaway 1: While effectively mitigating direct hijacking risk, the current partial ROV deployment significantly amplifies stealthy hijacking risk from 0 to a 14.1% overall success probability. This risk arises solely due to ROV deployment.

Aggregated Risk Level. Since each hijacking instance involves three distinct entities, *i.e.*, the victim, the target, and the hijacker, we aggregate instances by specific entity combinations to assess the risk under certain conditions. For example, aggregating by the “target-hijacker” pair creates a set of groups where each comprises all instances with the same target and hijacker but varying victims. Assessing the risk level within each group gives the probability that a *specific* hijacker can successfully hijack a *specific* target on any *random* victim. We refer to this probability as the *aggregated risk level* of the

TABLE IV: Key statistics of aggregated risk levels.

Statistics ¹		Hijacking Type ²			
		St./Sub.	Di./Sub.	St./Ex.	Di./Ex.
$\mathcal{P}(*,T,H)$	min	0.000▼0.000	0.000▼0.000	0.000▼0.000	0.000▼0.000
	25th	0.000▲0.000	0.001▼0.992	0.000▼0.000	0.000▼0.183
	50th	0.033▲0.033	0.582▼0.412	0.000▲0.000	0.069▼0.246
	75th	0.296▲0.296	0.668▼0.326	0.000▲0.000	0.170▼0.329
	max	0.995▲0.995	0.737▼0.259	0.993▲0.993	0.737▼0.259
$\mathcal{P}(V*,H)$	min	0.000▼0.000	0.000▼0.000	0.000▼0.000	0.000▼0.000
	25th	0.000▼0.000	0.000▼1.000	0.000▼0.000	0.000▼0.125
	50th	0.000▼0.000	0.000▼1.000	0.000▼0.000	0.000▼0.296
	75th	0.218▲0.218	1.000▼0.000	0.000▼0.000	0.091▼0.464
	max	0.994▲0.994	1.000▼0.000	0.994▲0.994	1.000▼0.000
$\mathcal{P}(V,T,*)$	min	0.000▼0.000	0.000▼0.000	0.000▼0.000	0.000▼0.000
	25th	0.000▲0.000	0.001▼0.992	0.000▼0.000	0.000▼0.068
	50th	0.002▲0.002	0.582▼0.412	0.000▼0.000	0.020▼0.253
	75th	0.141▲0.141	0.668▼0.326	0.000▼0.000	0.139▼0.471
	max	0.740▲0.740	0.737▼0.259	0.511▲0.511	0.737▼0.259
$\mathcal{P}(*,*,H)$	min	0.000▼0.000	0.000▼0.000	0.000▼0.000	0.000▼0.000
	25th	0.000▲0.000	0.001▼0.992	0.000▼0.000	0.000▼0.260
	50th	0.188▲0.188	0.582▼0.412	0.002▲0.002	0.102▼0.245
	75th	0.212▲0.212	0.668▼0.326	0.004▲0.004	0.173▼0.266
	max	0.236▲0.236	0.737▼0.259	0.030▲0.030	0.602▼0.270
$\mathcal{P}(*,T,*)$	min	0.000▼0.000	0.419▼0.567	0.000▼0.000	0.006▼0.014
	25th	0.039▲0.039	0.419▼0.567	0.000▲0.000	0.066▼0.175
	50th	0.155▲0.155	0.419▼0.567	0.000▲0.000	0.097▼0.228
	75th	0.228▲0.228	0.419▼0.567	0.000▲0.000	0.137▼0.312
	max	0.309▲0.309	0.419▼0.567	0.154▲0.154	0.419▼0.567
$\mathcal{P}(V,*,*)$	min	0.000▼0.000	0.000▼0.000	0.000▼0.000	0.000▼0.000
	25th	0.007▲0.007	0.001▼0.992	0.000▲0.000	0.001▼0.302
	50th	0.035▲0.035	0.582▼0.412	0.000▲0.000	0.087▼0.268
	75th	0.321▲0.321	0.668▼0.326	0.005▲0.005	0.179▼0.234
	max	0.720▲0.720	0.737▼0.259	0.400▲0.400	0.493▲0.030
$\mathcal{P}(*,*,*)$		0.141▲0.141	0.419▼0.567	0.002▲0.002	0.106▼0.248

¹ 25th, 50th, and 75th represent the respective percentiles.

² St., Di., Sub., and Ex. stand for “stealthy”, “direct”, “sub-prefix”, and “exact-prefix”. The difference (▼/▲) is based on the comparison with a no-ROV scenario.

corresponding group. Aggregated risk levels across all such groups, *e.g.*, all unique “target-hijacker” pairs in this example, collectively form a probability distribution.

We represent an entity combination as a 3-tuple, where absent entities are denoted by * and present entities by their initials; for example, (*,T,H) represents aggregation by the “target-hijacker” pair. For each type of BGP hijacking, we assess its aggregated risk level by all entity combinations except (V,T,H), since aggregating by individual instances provides little statistical insight. We use $\mathcal{P}(\cdot)$ to denote the corresponding probability distribution. Notably, (*,*,*) represents no aggregation, so $\mathcal{P}(*,*,*)$ reduces to a single value, *i.e.*, the overall risk level presented in previous analysis.

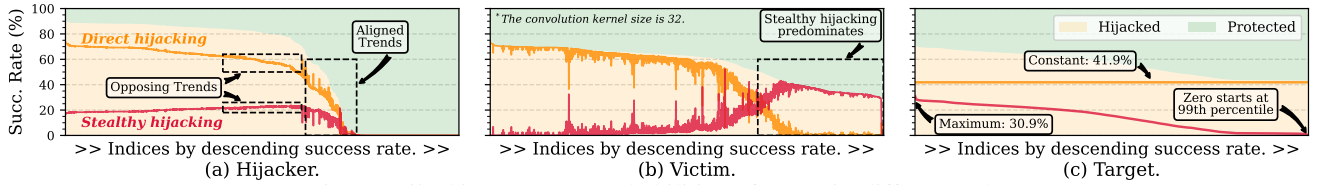


Fig. 9: Hijacking success probabilities of ASes in different roles.

Figure 8 shows the cumulative density function (CDF) of the aggregated risk levels, while Table IV presents key statistics and highlights differences compared to the no-ROV baseline. In general, the stealthy hijacking risk tends to concentrate on a few specific pairs when aggregated by two entities (see $\mathcal{P}(*,T,H)$, $\mathcal{P}(V*,H)$, and $\mathcal{P}(V,T,*)$), but spread more evenly across ASes when aggregated by a single entity (see $\mathcal{P}(*,*,H)$, $\mathcal{P}(*,T,*)$, and $\mathcal{P}(V,*,*)$). For example, $\mathcal{P}(*,T,H)$, the success probability distribution of a specific hijacker hijacking a specific target, has a median of 0.033 and a maximum of 0.995 under sub-prefix stealthy hijacking. In contrast, $\mathcal{P}(*,*,H)$, the success probability distribution of a specific hijacker hijacking any random route, has a higher median of 0.188 but a much lower maximum of 0.236. This disparity reflects the difference in hijacking capacity between *targeted* and *non-targeted* sub-prefix stealthy hijacking: the targeted type enables a few hijackers to achieve near-certain success, while the non-targeted yields only moderate success probability across all hijackers. A similar pattern is observed in exact-prefix stealthy hijacking, but not in direct hijacking, either exact-prefix or sub-prefix. Again, the double-edged effect of partial ROV deployment is evident, as most statistics of stealthy hijacking risk show an increase (highlighted in red in Table IV), while those of direct hijacking risk show a decrease (highlighted in blue). The only exception is the slight increase in the maximum value for exact-prefix direct hijacking, due to the increase of variance in probability distributions caused by random tie-breaking between competing routes for the exact prefixes.

Takeaway 2: Targeted stealthy hijacking achieves near-certain success on specific AS pairs (up to 99.5%), while non-targeted stealthy hijacking distributes risk more evenly across ASes (with a maximum of 23.6%). In contrast, direct hijacking does not exhibit these patterns.

Distribution across ASes. We further examine how hijacking risk distributes across ASes in different roles by assessing *role-specific* hijacking success probabilities, *i.e.*, $\mathcal{P}(V,*,*)$, $\mathcal{P}(*,T,*)$, and $\mathcal{P}(*,*,H)$. Interpretation of hijacking success probability differs by role: for a hijacker, it reflects its capability to hijack routes globally, while for a victim or target, it reflects its exposure to hijacking threats from any hijacker.

We now focus exclusively on sub-prefix hijacking, as it is generally more impactful. Figure 9(a)-(c) present the hijacking success probabilities for ASes acting as hijackers, victims, and targets, respectively, under both stealthy (red curve) and direct (orange curve) hijacking. In each figure, ASes are sorted along the X-axis in descending order of their overall hijacking success probability, *i.e.*, the sum of both stealthy and direct

hijacking success probabilities. To improve readability, the curves in Figure 9(b) are smoothed using a convolution kernel of size 32. We observe that over 50% of ASes, as attackers, can hijack over 80% of global routes despite ROV deployment; even the most protected 1% of ASes face considerable risk, with about 30% of their routes vulnerable as victims and 43% as targets. This highlights the limited protection ROV provides against BGP hijacking under its current deployment.

In Figure 9(a), as the overall success probability decreases along the X-axis, stealthy hijacking shows an upward trend, in contrast to the steady decline of direct hijacking. However, for a few ASes (as hijackers) where ROV filtering is highly effective, both direct and stealthy hijacking’s success probabilities align and drop towards zero. The opposing trend reveals a key tradeoff introduced by partial ROV deployment: while ROV blocks malicious routes, it also limits the route visibility of benign ASes, inadvertently exacerbating stealthy hijacking risk. Yet, the convergence of stealthy and direct hijacking’s success probabilities at the tail end suggests diminishing returns, *i.e.*, once ROV restrictions on attackers become sufficiently strong, stealthy hijacking also loses its advantage.

