Forwarding-Plane Anomaly Signals: Prefix Hijacking and Beyond

Yaoqing Liu
Fairleigh Dickinson University
Teaneck, NJ, USA
yliu@fdu.edu

Erdal Kose
Fairleigh Dickinson University
Teaneck, NJ, USA
ekose@fdu.edu

***Abstract*—A prefix hijack redirects live traffic faster than operators can react, but it is the most dangerous member of a broader class: high-impact changes to a network's own forwarding, which also includes route leaks, large re-homes, and withdrawal cascades. Most defenses watch the control plane; we study a complementary forwarding-plane signal that a standalone observer can compute at negligible cost. A shadow aggregation trie over the best-path stream reports, for each update, which forwarding decisions change and whether a long-stable covering block is captured by a previously unseen next hop. The signal is the address space redirected to a novel next hop, weighted by block stability; it is deliberately not churn magnitude, which we show tracks aggregation structure more than significance. Aggregation in fact amplifies such an event: by collapsing the table into a few large, long-stable blocks, it makes a high-impact redirection surface as an outsized change to one prominent block, its size scaling with the space that block had absorbed, rather than dissolving into per-prefix churn. On two independent tier-1 feeds (AS2914 and AS3356; five days of July 2025 updates over million-prefix tables) impact is heavily concentrated and cheap to compute, yet at a transit core two-thirds of next-hop changes reach a never-seen neighbor through benign re-routing, bounding the next-hop signal to an edge- or origin-keyed technique. Even so, gating on impact reduces the raw novelty flood to a few hundred high-impact alerts per day, dominated by large re-homes. Prefix hijacking is our driving case: we motivate it, work the mechanism, report preliminary measurements, and set out validation against historical incidents.**

I. Introduction and Motivation

Border Gateway Protocol (BGP) prefix hijacking, announcing or propagating a route for address space one does not own, can blackhole, intercept, or impersonate large volumes of traffic. The threat is neither rare nor dated. In 2008 a single more-specific announcement by Pakistan Telecom (AS17557) took YouTube offline worldwide for roughly two hours [15]. In 2018 more-specifics of Amazon's Route 53 DNS announced by an Ohio provider (AS10297) redirected users of a cryptocurrency wallet to a phishing site and drained about $150,000 in Ethereum [16]. In 2020 a route leak by Rostelecom (AS12389) redirected roughly 8,800 prefixes from about 200 networks, among them Google, Amazon, and Cloudflare [17]. Twelve years apart, each redirected live traffic before the victim could react. Hijacking is the most consequential instance of a broader phenomenon, an abrupt and high-impact change to a network's own forwarding; route leaks, large re-homes, upstream failures, and withdrawal cascades redirect traffic through the same mechanism, and an operator wants any of them surfaced.

The dominant countermeasures operate on the control plane, watching update streams for anomalous origins or paths. They answer "is *my* prefix being announced elsewhere" well, and speak less directly to a question an operator also cares about: has *my own* forwarding just been redirected in a way that matters, and toward whom? Hijacking is the sharpest instance; the same question covers leaks, re-homes, and failures. We study a signal for that second question that lives in the forwarding information base (FIB) and can be produced by a *standalone* observer that installs nothing. Routers that aggregate their FIB merge adjacent routes sharing a next hop; run an aggregation-shaped structure as a side analysis of the best-path stream and it reports, per update, exactly which forwarding decisions changed and how a covering entry's next hop moved. Contrary to intuition, the useful quantity is not the size of the update burst, which we show tracks aggregation structure more than significance, but the address space actually redirected to a previously unseen next hop, weighted by the stability of the affected block. Aggregation is what makes this signal stand out: by concentrating the address space into a small set of large, long-stable blocks, it turns routine per-prefix churn into a quiet backdrop against which a high-impact event erupts as an outsized disturbance of a single block, its magnitude proportional to how much that block had absorbed. The signal is, in effect, amplified by the very compression the aggregation performs. Because such an event is contained by whoever notices first, a cheap signal an operator can compute on its own feed, without depending on a global monitoring service, is directly useful as an alert on high-impact changes to its own forwarding.

We evaluate on two real, independent tier-1 feeds: AS2914 (NTT) and AS3356 (Lumen). Our contributions are: (i) a standalone forwarding-plane signal for high-impact changes, hijacks and other anomalies alike, with its impact metric and aggregation method (Section II); (ii) a worked example of how bursts arise and why they mislead (Section III); (iii) a two-vantage measurement that establishes a *core-vantage bound* (next-hop novelty collapses under benign churn) yet shows impact gating yields a manageable alert rate (Section IV); and (iv) a plan to validate against documented historical hijacks (Section V).