We observe a similar opposing trend in Figure 9(b), where 32% of ASes, as victims, face even higher stealthy hijacking risk than direct hijacking, with success probability reaching up to 72% in extreme cases. Moreover, stealthy hijacking’s success probabilities exhibit considerable variability, as indicated by noticeable spikes in the curves, suggesting that the stealthy hijacking risk faced by a victim is more case-specific and less predictable. In contrast, Figure 9(c) shows that stealthy hijacking risk distributes more evenly across ASes as targets, affecting over 99% of them with a maximum success probability of 30.9%. Notably, direct hijacking’s success probabilities remain constant at 41.9%, as sub-prefix direct hijacking depends solely on whether victims accept malicious routes from hijackers, irrespective of the targets.

Takeaway 3: While stealthy hijacking risk mostly opposes the overall risk trend across ASes, its diminishing gain is eventually suppressed as ROV’s restrictions on attackers prevail. Besides, the risk is more case-specific across victims but more evenly distributed across targets.

Distribution across Geolocations. We now look into the geographic distribution of stealthy hijacking risk, using the MaxMind GeoLite2 dataset [36] for AS geolocation. Figure 10(a)-(c) map all ASes globally, shaded by the ratio of hijacking instances involving them as hijackers, victims, and targets, respectively. We observe that the most capable potential hijackers cluster in Europe, South America (especially

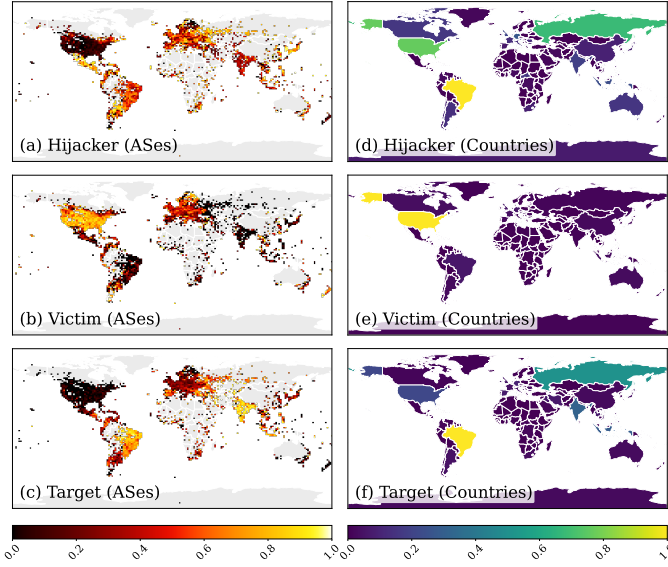


Fig. 10: Geographic distribution of stealthy hijacking risk.

Brazil), and North America (notably Mexico), while victims are mainly located in North America, with the US being the most affected. Meanwhile, the most targeted ASes concentrate in South America (particularly Brazil) and South Asia (notably India). These observations likely stem from differences in regional Internet connectivity. Figure 9(d)-(f) further show these role-specific ratios averaged over each country's total hijacking instances, capturing ASes' relative role tendency per country. We emphasize that this metric reflects statistical role-country correlation, rather than actual or intentional behavior of any country. Besides earlier observations, we find that ASes in the US and Russia are more likely (and capable) to act as hijackers if involved in stealthy hijacking, while both countries also exhibit a considerable fraction of ASes at risk as targets.

Takeaway 4: *ASes most effective in launching stealthy hijacking are mainly in Europe, South America, and North America; victim-prone ASes are mainly in North America; and target-prone ASes are mainly in South America and South Asia.*

Influencing Factors. To understand how Internet topology affects stealthy hijacking risk, we examine a broad range of topological features and evaluate their statistical correlation with the risk. As shown in Table V, we consider 24 features, including both basic metrics and composite statistics, grouped into seven categories: degree related (De0-De4), cone related (Co0/Co1), funnel related (Fu0/Fu1), hegemony related (He0-He3), distance related (Di0-Di2), cumulative statistics (Cm0-Cm3), and average statistics (Av0-Av3). Degree and distance related features reflect AS connectivity and centrality in the Internet topology [37]. Cone and funnel related features indicate the capability of ASes to provide and access transit services, respectively [31], [38], [39]. Hegemony related features measure AS interdependencies in forming routing paths [40]. Cumulative and average statistics, derived by aggregating these features over AS neighborhoods, capture broader topological characteristics at various scales.

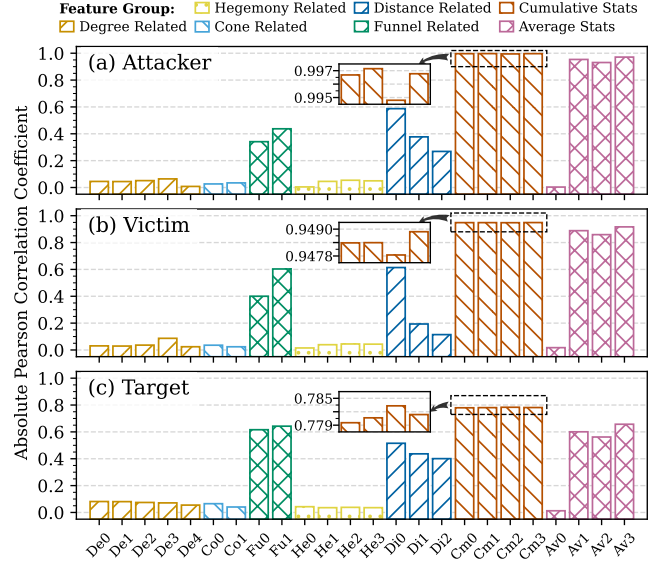


Fig. 11: Feature correlation with stealthy hijacking risk.

We use the Pearson Correlation Coefficient (PCC) [41] to quantify the linear correlation between each feature and stealthy hijacking risk (measured by hijacking success probability). This coefficient, denoted r_1 , ranges from -1 to 1, with values closer to 1 (or -1) indicating stronger positive (or negative) correlation. To capture potential non-linear correlation, given Internet topology's scale-free nature [37], we further apply quadratic regression and compute the PCC between the fitted and observed risk levels. The resulting coefficient, denoted r_2 , reflects the strength of a quadratic fit. For each feature, we report the higher absolute value of r_1 and r_2 in Figure 11. Notably, cumulative statistics exhibit the strongest correlation overall, with r_2 of Cm1 (Cumulative ROV Hegemony) reaching up to 0.997, 0.948, and 0.782 for hijackers, victims and targets, respectively. Certain funnel related features, e.g., Fu1 (ROV Funnel Size), distance related features, e.g., Di0 (Minimum ROV Distance), and average statistics, e.g., Av1 (Average ROV Hegemony), also exhibit relatively strong correlation with stealthy hijacking risk.

To further examine the most correlated features, we plot ASes on a plane, where the X-axis represents the feature value and the Y-axis represents the stealthy hijacking risk level, as shown in Figure 12. Fitted values from linear and quadratic regressions are presented with blue lines and orange curves, respectively, with shaded areas indicating estimation errors. The risk levels show a clear quadratic correlation with cumulative statistics (see Figure 12(a1), (b1), and (c1)), but less so with other features. Particularly, Cm1 in (a1) measures the fraction of routes traversing any ROV-enabled AS before reaching the given AS's reachable ASes within malicious reach. This statistic reflects the range of potential risk-critical ASes that the given AS as a hijacker can compromise. As this range expands, more ASes become potential targets, yet potential victims become fewer, resulting in the observed parabolic curve. The near-perfect quadratic fit (with r_2 approaching 1.00)

TABLE V: Topological features used to analyze factors influencing stealthy hijacking risk.

Feature Group	Feature Name	Description
Degree Related	De0 (Node Degree)	Number of edges (AS relationships) connected to the node (AS).
	De1 (Out Degree)	Number of outbound edges (P2C/P2P relationships) from the node (AS).
	De2 (In Degree)	Number of inbound edges (C2P/P2P relationships) to the node (AS).
	De3 (Provider Degree)	Number of direct providers of the AS.
	De4 (Customer Degree)	Number of direct customers of the AS.
Cone Related	Co0 (Customer Cone Size)	Number of direct or indirect customers of the AS.
	Co1 (ROV Cone Size)	Number of direct or indirect ROV-enabled customers of the AS.
Funnel Related	Fu0 (Provider Funnel Size)	Number of direct or indirect providers of the AS.
	Fu1 (ROV Funnel Size)	Number of direct or indirect ROV-enabled providers of the AS.
Hegemony Related	He0 (AS Hegemony)	Ratio of routes that traverse the AS.
	He1 (ROV Hegemony)	Ratio of routes traversing both ROV-enabled ASes and the given AS.
	He2 (Pre-ROV Hegemony)	Ratio of routes traversing ROV-enabled ASes before the given AS.
	He3 (Post-ROV Hegemony)	Ratio of routes traversing ROV-enabled ASes after the given AS.
Distance Related	Di0 (Minimum ROV Distance)	Minimum path length from the AS to any reachable ROV-enabled AS.
	Di2 (Maximum ROV Distance)	Maximum path length from the AS to any reachable ROV-enabled AS.
	Di3 (Average ROV Distance)	Average path length from the AS to all its reachable ROV-enabled ASes
Cumulative Statistics	Cm0 (Cumulative AS Hegemony)	Sum of He0 (AS Hegemony) over ASes reachable by the given AS within malicious reach.
	Cm1 (Cumulative ROV Hegemony)	Sum of He1 (ROV Hegemony) over ASes reachable by the given AS within malicious reach.
	Cm2 (Cumulative Pre-ROV Hegemony)	Sum of He2 (Pre-ROV Hegemony) over ASes reachable by the given AS within malicious reach.
	Cm3 (Cumulative Post-ROV Hegemony)	Sum of He3 (Post-ROV Hegemony) over ASes reachable by the given AS within malicious reach.
Average Statistics	Av0 (Average AS Hegemony)	Average of He0 (AS Hegemony) over ASes reachable by the given AS within malicious reach.
	Av1 (Average ROV Hegemony)	Average of He1 (ROV Hegemony) over ASes reachable by the given AS within malicious reach.
	Av2 (Average Pre-ROV Hegemony)	Average of He2 (Pre-ROV Hegemony) over ASes reachable by the given AS within malicious reach.
	Av3 (Average Post-ROV Hegemony)	Average of He3 (Post-ROV Hegemony) over ASes reachable by the given AS within malicious reach.

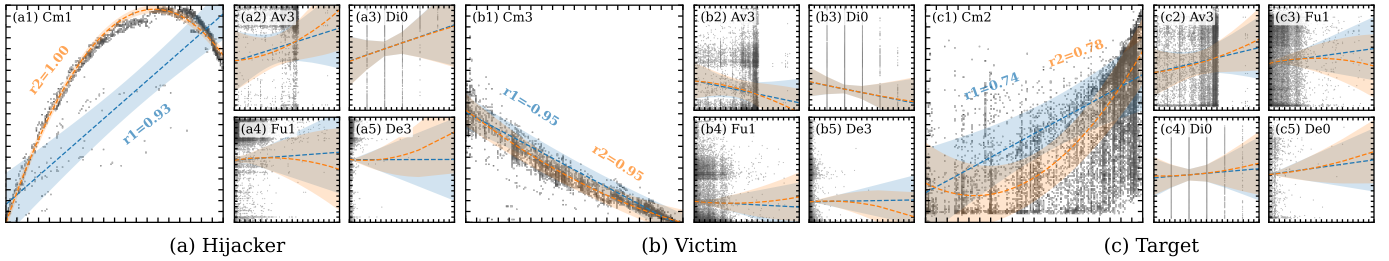


Fig. 12: Illustration of statistical correlation between selected features and stealthy hijacking risk.

suggests that cumulative statistics are reliable indicators of stealthy hijacking risk. Notably, the opposite concavity of the parabola in (a1) compared to (b1) and (c1) reflects the opposite interpretation of risk levels in terms of different roles, *i.e.*, hijacking capability for ASes acting as hijackers, and hijacking susceptibility for those as victims or targets.

Takeaway 5: Cumulative statistics of AS hegemony show the strongest quadratic correlation with stealthy hijacking risk, making them powerful indicators for predicting risk levels.

Risk Attribution. To better understand how risk-critical ASes and ROV-enabled ASes contribute to the risk, we examine the frequency of stealthy hijacking instances associated with them. We define the risk-critical AS as the first AS along the victim-to-target route that has a route to the hijacker, and the responsible ROV-enabled AS as the first ROV-enabled AS along that path. We attribute all stealthy hijacking instances to 8,323 unique risk-critical ASes and 1,608 unique ROV-enabled ASes. Figure 13 shows the contribution of these ASes to stealthy hijacking risk. The blue curves represent the count of instances associated with each AS, and the orange curves show the cumulative percentage. Dashed lines mark the 80% cut-off, revealing a pronounced long-tail effect, *i.e.*, a vital few account for the majority of the risk. That is, 2.94% of risk-critical ASes (245 out of 8,323) or 2.24% of ROV-enabled

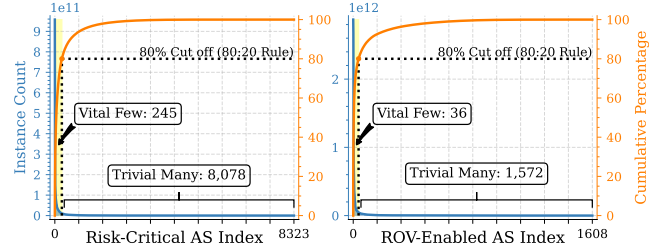


Fig. 13: Attribution of stealthy hijacking risk to risk-critical ASes (left) and ROV-enabled ASes (right).

ASes (36 out of 1,608) are responsible for 80% of all stealthy hijacking instances. This observation highlights the importance of focusing mitigation efforts on a small set of ASes to effectively reduce the overall risk of stealthy hijacking.

Takeaway 6: A small fraction of risk-critical and ROV-enabled ASes account for the majority of stealthy hijacking risk, calling for focused risk mitigation efforts on these key ASes.

Evolution Pattern. Finally, we analyze how stealthy hijacking risk evolves with increasing ROV deployment, as shown in Figure 14. Our preliminary assessment based on data from October 1, 2023 identifies 778 ROV-enabled ASes, which accounts for approximately 1% of the 75k ASes active at the time. It reports an overall stealthy hijacking success probability

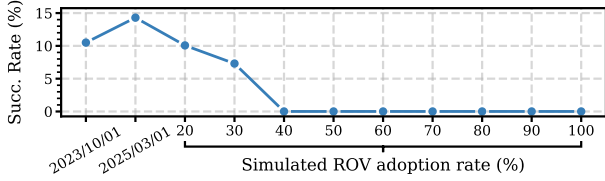


Fig. 14: Stealthy hijacking risk evolution.

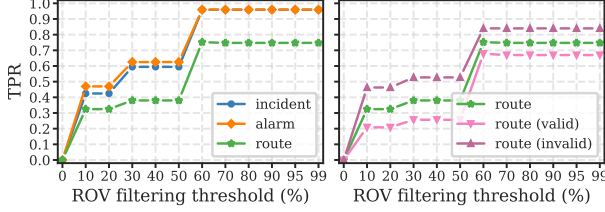


Fig. 15: True positive rate of our framework.

(including both sub-prefix and exact-prefix ones) of 0.105. In comparison, data from March 1, 2025 identify 7,275 ROV-enabled ASes (roughly 10% of all ASes), with the risk rising to 0.145, as reported earlier. This reflects an increase in overall risk alongside real-world ROV deployment. Moreover, we simulate future ROV deployment by progressively relaxing the threshold for identifying ASes as ROV-enabled, and the results suggest that the risk begins to decline as deployment increases. Specifically, a 10% additional increase in the ROV deployment rate reduces the hijacking success probability to 0.101, and a rate exceeding 40% reduces the risk to near zero.

Takeaway 7: The stealthy BGP hijacking risk shows an overall “rise-then-decline” evolution pattern along ROV deployment, and we may now be entering the declining phase.

VII. FRAMEWORK PERFORMANCE EVALUATION

A. Risk Discovery Effectiveness

When evaluating the effectiveness of SHAMAN in risk discovery, we use the curated real-world incident dataset described in §IV as the ground truth. The dataset contains 318 high-confidence stealthy hijacking incidents captured in the wild. Each incident consists of one or more related alarms triggered by the same misbehaving origin, and each alarm includes one or more pairs of routes indicating stealthy hijacking to the same prefix. In total, the dataset contains 347 alarms and 2,178 routes. As the best-effort ground truth contains only positive instances³, we report the True Positive Rate (TPR) as the main metric, reflecting how effectively SHAMAN infers real-world instances.

We begin by measuring SHAMAN’s TPR at the route level. As per the strict heuristics in §IV, each ground truth route pair includes one RPKI-valid and one RPKI-invalid route. For both routes, we extract the vantage point and origin AS to look up the corresponding route inferred by SHAMAN. If the inferred route pair also satisfies the strict heuristics, meaning that SHAMAN’s inference matches the ground truth, we consider

³It is infeasible to establish negative ground, because it is inherently difficult to find a case and tell whether it has not occurred or cannot occur.

TABLE VI: True positive rate under ablation of ROV sources.

Sources ¹			True Positive Rate ²					#ASes
A	R	C	Incident	Alarm	Route	Rt.V.	Rt.Iv.	
●	●	●	0.9591	0.9597	0.7521	0.6782	0.8400	7,275
○	●	●	0.7862	0.8012	0.6290	0.4519	0.8400	6,725
●	○	●	0.2767	0.2882	0.4201	0.0971	0.8048	2,668
●	●	○	0.8019	0.8184	0.5969	0.3902	0.8431	7,209
●	○	○	0.2767	0.2882	0.4201	0.0971	0.8048	2,575
○	●	○	0.6164	0.6484	0.5202	0.2492	0.8431	6,655
○	○	●	0.0629	0.0576	0.4656	0.0169	1.0000	165
○	○	○	0.0618	0.0720	0.4449	0.0166	0.9551	1,000

¹ A, R, and C denote APNIC, RoVista, and Cloudflare, resp. The row in **bold** is ours. The row without any source randomly selects 1,000 ASes as ROV-enabled.

² Rt.V. and Rt.Iv. indicate valid and invalid routes, resp. Highest values are in bold.

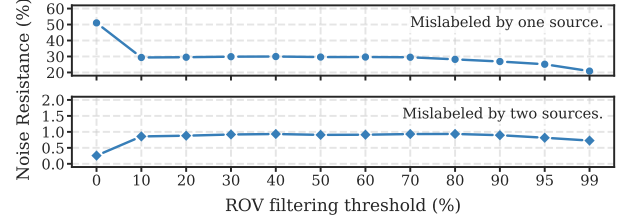


Fig. 16: Noise resistance of multi-source ROV input.

both routes are effectively inferred. Route-level TPR is defined as the fraction of such effectively inferred routes. We further measure alarm-level TPR as the fraction of alarms with at least one route pair effectively inferred, and incident-level TPR as the fraction of incidents containing at least one such alarm.