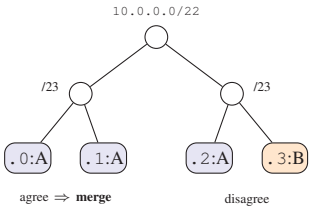


Fig. 1. **Algorithm 1** on four entries `10.0.0.0/24`→A, `.1.0/24`→A, `.2.0/24`→A, `.3.0/24`→B. The agreeing pair `.0`, `.1` merges into `10.0.0.0/23`→A; the disagreeing `.2`, `.3` stay separate. Output FIB (3 entries): {`10.0.0.0/23`:A, `10.0.2.0/24`:A, `10.0.3.0/24`:B}.

II. Detector and Method

Shadow trie. The detector is a separate observer over the same best-path stream the FIB sees; nothing it computes is installed, so it is free to pick the substrate that gives the cleanest signal rather than the smallest table. It maintains a binary radix trie in which a node carries a next hop iff a route for that exact prefix is present, aggregated by the Level-1 rule [2]: merge only where routed space genuinely agrees, with no prevalence step that invents a covering next hop to save space. This strong-correctness, no-prevalence choice ties every covering change to a real change in some address's next hop; size-optimal schemes (ORTC [1], FIFA-S [3]) instead invent covers by majority and so manufacture benign "representation flips," while churn-suppressing schemes [4]–[6] dampen the very changes we watch. Algorithm 1 states the aggregation: a subtree is *uniform to* h if all its routed space resolves to h; agreeing children merge and emit nothing, disagreeing ones emit a covering entry. The covering FIB is $\textsc{Agg}(\text{root}, \perp)$; per update, only the affected root-to-node path and subtree are recomputed, and the emitted set is exactly the forwarding-equivalence delta. Fig. 1 works a four-entry example.

Algorithm 1 $\textsc{Agg}(n, inh)$: Level-1 aggregation, no prevalence

Require: trie node n; inherited next hop inh (root: $inh = \perp$)
Ensure: entries E; subtree next hop u ($u = \perp$ if not uniform)

1: $own \leftarrow n.\text{hop}$ if n carries a route, else inh
2: **if** n has no children **then**
3: **return** $(\emptyset, own)$ if $own{=}inh$, else $(\{(n.\text{pfx}, own)\}, own)$
4: **end if**
5: $(E_0, u_0) \leftarrow \textsc{Agg}(n.\text{child}_0, own)$
6: $(E_1, u_1) \leftarrow \textsc{Agg}(n.\text{child}_1, own)$
7: **if** $u_0 \neq \perp$ **and** $u_0 = u_1$ **then**
8: **return** $(\emptyset, u_0)$ if $u_0{=}inh$, else $(\{(n.\text{pfx}, u_0)\}, u_0)$
9: **end if**
10: $E \leftarrow E_0 \cup E_1$; **if** $own \neq inh$ **then** $E \leftarrow \{(n.\text{pfx}, own)\} \cup E$
11: **return** $(E, \perp)$

Impact, not burst. We define *burst* as the number of covering entries installed, withdrawn, or modified by an update (the FIB write cost). Scoring on burst inflates small hijacks, because a tiny hijack shatters a large block's *representation* while redirecting almost no traffic. The metric is instead the *redirected novel space* R: the number of address cells whose effective next hop actually changed to a next hop never before seen for that space, weighted by the affected block's stability and gated by novelty. Flap between already-seen next hops yields $R{=}0$ and is silenced; a single captured cell yields $R{\approx}1$; capturing a large stable block yields R proportional to the block. Aggregation thus amplifies what matters: the larger and longer-lived the block a capture disturbs, the larger R and the more the event stands out against the blocks that never move. Stability saturates with the affected block's age, so a hijack of a long-coherent aggregate outweighs churn in a freshly formed one, and novelty is tracked per address cell against the next hops seen for that cell so far. Each update recomputes only the affected root-to-node path and subtree, making the footprint exact and cheap; we validated the incremental footprint against full re-aggregation.

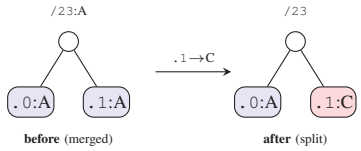


Fig. 2. **Algorithm 2** on update `10.0.1.0/24`→C applied to the FIB of Fig. 1. Only the left `/23` is recomputed; its children now disagree, so it de-aggregates. Footprint: withdraw `/23`:A, install `.0`:A and `.1`:C (burst 3). One `/24` reaches the novel hop C, so $R{=}1$.

On each update, only the affected region is re-aggregated (Algorithm 2); the changed covering entries are the burst, the cells that reach a never-seen next hop give R, and the oldest disturbed block gives the stability weight, all read off the same pass. Fig. 2 continues that example with one update.

Algorithm 2 Per-update footprint and impact

Require: update (p, h); per-cell history of seen next hops
apply (p, h) to the trie; re-aggregate the affected path and subtree
burst ← # covering entries installed / withdrawn / modified
$C \leftarrow$ cells whose effective next hop changed
$R \leftarrow |\{\, c \in C :$ new hop of c never seen for $c \,\}|$
$s \leftarrow \text{stability}(\text{oldest covering block over } C)$
update seen-history; **record** any covering-flip to a novel hop
return impact $R \cdot s$, and burst

III. A Worked Example

Fig. 3 makes the mechanism concrete on a small *constructed* example (real-trace behaviour follows in Section IV). Adjacent `/24`s that all forward to A merge into one covering entry `/20`→ A. A single more-specific `/24`→ H punched into that *merged* block breaks the agreement and de-aggregates the whole path, emitting one covering entry per level (a burst of six), yet only that one `/24` reaches a novel next hop, so $R{=}1$: the burst is large while the impact is tiny.

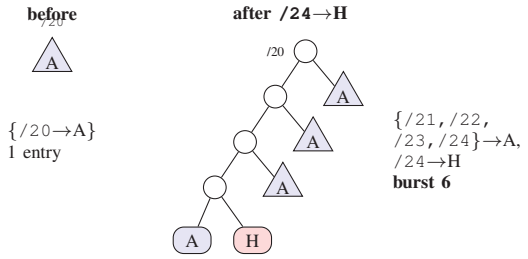


Fig. 3. A more-specific into a *merged* /20 block de-aggregates the whole path, one covering entry per level. Burst grows with block depth (6 for /20, 10 for /16); under an *explicit* /20 route the same hijack adds one exception (burst 1).

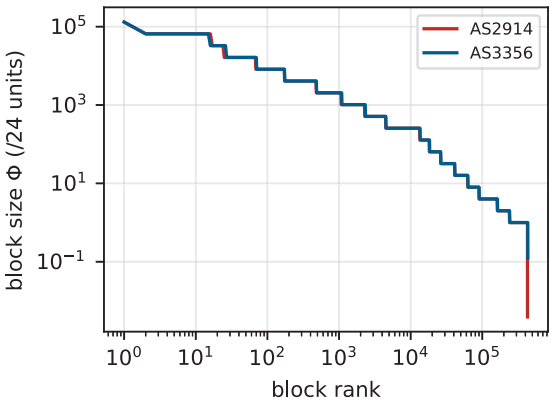


Fig. 4. Block-size distribution of both tables under next-hop aggregation: heavy-tailed and nearly identical, the top 1% of blocks holding ≈47–48% of routed space, so impact concentrates in a few large blocks.

The burst is set by the disturbed structure, not the hijack. Verified against the aggregator, the cascade grows with block depth (4 for a merged /22, 6 for a /20, 10 for a /16); the *same* hijack under an explicitly announced covering route adds only one exception (burst 1), because the base is kept. The largest bursts come from the reverse move: withdrawing a covering less-specific that suppressed k same-next-hop more-specifics re-installs all k at once (burst $k{+}1$), the mechanism behind the real maximum in Section IV. Crucially burst and impact are *inverted*: the cascade bursts six but redirects one /24 ($R{=}1$), while capturing the *whole* block flips one entry (burst 1) yet redirects it all (R equal to the block size). A large burst can carry negligible impact, which is why the detector scores on R, not burst.