We present SHAMAN’s TPR across varying ROV filtering thresholds in Figure 15 (left). As the threshold increases, fewer but more confident ROV-enabled ASes are selected, and the TPR increases steadily. Beyond the 0.6 threshold, inference remains consistently strong, reaching 0.959, 0.960, and 0.752 at the incident, alarm, and route levels, respectively. This means that over 95% of these real-world incidents are successfully flagged risky by SHAMAN. Figure 15 (right) further breaks down route-level TPR by RPKI validity. As expected, SHAMAN is slightly less effective on RPKI-valid routes due to the stealthiness requirement (condition 3 in §IV-A) that applies exclusively to them. Still, SHAMAN achieves up to 0.678 TPR on these routes. Overall, these results demonstrate the fidelity of SHAMAN’s analytical output and inform a recommended threshold range, which justifies our default choice of 0.8.

B. Input Ablation and Robustness

We examine how incorporating multiple ROV measurement sources improves SHAMAN’s effectiveness and robustness. Table VI reports SHAMAN’s TPR under ablation of the three sources, with the ROV filtering threshold fixed at 0.8. In general, excluding any source leads to a noticeable drop in the coverage of ROV-enabled ASes and a corresponding decline in SHAMAN’s effectiveness, except that using only the Cloudflare source yields the highest TPR on RPKI-invalid routes. This is because Cloudflare contributes only 165 ROV-enabled ASes, which, according to SHAMAN’s assessment, have limited effect in blocking propagation of RPKI-invalid routes. As a result, SHAMAN’s inference overestimates observation of

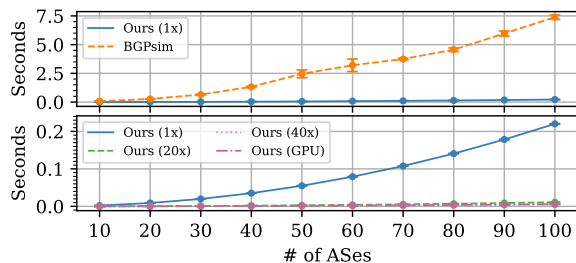


Fig. 17: Runtime overhead of route inference.

RPKI-invalid routes and thus aligns well with the ground truth. However, this setting is impractical and results in just 0.0169 TPR on RPKI-valid routes and 0.0629 at the incident level. After all, we conclude that integrating multiple reliable sources to obtain a more complete view of ROV deployment is crucial to achieve effective stealthy hijacking risk assessment.

We further look into SHAMAN’s robustness to noise in the three ROV measurements. We simulate scenarios where one or two sources randomly mislabel an ROV-enabled AS as non-ROV-enabled, and measure the probability that such ASes remain included in the final ROV set. The results are shown in Figure 16. When all three sources are used, SHAMAN retains a 20%-50% probability of preserving an AS despite a mislabel from a single source (see the top panel), and about a 1.0% probability when two sources mislabel the same AS. By comparison, using only one source result in zero probability of noise resistance, as any mislabel is directly accepted to the final ROV set. These results emphasize the robustness benefit of SHAMAN’s design using multiple ROV sources as input.

C. Runtime Overhead

For comparison, we select BGPSim [13], a highly optimized BGP simulator widely used in prior studies [42]–[44]. We test SHAMAN and BGPSim on the same Internet topology derived from CAIDA AS relationship data. Figure 17 reports their runtime overhead in generating routes within random AS subsets ranging from 10 to 100 ASes. Each run is repeated 10 times, with error bars indicating the 95% confidence interval. The top panel compares the runtime of SHAMAN (in a single-thread CPU setting) with BGPSim’s, with SHAMAN over 40 times faster in the worst case. The bottom panel evaluates SHAMAN’s runtime under different settings (1, 20, and 40 CPU threads, or a single GPU). The settings with 40 CPU threads or a single GPU show the best performance, completing route generation across 75,846 ASes in 5.22 hours with peak memory usage under 20 GiB. In contrast, an exponential fit of BGPSim’s overhead estimates 110 days for full route generation. As such, SHAMAN achieves a 500-fold speedup, providing the efficiency necessary for comprehensive Internet-scale stealthy hijacking risk assessment.

VIII. DISCUSSION

Complex Routing Policy. Our framework infers BGP routes based on the established Gao-Rexford model [13], [35], [42], [43]. While it cannot capture all real-world nuances, *e.g.*, hybrid or partial-transit relationships [31] and selective ROV

filtering [10], such cases are rare, *e.g.*, only seven ASes exhibit selective filtering [10]. Our evaluation on real incidents confirms the framework’s reliability despite these complexities. The design is also extensible: additional bytes can encode nuanced relationships, and selective filtering can be modeled by pruning certain links. We leave these to future work.

Mitigation Strategies. Increasing ROV deployment remains central to mitigating stealthy hijacking, as a sufficiently high adoption rate can suppress the risk (see Takeaway 3). ROV++ [12], which extends ROV with proactive rerouting and blackholing upon detecting invalid routes, can also mitigate the threat. Besides, detecting routing table discrepancies across vantage points, as demonstrated in §IV, enables timely alert of ongoing incidents. Operational practices such as announcing /24 IPv4 or /48 IPv6 prefixes also significantly limits the sub-prefix risk. Furthermore, information-sharing platforms, *e.g.*, mailing-lists where ROV-adopters publish digests of dropped routes, help potential victims identify risk-critical ASes. A similar routing policy was described in Cisco’s patent [45].

Incident Intent. Identifying intent behind BGP incidents, *i.e.*, whether malicious or not, is inherently challenging. We attempt systematic attribution methods (see Table I) and manual investigation (see Appendix A). While these help narrow down likely causes, definitive confirmation requires engagement with network operators, which we attempted in selected cases. Cross-layer analysis with external intelligence may offer further insights, which we leave as future work. However, regardless of intent, the operational impact of stealthy hijacking remains significant. Unintentional misconfigurations can lead to the same consequences as deliberate attacks.

IX. RELATED WORK

Stealthy Hijacking Analysis. Prior work mainly focuses on non-ROV-related stealthy hijacking based on short-lived routes [46], AS-path poisoning [47], or BGP communities [48]. To our knowledge, ROV++ [12] is the only recent work addressing ROV-related stealthy hijacking. It extends ROV with proactive rerouting and blackholing to mitigate the threat, showing promising benefits at early adoption. Particularly, ROV++ adopters can effectively secure all routes traversing them. If AS B in Figure 1 deploys ROV++, it drops the invalid route and seeks an alternative route in its RIB that avoids ASes in the invalid route (*i.e.*, AS C, F, and G), thus avoiding the risk-critical ASes. If such route is not available, it blocks traffic to the prefix to prevent hijacking. However, in certain mixed-deployment scenarios, *e.g.*, when AS B deploys ROV and AS A deploys ROV++, stealthy hijacking is still possible because the malicious route never reaches AS A, thus failing to trigger the countermeasures in ROV++. ROV++ focuses on mitigation strategies and does not provide real-world evidence or systematic assessment of the threat’s prevalence and impact. In contrast, our work aims at real-world stealthy hijacking discovery and systematic risk assessment, addressing this critical gap. Moreover, our risk attribution analysis complements ROV++ by identifying where its deployment would be most effective. For example, the 36

ROV-enabled ASes in Figure 13, which are mostly Tier-1 ASes and account for 80% of risk instances, are prime candidates for deploying ROV++.

ROV Deployment Measurement. ROV deployment is typically measured by analyzing ASes’ data-plane reachability to RPKI-valid and invalid prefixes [9]–[11], [18], [19], [33], [49]–[51]. These studies vary in probing sources and prefix selection. For example, RoVista [9] uses IPID side channels; APNIC [11] conducts large-scale probing via its infrastructure; and Cloudflare [10] hosts test sites and crowdsources measurements. Despite diverse techniques, each offers only partial coverage due to scalability limits. To improve visibility, we consolidate these three representative sources [9]–[11] for a more comprehensive view of ROV deployment.

BGP Route Inference. Existing BGP route inference methods are either simulation-driven [13], [14], [52]–[54] or data-driven [15], [55]–[58]. Simulation-driven methods simulate the route exchange using heuristics such as the Gao-Rexford model [59]. For example, Brandt et al. [13] implement BGPsim with bi-directional search to improve performance. Data-driven methods infer AS-level paths from measurements. For example, Cunha et al. [57] design a traceroute-based prediction system. While effective at certain scales, these methods cannot generate complete Internet-scale routes efficiently. Our work, by contrast, computes all AS-level routes within a few hours.

X. CONCLUSION

In this paper, we develop effective heuristics to discover stealthy BGP hijacking and conduct the first empirical study to track it in the wild, contributing a curated dataset and a monitoring service. To assess the risk comprehensively, we further design SHAMAN, a framework that integrates multiple sources for accurate ROV deployment, leverages matrix operations to infer Internet-wide routes, and enables systematic risk analysis via a 3-tuple model. It generates Internet-scale routes within hours and achieves 95.9% incident-level accuracy. Assessing over 8.3 billion routes reveals a 14.1% success probability for stealthy hijacking, with targeted attacks reaching up to 99.5%.

ETHICS CONSIDERATIONS

We carefully consider several ethical aspects to ensure that our study adheres to established ethical standards. Our study only uses publicly available data, and we strictly comply with all terms of use. Our study does not disclose any personally identifiable information or private routing policies beyond what is already publicly available. We do not perform any large-scale active probing or interfere with live routing systems, thereby ensuring no impact on real-world traffic or network stability. The real-world incidents captured during our study are responsibly disclosed to relevant network operators.