IV. Preliminary Results

Dataset. Both feeds come from the University of Oregon *Route Views* project [14], the public BGP collector infrastructure that has peered with operational networks and archived their routing data in MRT format since 1997 and now underpins over a thousand measurement studies. Per vantage it publishes periodic routing-table snapshots (RIBs) and the continuous stream of BGP update messages. We use two full-feed peers, AS2914 (NTT) and AS3356 (Lumen): each a complete IPv4 table (978,045 and 977,564 prefixes) taken from the RIB snapshot at 2025-07-01 00:00 UTC, followed by their live update streams through 2025-07-05. This is real operational routing data as two tier-1 carriers saw it, not a synthetic or emulated workload, and because the archives are public and permanently retained every measurement here is reproducible from the same source. From each announcement we keep the prefix and the next-hop AS the vantage forwards through (the neighbour immediately after the peer on its best path). The RIB is the $t{=}0$ state; we replay the updates in timestamp order, seeding each prefix's set of seen next hops from the RIB. The label is a per-vantage next hop, not the origin, and both vantages are transit cores with large neighbour sets (1,520 and 6,495 distinct next hops).

Impact is concentrated (the metric is well-founded). Aggregating each real table by its next-hop AS yields covering-to-prefix ratios of 0.438 (NTT) and 0.442 (Lumen), in the range classic FIB-aggregation studies report, with strongly heavy-tailed block sizes: the top 1% of blocks hold 47–48% of routed space, a 131,072-/24 block at the head on both feeds (Table II, Fig. 4). Because impact is dominated by a handful of large blocks, a high Φ-weighted score necessarily involves a large block, and the two independent feeds agree on the shape. This concentration is precisely the amplification: because half the routed space sits in the top 1% of blocks, disturbing one is a conspicuous, large-Φ change rather than a needle in a million-prefix haystack, and the de-aggregation it triggers scales with the same Φ.

Novelty collapses at the core (the limit). Replaying the updates, next-hop changes number 411,321 (NTT) and 419,623 (Lumen); of these 66.9% and 68.1% reach a next hop never before seen for that prefix. This is not a cold-start artifact: the daily novel rate *rises* as history accumulates (64% → 78% at NTT, 61% → 81% at Lumen; Fig. 5), because stable prefixes stop moving and a churny residue keeps exploring fresh egresses. The loudest events by impact are benign: large /11–/12 blocks shifting egress among Cogent, TATA, Arelion, and between NTT and Lumen themselves, so impact ranks them at the top although none is an attack. Impact does not imply intent, at scale on real data.

Bursts are small; the rare large ones are benign. The per-update burst is almost always trivial: about half of updates change no covering entry, a third change exactly one, the 99th percentile is five, and the mean is 0.87 (NTT) and 0.83 (Lumen). The tail is thin: over five days only 26 and 17 updates exceed a hundred entries (Fig. 6). The single largest burst is *identical* on both vantages: the withdrawal of 180.96.0.0/11, a China Telecom aggregate, which de-aggregates everything inheriting it and changes 320 (NTT) and 330 (Lumen) entries. That the same benign event tops both independent feeds is the lesson: a large re-routing looks the same everywhere, and burst size alone says nothing about intent.

From flood to alert volume. The novelty rate looks alarming only until it is weighted by impact. About two-thirds

TABLE I. ALERTS PER DAY AS THE IMPACT THRESHOLD RISES (NOVEL NEXT-HOP CHANGES)

alert if redirected space ≥	AS2914 (NTT)	AS3356 (Lumen)
any novel change	∼55,000	∼48,000
/20	∼4,100	∼3,000
/16	∼270	∼215
/12	∼5	∼3

TABLE II. TWO INDEPENDENT TIER-1 VANTAGES, JULY 2025

	AS2914 (NTT)	AS3356 (Lumen)
unique IPv4 prefixes	978,045	977,564
next-hop fanout	1,520	6,495
aggregation ratio	0.438	0.442
top-1% blocks (% space)	47%	48%
next-hop changes	411,321	419,623
reaching a novel hop	**66.9%**	**68.1%**
mean / max burst	0.87 / 320	0.83 / 330

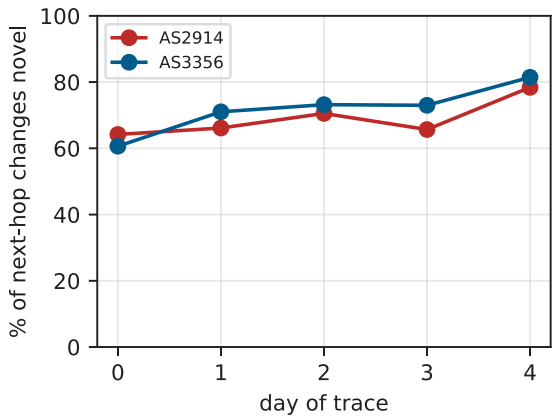


Fig. 5. Daily fraction of next-hop changes reaching a never-seen neighbour. At both cores it is high (≈67%) and *rises* over the trace (benign egress churn, not a cold start), so the novelty gate offers no false-positive relief here.