ACKNOWLEDGMENT

We sincerely thank our Shepherd and all anonymous reviewers for their valuable comments. This work is supported in part by NSFC under Grant 62132011, Grant 62472247, and Grant 62425201. Qi Li is the corresponding author of the paper.

REFERENCES

- [1] M. Lepinski and K. Sriram, “Rfc 8205: Bgpsec protocol specification,” 2017.
- [2] P. v. Oorschot, T. Wan, and E. Kranakis, “On interdomain routing security and pretty secure bgp (psbgp),” *ACM TISSEC*, 2007.
- [3] R. White, “Securing bgp through secure origin bgp (sobgp),” *ABI/INFORM Business Communications Review*, 2003.
- [4] S. Kent, C. Lynn, and K. Seo, “Secure border gateway protocol (s-bgp),” *IEEE JSAC*, 2000.
- [5] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein, “Rfc 6811: Bgp prefix origin validation,” 2013.
- [6] M. Lepinski and S. Kent, “Rfc 6480: an infrastructure to support secure internet routing,” 2012.
- [7] R. Bush, “Rfc 7115: Origin validation operation based on the resource public key infrastructure (rpki),” 2014.
- [8] NIST, “Rpki-rov history of unique prefix-origin pairs (ipv4),” 2025, accessed Mar. 18, 2025. [Online]. Available: <https://rpki-monitor.antd.nist.gov/>
- [9] W. Li, Z. Lin, M. I. Ashiq, E. Aben, R. Fontugne, A. Phokeer, and T. Chung, “Rovista: Measuring and analyzing the route origin validation (ROV) in RPKI,” in *ACM IMC*, 2023.
- [10] Cloudflare, “Is bgp safe yet?” 2025, accessed Mar. 18, 2025. [Online]. Available: <https://isbgpsafeyet.com/>
- [11] Huston, Geoff, “Measuring roas and rov,” 2025, accessed Mar. 18, 2025. [Online]. Available: <https://stats.labs.apnic.net/rpki>
- [12] R. Morillo, J. Furuness, C. Morris, J. Breslin, A. Herzberg, and B. Wang, “ROV++: improved deployable defense against BGP hijacking,” in *ISOC NDSS*, 2021.
- [13] M. Brandt and H. Schulmann, “Optimized BGP simulator for evaluation of internet hijacks,” in *IEEE INFOCOM WKSHPs*, 2021.
- [14] J. Furuness, C. Morris, R. Morillo, A. Herzberg, and B. Wang, “Bgpy: The BGP python security simulator,” in *ACM CSET WKSHPs*, 2023.
- [15] X. Li, T. Zhou, Z. Cai, and J. Su, “Realizing fine-grained inference of as path with a generative measurable process,” *IEEE/ACM TON*, 2023.
- [16] H. Geoff, “The 32-bit as number report,” APNIC, Tech. Rep., 2025. [Online]. Available: <https://www.potaroo.net/tools/asn32/>
- [17] M. Doug, “A brief history of the internet’s biggest bgp incidents,” Kentik, Tech. Rep., 2023. [Online]. Available: <https://www.kentik.com/blog/a-brief-history-of-the-internets-biggest-bgp-incidents/>
- [18] W. Chen, Z. Wang, D. Han, C. Duan, X. Yin, J. Yang, and X. Shi, “ROV-MI: large-scale, accurate and efficient measurement of ROV deployment,” in *ISOC NDSS*, 2022.
- [19] T. Hlavacek, H. Schulmann, N. Vogel, and M. Waidner, “Keep your friends close, but your routerservers closer: Insights into RPKI validation in the internet,” in *USENIX Security*, 2023.
- [20] Y. Chen, Q. Yin, Q. Li, Z. Liu, K. Xu, Y. Xu, M. Xu, Z. Liu, and J. Wu, “Learning with semantics: Towards a {Semantics-Aware} routing anomaly detection system,” in *USENIX Security*, 2024.
- [21] Y. Dong, Q. Li, R. O. Sinnott, Y. Jiang, and S. Xia, “Isp self-operated bgp anomaly detection based on weakly supervised learning,” in *IEEE ICNP*, 2021.
- [22] T. Shapira and Y. Shavitt, “Ap2vec: an unsupervised approach for bgp hijacking detection,” *IEEE TNSM*, 2022.
- [23] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, “Profiling bgp serial hijackers: capturing persistent misbehavior in the global routing table,” in *ACM IMC*, 2019.
- [24] CAIDA, “As relationships dataset,” 2023, accessed Mar. 18, 2025. [Online]. Available: <https://www.caida.org/catalog/datasets/as-relationships/>
- [25] J. Mitchell, “Rfc 6996: Autonomous system (AS) reservation for private use,” 2013.
- [26] CAIDA, “As organization dataset,” 2023, accessed Mar. 18, 2025. [Online]. Available: <https://www.caida.org/catalog/datasets/as-organizations/>
- [27] University of Oregon Route Views Project, “Mrt format ribs and updates,” 2023, accessed Mar. 18, 2025. [Online]. Available: <http://routeviews.org/>
- [28] RIPE NCC, “Rpki repository archive,” 2025. [Online]. Available: <https://ftp.ripe.net/rpki>
- [29] RADb, “The internet routing registry,” 2025. [Online]. Available: <ftp.radb.net>
- [30] Marco d’Itri, “Intelligent whois client,” 2025. [Online]. Available: <https://github.com/rfc1036/whois>
- [31] V. Giotas, M. J. Luckie, B. Huffaker, and kc claffy, “Inferring complex AS relationships,” in *ACM IMC*, 2014.

- [32] RIPE NCC, “Ripe atlas,” 2025. [Online]. Available: <https://atlas.ripe.net/>
- [33] Y. Gilad, A. Cohen, A. Herzberg, M. Schapira, and H. Schulmann, “Are we there yet? on rpki’s deployment and security,” in *ISOC NDSS*, 2017.
- [34] T. McDaniel, J. M. Smith, and M. Schuchard, “Flexsealing BGP against route leaks: Peerlock active measurement and analysis,” in *ISOC NDSS*, 2021.
- [35] L. Gao and J. Rexford, “Stable internet routing without global coordination,” *IEEE/ACM TON*, 2001.
- [36] MaxMind, “Geolite2 free geolocation data,” 2023, accessed Mar. 18, 2025. [Online]. Available: <https://dev.maxmind.com/geoip/geolite2-free-geolocation-data>
- [37] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the internet topology,” *ACM SIGCOMM CCR*, 1999.
- [38] M. J. Luckie, B. Huffaker, A. Dhamdhare, V. Giotsas, and K. Claffy, “AS relationships, customer cones, and validation,” in *ACM IMC*, 2013.
- [39] L. Prehn, P. Foremski, and O. Gasser, “Kirin: Hitting the internet with distributed BGP announcements,” in *ACM AsiaCCS*, 2024.
- [40] R. Fontugne, A. Shah, and E. Aben, “The (thin) bridges of AS connectivity: Measuring dependency using AS hegemony,” in *Springer PAM*, 2018.
- [41] K. Pearson, “VII. note on regression and inheritance in the case of two parents,” *Proc R Soc (London)*, 1895.
- [42] T. Hlavacek, P. Jeitner, D. Mirdita, H. Schulmann, and M. Waidner, “Behind the scenes of RPKI,” in *ACM CCS*, 2022.
- [43] T. Hlavacek, H. Schulmann, and M. Waidner, “Smart RPKI validation: Avoiding errors and preventing hijacks,” in *Springer ESORICS*, 2022.
- [44] M. Brandt, H. Schulmann, and M. Waidner, “Evaluating resilience of domains in PKI,” in *ACM CCS*, 2021.
- [45] J. Heitz, “Poison-path routing policy,” US Patent US10015081B2, Jul. 2018, filed Sep. 29, 2016, granted Jul. 3, 2018. Application No. 15/279,480.
- [46] P.-A. Vervier, O. Thonnard, and M. Dacier, “Mind your blocks: On the stealthiness of malicious bgp hijacks,” in *ISOC NDSS*, 2015.
- [47] A. Milolidakis, T. Bühler, K. Wang, M. Chiesa, L. Vanbever, and S. Vissicchio, “On the effectiveness of bgp hijackers that evade public route collectors,” *IEEE Access*, 2023.
- [48] H. Birge-Lee, M. Apostolaki, and J. Rexford, “Global bgp attacks that evade route monitoring,” in *Springer PAM*, 2025.
- [49] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, “To filter or not to filter: Measuring the benefits of registering in the RPKI today,” in *Springer PAM*, 2020.
- [50] T. Hlavacek, A. Herzberg, H. Schulmann, and M. Waidner, “Practical experience: Methodologies for measuring route origin validation,” in *IEEE DSN*, 2018.
- [51] N. Rodday, Í. S. Cunha, R. Bush, E. Katz-Bassett, G. D. Rodosek, T. C. Schmidt, and M. Wählisch, “Revisiting RPKI route origin validation on the data plane,” in *IFIP TMA*, 2021.
- [52] M. Caesar and J. Rexford, “Bgp routing policies in isp networks,” *IEEE Network*, 2005.
- [53] George Eptaminotakis, “Bphs: A bgp prefix hijacking simulation tool supporting rpki filtering,” 2024, accessed Mar. 18, 2025. [Online]. Available: <https://github.com/georgeepta/BGP-Simulator>
- [54] X. A. Dimitropoulos and G. F. Riley, “Efficient large-scale bgp simulations,” *Elsevier Computer Networks*, 2006.
- [55] H. V. Madhyastha, E. Katz-Bassett, T. E. Anderson, A. Krishnamurthy, and A. Venkataramani, “iplane nano: Path prediction for peer-to-peer applications,” in *USENIX NSDI*, 2009.
- [56] X. Li, Z. Cai, B. Hou, N. Liu, F. Liu, and J. Cheng, “Probinfer: Probability-based as path inference from multigraph perspective,” *Elsevier Computer Networks*, 2020.
- [57] Í. Cunha, P. Marchetta, M. Calder, Y. Chiu, B. Schlinker, B. V. A. Machado, A. Pescapè, V. Giotsas, H. V. Madhyastha, and E. Katz-Bassett, “Sibyl: A practical internet route oracle,” in *USENIX NSDI*, 2016.
- [58] N. Tao, X. Chen, and X. Fu, “AS path inference: From complex network perspective,” in *IFIP Networking*, 2015.
- [59] L. Gao, “On inferring autonomous system relationships in the internet,” *IEEE/ACM TON*, 2001.
- [60] Wikipedia, “Globe telecom,” 2025, accessed Mar. 18, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Globe_Telecom
- [61] —, “Singtel,” 2025, accessed Mar. 18, 2025. [Online]. Available: <https://en.wikipedia.org/wiki/Singtel>
- [62] Globe Telecom, “Shareholding structure,” 2025, accessed Mar. 18, 2025. [Online]. Available: <https://www.globe.com.ph/about-us/corporate-governance/shareholding-structure>

APPENDIX A

CASE STUDY: STEALTHY HIJACKING ON 203.127.225.0/24

To verify the discovered stealthy hijacking incident targeting 203.127.225.0/24 (depicted in Figure 5), we manually investigate AS37100’s control-plane visibility and data-plane reachability based on first-hand observations from its looking glass⁴ “lg-01-ams.nl”. All observations presented below were captured on February 10, 2025.

We begin by examining AS37100’s route to the prefix 203.127.0.0/16. As shown in Figure 18, running the command “show ip bgp 203.127.0.0/16” on the looking glass reveals that AS37100 has a route to the prefix announced by the legitimate origin AS3758, via the AS path 37100 6762 6461 7473 3758. This confirms that the legitimate route is visible to AS37100 on the control plane, aligning with our expectations.

Next, we investigate AS37100’s control-plane visibility of the sub-prefix 203.127.225.0/24. As shown in Figure 19, executing “show ip bgp 203.127.225.0/24” on the looking glass returns no matching routes, indicating that AS37100 does not accept the bogus route announcement from the unauthorized origin AS17894. This corroborates our analysis that the bogus route is not visible to the victim on the control plane.

To obtain the actual data-plane forwarding path, we perform traceroute probing from AS37100 to 203.127.225.0/24. As shown in Figure 20, the resulting per-hop data-plane path reveals that the last two hops (line 13 and 14) belong to AS17894. This confirms that traffic from AS37100 to the sub-prefix is indeed diverted to the illegitimate origin, demonstrating actual hijacking at the data-plane level.

Taken together, these observations provide strong evidence of a stealthy BGP hijacking incident, where AS37100’s traffic is misrouted despite its control-plane filtering. However, we emphasize that the intent behind this incident remains uncertain. Given our broader investigation showing that (i) AS17894’s parent organization, *Innove Communications*, is a subsidiary of *Globe Telecom* [60], (ii) AS3758’s parent organization, *SingNet*, is operated by *SingTel* [61], and (iii) *SingTel* is the principal shareholder of *Globe Telecom* [62], we suspect that this incident, despite manifesting as stealthy hijacking, is likely the result of overlooked misconfigurations rather than a deliberate attack. As of this writing, we are awaiting confirmation from *Globe Telecom*. Note that this does not diminish the value of the case study, as misconfigurations can cause stealthy hijacking just like intentional attacks.

APPENDIX B

TOPOLOGY COMPRESSION METHOD

Given that the route inference complexity grows quadratically with topology size, SHAMAN applies a topology compression method to reduce the topology size after obtaining the Internet topology. The compression is based on the concept

⁴<https://lg.seacomnet.com/>


```

SEACOM Locking Glass
Router: lg-01-ama.nl
Command: show ip bgp 203.127.0.0/16

BGP routing table entry for 203.127.0.0/16, version 3804070796
  Path: 12 available, best #2, table default)
  Not advertised to any peer
  Refresh Epoch 1
  37100 6762 6461 7473 3758
  105.26.64.17 from 105.26.64.17 (105.16.0.131)
    Origin 100, metric 0, localpref 100, valid, external
  Community: 37100:1 37100:13
    path 1087100 8981 State valid
    rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  37100 6762 6461 7473 3758
  105.26.64.1 from 105.26.64.1 (105.16.0.131)
    Origin 100, metric 0, localpref 100, valid, external, best
  Community: 37100:1 37100:13
    path 0A3654C 8981 State valid
    rx pathid: 0, tx pathid: 0a0

Disclaimer: All commands will be logged for possible analysis and statistics. If you do not like this policy, please disconnect now.
Queries: peering@seacom.com

```

Fig. 18: AS37100's routes to 203.127.0.0/16.

```

SEACOM Locking Glass
Router: lg-01-ama.nl
Command: show ip bgp 203.127.225.0/24

% Network not in table

Disclaimer: All commands will be logged for possible analysis and statistics. If you do not like this policy, please disconnect now.
Queries: peering@seacom.com

```

Fig. 19: AS37100's routes to 203.127.225.0/24.

of *branch*. Specifically, a branch is defined as a sequence of ASes starting from a stub AS (*i.e.*, an AS with no neighbor other than a single provider) and recursively tracing the subject AS's single provider until encountering an AS with more than one provider, more than one customer, or any peers. This AS, leading to the end of the recursion, is denoted as the *access AS* of the branch. The key intuition behind SHAMAN's topology compression strategy is that all routes from branch ASes to non-branch ASes must traverse consecutive C2P links and pass through the corresponding access AS before entering the broader Internet, and vice versa. As a result, the routing tables of branch ASes can be fast computed based on the access AS's routing table, *e.g.*, by concatenating a certain slice of the branch. Similarly, the routes towards branch ASes can also be fast computed based on the routes towards the access AS. Besides, routes between ASes within the same branch are trivial. Thus, the routes regarding branch ASes can all be fast computed without involving the branch ASes in the BGP route inference process. Therefore, SHAMAN identifies all branches in the Internet topology, establishes their correspondence with access ASes, and prunes the branches to capture the remaining topology, referred to as the core topology. BGP route inference is performed only on the core topology or its sub-topologies, and routes regarding branch ASes are efficiently computed afterwards based on the inference results. In practice, this compression method reduces the topology size by 36.3% in

```

SEACOM Locking Glass
Router: lg-01-ama.nl
Command: traceroute ip 203.127.225.1

Tracing the route to 203.127.225.1
  VRF info: (vrf in name/id, vrf out name/id)
  1 ae-0-21.az-01-mrs.fr.seacomnet.com (105.26.64.1) [AS 37100] 0 msec 200 msec 0 msec
  2 ce-0-0-11.cr-02-mrs.fr.seacomnet.com (105.16.8.209) [AS 37100] MPLS: Label 2242 Exp 0] 204 msec
    ce-0-0-11.cr-02-mrs.fr.seacomnet.com (105.16.8.201) [AS 37100] MPLS: Label 4474 Exp 0] 204 msec
    ce-0-0-11.cr-02-mrs.fr.seacomnet.com (105.16.8.209) [AS 37100] MPLS: Label 2242 Exp 0] 20 msec
  3 ce-0-0-1.az-02-mrs.fr.seacomnet.com (105.16.33.253) [AS 37100] 20 msec
    ce-0-0-1.az-02-mrs.fr.seacomnet.com (105.16.33.253) [AS 37100] 24 msec
    ce-0-0-1.az-02-mrs.fr.seacomnet.com (105.16.33.253) [AS 37100] 20 msec
  4 213.144.184.130 [AS 6762] 24 msec 20 msec 24 msec
  5 213.144.170.125 [AS 6762] 40 msec 44 msec 40 msec
  6 ae10.0.cj01.mrs005.flagtel.com (62.216.131.154) [AS 15412] MPLS: Label 7391 Exp 0] 172 msec 172 msec 168 msec
  7 ae10.0.cj02.slm001.flagtel.com (62.216.129.181) [AS 15412] MPLS: Label 3621 Exp 0] 168 msec 156 msec 156 msec
  8 ae10.0.cj01.slm001.flagtel.com (62.216.137.145) [AS 15412] 160 msec 160 msec 172 msec
  9 80.81.75.186 [AS 15412] 164 msec 164 msec 160 msec
  10 112.198.1.185 [AS 4775] 204 msec 216 msec 204 msec
  11 * * *
  12 120.28.4.38 [AS 4775] 220 msec 220 msec 216 msec
  13 202.126.45.138 [AS 17894] 224 msec
  202.126.45.134 [AS 17894] 220 msec 232 msec
  14 202.126.45.180 [AS 17894] 208 msec 216 msec 224 msec
  15 * * *
  16 * * *
  17 * * *
  18 * * *
  19 * * *
  20 * * *
  21 * * *
  22 * * *
  23 * * *
  24 * * *
  25 * * *
  26 * * *
  27 * * *
  28 * * *
  29 * * *
  30 * * *

Disclaimer: All commands will be logged for possible analysis and statistics. If you do not like this policy, please disconnect now.
Queries: peering@seacom.com

```

Fig. 20: Traceroute from AS37100 to 203.127.225.1.

terms of vertices and 3.97% in terms of edges, with a total of 27,775 branches left out.

APPENDIX C NON-BRANCH ~PL COMPUTATION VALIDITY

The two-branch ~PL field computation described by Equation (6) can be effectively replaced by the branchless computation described by Equation (7) without affecting the integrity of the results. Here, we prove the validity of this replacement.

Theorem 1. Equations (6) and (7) produce identical inference results after each iteration.

Proof. The first branch of Equation (6), where $LP[\hat{p}_{ikj}^{T+1}] \neq (00)_2$, results in $\sim PL[\hat{p}_{ikj}^{T+1}] = \sim PL[p_{kj}^T] - 1$. In comparison, Equation (7) only differs in the term subtracted, *i.e.*, $PL[p_{ik}^1]$ in Equation (7) versus 1 in Equation (6). Thus, we aim to prove $PL[p_{ik}^1] = 1$ under the condition $LP[\hat{p}_{ikj}^{T+1}] \neq (00)_2$. This is evident, as according to Equation (3) and Table III, when $LP[\hat{p}_{ikj}^{T+1}] \neq (00)_2$, it follows that $LP[p_{ik}^1] \neq (00)_2$, indicating an a_i -to- a_k relationship exists. In this case, the a_i -to- a_k route would be a one-hop, ensuring $PL[p_{ik}^1] = 1$.

The second branch of Equation (6), where $LP[\hat{p}_{ikj}^{T+1}] = (00)_2$, results in \hat{p}_{ikj}^{T+1} having an overall value of $(00\ 11111)_2$. Since the best route selection subsequently compares p_{ij}^T with \hat{p}_{ikj}^{T+1} to choose the higher value, and given that the p_{ij}^T is by definition at least $(00\ 11111)_2$, \hat{p}_{ikj}^{T+1} in this case does not affect the result of p_{ij}^{T+1} at the end of this iteration. Similarly, since the output $\sim PL[\hat{p}_{ikj}^{T+1}]$ from Equation (7) will be either $\sim PL[p_{kj}^T] - 1$ or $\sim PL[p_{kj}^T]$, $\hat{p}_{ikj}^{T+1} \leq (00\ 11111)_2$ will hold and will not affect the result of p_{ij}^{T+1} either. Thus, Equations (6) and (7) produce the same result under the condition $LP[\hat{p}_{ikj}^{T+1}] = (00)_2$ as well. \square

APPENDIX D

SIMPLIFYING PRIORITY BYTE SELECTION

We can effectively eliminate p_{ij}^T from the comparison among route priority bytes in each iteration, thereby simplifying Equation (8) to (9). The rationale behind this simplification is that there is always some k such that $\hat{p}_{ikj}^{T+1} \geq p_{ij}^T$. Here, we provide the proof.

Lemma 1. There exist a certain T' ($0 \leq T' \leq T$) and a certain k ($k \in \{0, \dots, n-1\}$) such that $p_{ij}^T = \hat{p}_{ikj}^{T'}$.

Proof. According to Equation (8), p_{ij}^T either equals $\max\{\hat{p}_{ikj}^T\}_{k=0}^{n-1}$ or p_{ij}^{T-1} . If $p_{ij}^T = \max\{\hat{p}_{ikj}^T\}_{k=0}^{n-1}$, apparently there exists a certain k when $T' = T$ such that $p_{ij}^T = \hat{p}_{ikj}^T = \max\{\hat{p}_{ikj}^T\}_{k=0}^{n-1}$. If $p_{ij}^T = p_{ij}^{T-1}$, the problem reduces to finding a certain T' ($0 \leq T' \leq T-1$) and a certain k ($k \in \{0, \dots, n-1\}$) such that $p_{ij}^{T-1} = \hat{p}_{ikj}^{T'}$. We repeat this reduction recursively until the problem reduces to finding a certain T' ($0 \leq T' \leq 1$) and a certain k ($k \in \{0, \dots, n-1\}$) such that $p_{ij}^1 = \hat{p}_{ikj}^1$. Let $T' = 1$ and $k = j$, and then we need prove $p_{ij}^1 = \hat{p}_{ijj}^1 = \max\{\hat{p}_{ikj}^1\}_{k=0}^{n-1}$. Note that \hat{p}_{ikj}^1 is computed based on p_{ik}^1 and p_{kj}^0 . By definition, $p_{kj}^0 = (11 \ 11111)_2$ when $k = j$, otherwise $p_{kj}^0 = (00 \ 11111)_2$. Then, according to the definition of p_{ij}^1 (see Equation (12)) and the computation of \hat{p}_{ikj}^1 (see Equations (3) to (7)), it follows that $p_{ij}^1 = \hat{p}_{ijj}^1 \geq (00 \ 11111)_2$, while all $\hat{p}_{ikj}^1 = (00 \ 11111)_2$ when $k \neq j$. Thus, we have proved the reduced problem, and consequently, the original problem is also proved. \square

Lemma 2. $\hat{p}_{ikj}^{T'} \leq \hat{p}_{ikj}^T$ when $T' < T$.

Proof. \hat{p}_{ikj}^T is computed based on p_{ik}^1 and p_{kj}^T . According to Table III, when p_{ik}^1 is fixed, the LP field of \hat{p}_{ikj}^T either increases or remains unchanged as p_{kj}^T increases. Since, according to Equation (8), $p_{kj}^T \geq p_{kj}^{T'}$ when $T > T'$, it follows that $\text{LP}[\hat{p}_{ikj}^T] \geq \text{LP}[\hat{p}_{ikj}^{T'}]$ when $T > T'$. If $\text{LP}[\hat{p}_{ikj}^T] > \text{LP}[\hat{p}_{ikj}^{T'}]$, it is apparent that $\hat{p}_{ikj}^{T'} \leq \hat{p}_{ikj}^T$, as the LP field represents the two most significant bits of the priority byte. If $\text{LP}[\hat{p}_{ikj}^T] = \text{LP}[\hat{p}_{ikj}^{T'}]$, there are two cases to consider: First, if $p_{kj}^T = p_{kj}^{T'}$, then $p_{ikj}^T = p_{ikj}^{T'}$. Second, if $p_{kj}^T > p_{kj}^{T'}$ but both $\text{LP}[p_{ik}^1] \odot \text{LP}[p_{kj}^T]$ and $\text{LP}[p_{ik}^1] \odot \text{LP}[p_{kj}^{T'}]$ result in $(00)_2$, then according to Equation (6), both p_{ikj}^T and $p_{ikj}^{T'}$ equal $(00 \ 11111)_2$. Therefore, $p_{ikj}^T = p_{ikj}^{T'}$ is ensured when $\text{LP}[\hat{p}_{ikj}^T] = \text{LP}[\hat{p}_{ikj}^{T'}]$. In all cases, $\hat{p}_{ikj}^{T'} \leq \hat{p}_{ikj}^T$ when $T' < T$. \square

Theorem 2. There exists a certain k such that $\hat{p}_{ikj}^{T+1} \geq p_{ij}^T$.

Proof. According to Lemma 1, there exist a certain T' ($0 \leq T' \leq T$) and a certain k ($k \in \{0, \dots, n-1\}$) such that $p_{ij}^T = \hat{p}_{ikj}^{T'}$. Then, according to Lemma 2, $\hat{p}_{ikj}^{T'} \leq \hat{p}_{ikj}^{T'+1}$ since $T' < T+1$. So $\hat{p}_{ikj}^{T'+1} \geq p_{ij}^T$ with this k . \square

APPENDIX E

RESTORING ROUTES FROM NEXT-HOP MATRIX

Here, we describe the process of restoring the complete path of a route between two ASes using the state matrix P^T and the

next-hop matrix N^T . Algorithm 1 outlines the steps to achieve this. The function `RESTOREPATH` takes as inputs the vantage point AS a_i , the origin AS a_j , the state matrix P^T , and the next-hop matrix N^T . It initializes the next-hop index with i (see Line 2) and an empty list to store the path (see Line 3). The algorithm proceeds in a loop, continuously updating the next-hop AS until it reaches the origin AS a_j . If at any point the state matrix indicates that the origin is unreachable (see Line 5), the function returns `None`. Otherwise, it updates the next-hop using the next-hop matrix N^T (see Line 7) and appends the corresponding AS number to the path (see Line 8). The process repeats until the origin AS is reached, at which point the complete path is returned.

Algorithm 1 ensures that we can restore the AS path for any given (i, j) pair based on the information stored in the state and next-hop matrices, providing a systematic approach to generate the complete route set from the matrix-based results.

Algorithm 1 Restoring Path from Next-Hop Matrix

```

1: function RESTOREPATH( $a_i, a_j, P^T, N^T$ )
2:    $next\_hop = i$ 
3:    $path = []$ 
4:   while  $next\_hop \neq j$  do
5:     if  $P^T[next\_hop, j] \leq 0b00111111$  then
6:       return None
7:      $next\_hop = N^T[next\_hop, j]$ 
8:     APPEND( $path, a_{next\_hop}$ )
9:   return  $path$ 
```

APPENDIX F

COMMUNITY ENGAGEMENT

We preliminarily shared our work with APNIC experts, who raised concerns about the practical significance of real-world incidents. In response to this feedback, we carried out the empirical study presented in Section IV, providing quantitative evidence of the prevalence and impact of stealthy hijacking in today's Internet. Additionally, we are actively promoting an informational Internet Draft⁵ that formalizes the mechanism and properties of stealthy hijacking and aims to raise community awareness of this threat.

APPENDIX G

ADDITIONAL TECHNICAL REPORTS

We provide extended technical details covering algorithms, implementation notes, and in-depth analyses in standalone technical reports available via external links. Readers interested in these topics may refer to:

- Report on *Stealthy Hijacking Discovery as a Service*
- Report on *Matrix-Based Route Priority Update*
- Report on *Time-Space Tradeoff in Matrix Update*

⁵<https://datatracker.ietf.org/doc/draft-li-sidrops-stealthy-hijacking/>

A. Description & Requirements

This artifact supports the paper *Understanding the Stealthy BGP Hijacking Risk in the ROV Era*, and enables full reproduction of all experiments and results presented therein. The artifact includes three parts: (i) an empirical study based on real-world BGP data (presented in §IV of the paper), (ii) an analytical study through matrix-based BGP route inference (presented in §VI of the paper), and (iii) a performance evaluation of the proposed analytical framework (presented in §VII of the paper). Additionally, the implementation of our matrix-based BGP route inference algorithm is encapsulated in a Python package named `matrix-bgpsim`⁶, and our service in production is available at <https://yhchen.cn/stealthy-bgp-hijacking>.

Overall, the artifact contains following components:

- Source code and scripts for each part of the experiment.
- Pre-processed datasets (e.g., inferred matrices).
- Pre-computed results and cache files for boosting long-running steps.
- Docker and Conda configurations for environment setup.
- Documentation and README for usage instructions.

To facilitate reproduction and reduce platform dependency, we provide a fully-configured cloud-based evaluation platform (access credentials available via HotCRP). Alternatively, users can run the artifact locally via Docker or by manually setting up the environment with the README instructions.

How to access. DOI 10.5281/zenodo.16565359 or GitHub.

Hardware dependencies. The artifact requires the following resources to complete all experiments:

- At least 120 GB of system free memory,
- At least 60 GB of available disk space,
- Internet access for data downloads, and
- Nvidia GPU and CUDA support for full benchmarking.

Software dependencies. The artifact provides two setups:

- Docker-based: A pre-built Docker image is provided. The user only needs the Docker tool suite installed.
- Manual setup: Detailed scripts are included to set up required packages and tools. These include:
 - Conda environment with Python 3.10 and dependencies listed in “environment.yml”,
 - Node.js for the frontend display, and
 - C/C++ toolchains for building third-party tools, including `bgpdump` and `bgpsim`.

Benchmarks. The artifact relies on several data sources and benchmark components:

- BGP routing data: RIB snapshots from RouteViews collectors *wide*, *amsix*, and *route-views2* since January 1, 2025, and discovered incidents from a two-month period (January to February 2025) of our deployed service.

⁶Stable versions of the package are officially released on <https://pypi.org/project/matrix-bgpsim/> and can be installed via `pip`.

- Intermediate matrices: Pre-computed Internet-scale BGP route matrices, derived from CAIDA AS relationship datasets as of 2025/01/01. These are required for the analytical and performance studies.
- Synthetic topologies: Generated using a sampled Internet AS graph (10,000 ASes) for benchmarking BGP route inference performance.

B. Artifact Installation & Configuration

The artifact supports three installation modes:

Cloud Platform. A fully pre-configured cloud environment is provided for evaluation purposes. Users can simply SSH into the platform (credentials provided via HotCRP) and launch the evaluation environment using a one-line startup script. No additional setup is required.

Docker-based. The artifact includes Docker Compose configurations. Users only need to install the Docker tool suite and download the provided container image and dataset archive. BASH scripts are provided to launch the evaluation environment with all components properly mounted and configured.

Manual Setup. This involves installing required system packages, Python dependencies (via Conda), and third-party tools such as `bgpdump` and `bgpsim`. Detailed instructions and setup scripts are included in the repository.

C. Experiment Workflow

The experiment has three main parts that correspond to the core sections of the paper: an empirical study (§IV), an analytical study (§VI), and performance evaluation (§VII). Each part is self-contained and run via a corresponding “run.sh”

Empirical Study. This part runs a backend routine to analyze real-world BGP routing data and discover stealthy hijacking incidents. Then, it characterizes the discovered incidents and reproduces all relevant figures and tables in §IV. Finally, it sets up a frontend service locally for interactive display of the discovered incidents. A production version of this service is also available at <https://yhchen.cn/stealthy-bgp-hijacking>.

Analytical Study. This part first uses a matrix-based approach to infer BGP routes and quantify the stealthy hijacking risk under partial ROV deployment. It stores the analysis results in intermediate matrices on the disk, and then reproduces all figures and tables in §VI.

Performance Evaluation. This part benchmarks the runtime of our approach against existing tools on synthetic Internet topologies. It then validates the analytical results against the empirical results to evaluate the accuracy of our analytical framework. It further evaluates the accuracy of our analytical framework under ablation of input data sources and statistically analyzes how robust our analytical framework is against mislabels in input data sources. It finally reproduces all figures and tables in §VII.

D. Major Claims

The major claims supported by our artifact are as follows:

- (C1) Stealthy hijacking in the wild is mostly short-lived and targets sub-prefixes, with new cases emerging

almost daily and some persisting long-term. Its exposure is sensitive to vantage point selection.

- (C2) The current partial ROV deployment significantly amplifies stealthy hijacking risk from 0 to a 14.1% overall success probability.
- (C3) Targeted stealthy hijacking achieves near-certain success on specific AS pairs (up to 99.5%).
- (C4) Stealthy hijacking risk mostly opposes the overall risk trend across ASes but is eventually suppressed as ROV’s restrictions on attackers prevail.
- (C5) ASes most effective in launching stealthy hijacking are in Europe, South America, and North America.
- (C6) Cumulative AS hegemony shows the strongest quadratic correlation with stealthy hijacking risk.
- (C7) A small fraction of risk-critical and ROV-enabled ASes account for the majority of stealthy hijacking risk.
- (C8) Validation on real-world datasets shows up to 95.9% incident-level accuracy of our analytical framework.
- (C9) Integrating multiple reliable sources to obtain a more complete view of ROV deployment is crucial to achieve accurate stealthy hijacking risk assessment.
- (C10) Our matrix-based route inference achieves a 500-fold speedup against existing baseline methods.

E. Evaluation

Experiment (E1) [Empirical Study]: discover stealthy hijacking incidents with real-world routing data, reproduce all figures and tables in §IV to support C1, and additionally set up a frontend service to display discovered incidents interactively.

[Preparation] No extra preparation is needed.

[Execution] Running “`empirical-study/run.sh`” is all you need. This script will execute the following steps:

- 1) Run the backend routine that discovers stealthy BGP hijacking incidents using RouteViews RIBs from collectors *wide*, *amsix*, and *route-views2*, each captured at 12:00 January 1, 2025. This is a scaled-down demo for one-day discovery. In actual deployment, we register a cron-job to call this backend routine daily.
- 2) Reproduce all figures and tables in §IV to support C1. It uses all incidents and alarms captured in the first two months of year 2025, which are preserved in the artifact beforehand and are exactly a snapshot of the results by 2025/07/11 from our service in production at <https://yhchen.cn/stealthy-bgp-hijacking>.
- 3) Set up a frontend service to display discovered incidents.

[Results] The discovered incidents will be saved to “`empirical-study/results`”, including one JSON file for alarms and one JSON file for incidents. Figure 2-4 in PDF format and Table II in JSON format are reproduced under the same result directory, which support C1. Once the frontend service is up, it can be accessed at <http://localhost:3000/> using a browser.

Experiment (E2) [Analytical Study]: perform matrix-based BGP route inference using CAIDA AS relationship data, analyze the stealthy BGP hijacking risk, and reproduce all figures and tables in §VI to support C2-C7.

[Preparation] No extra preparation is needed.

[Execution] Running “`analytical-study/run.sh`” is all you need. This script will execute the following steps:

- 1) Run a risk analysis process that infers complete BGP routes on the benign reach and the malicious reach. It then characterizes stealthy hijacking risk based on these inferred routes and stores the results in intermediate matrices on the disk. It further computes various topological features on each AS and cache the results on the disk.
- 2) Reproduce all figures and tables in §VI using the intermediate matrices and AS features generated in the previous step.

[Results] The intermediate results including matrices and AS features will be saved under “`analytical-study/data/matrices`” in the format of compressed pickle objects and CSV files. A JSON file including statistics of routes, Figure 7-13 in PDF format, and Table IV in Tex format are reproduced under “`analytical-study/results`”, which support C2-C7.

Experiment (E3) [Performance Evaluation]: benchmark the runtime of `matrix-bgpsim` and the baseline `bgpsim`, evaluate accuracy and robustness of our analytical framework, and reproduce all figures and tables in §VII to support C8-C10.

[Preparation] No extra preparation is needed.

[Execution] Running “`performance-evaluation/run.sh`” is all you need. This script will execute the following steps:

- 1) Benchmark the runtime of our `matrix-bgpsim` and the baseline `bgpsim`. This first generates a sampled Internet topology that contains 10,000 ASes, based on CAIDA serial-2 AS relationship dataset on 2025/01/01. This process starts with Tier-1 mesh and progressively adds new ASes that connects to the existing topology, until the number of ASes reaches 10,000. Then, it tests how long `matrix-bgpsim` and `bgpsim` generates all routes between any random 10 to 100 ASes, respectively, on the aforementioned sampled topology. `matrix-bgpsim` is also tested under varying number of CPU processes and GPU.
- 2) Use the results from of the previous step and the discovered real-world incidents to evaluate the performance of our framework in terms of accuracy, robustness, and efficiency, and reproduce all figures and tables in §VII.

[Results] The benchmark results are stored in CSV format under “`performance-evaluation/.cache`”. Figure 15-17 in PDF format and Table VI in Tex format are created under “`performance-evaluation/results`”, which support C8-C10.