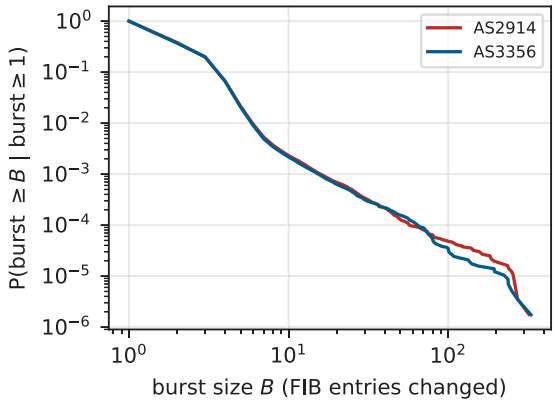


Fig. 6. Complementary CDF of per-update burst size (bursts ≥ 1) on both feeds. Bursts are overwhelmingly tiny (median 1); the single largest, a /11 withdrawal, is the same benign event on both vantages.

of novel changes redirect a single /24, and just ∼0.5% touch a /16 or more, so gating on redirected space turns the raw stream into a rate an operator can watch (Table I): unusable at ∼50k/day for any novel change, but ∼270 (NTT) and ∼215 (Lumen) per day at a /16 threshold and only a handful at a /12. Stability gating trims further, since prefixes that churn three or more times over the window account for 32% (NTT) and 44% (Lumen) of the novel changes. The surviving high-impact alerts are the meaningful ones: on both feeds the largest are /12-scale blocks re-homing among large carriers (Arelion, NTT, Lumen, GTT, TATA), benign here but each a real, large shift in one's own forwarding. Impact is measured nominally (the prefix's own size, an upper bound on effective redirected space) and the thresholds are illustrative absent labelled events; the operative point is the heavy tail, which makes an impact-gated alert stream tractable.

V. DISCUSSION AND NEXT STEPS

A two-sided bound. The measurement bounds the next-hop signal at the core: two-thirds of changes are novel and the loudest are benign transit re-homes, so a next-hop-keyed alarm at a core drowns in false positives. The edge removes that noise but introduces the reverse failure: with a few stable upstreams, a hijack arriving over an *already-used* neighbour never changes the next-hop AS and is silently missed. Next-hop keying thus trades false positives at the core for false negatives at the edge; origin AS, stable at every fanout, is the durable key, but our feeds carry next hop, not origin. The metric itself behaves as designed: it faithfully surfaces high-impact changes, and because aggregation concentrates the table, the heavy tail amplifies them into a small, conspicuous, rankable set. But it cannot infer intent, so it must be combined with origin authorization [11], [12] or cross-vantage evidence to become a detector rather than a change monitor. Although motivated by hijacking, the signal is agnostic to cause: the same covering-flip to a novel next hop over a stable block also marks a route leak, a large legitimate re-home, an upstream failure, a misbehaving route optimizer, or a withdrawal cascade. For an operator that breadth is the point. Each of these silently redirects a large, long-stable slice of its own forwarding, and each is worth an alert; even before intent is settled, a warning that a /12's worth of traffic has begun leaving through a neighbour it has never used is actionable, whether the response is to page security, call an upstream, or record a planned change. The signal is thus best read as cheap, self-hosted situational awareness that ranks events by how much forwarding they move, with the malicious-versus-benign verdict left to a second layer. Prior detection is largely control-plane (PHAS [7], ARTEMIS [8], live feeds [10]) or data-plane probing [9]; we instead repurpose the strong-correctness aggregation delta, treated as a cost in the aggregation literature, as a security observable.

Next steps. The essential next step is to measure detection *effectiveness* against documented historical hijacks rather than only benign behaviour. The three incidents in Section I are

natural test cases: each is preserved in public archives (RouteViews/RIPE RIS [14]), so a vantage that carried it can be replayed through the shadow trie to check whether it raises a covering-flip to a novel next hop, with what impact and at what latency. We will build a labelled corpus from these and from BGPStream and RPKI-invalid events, replay it keyed on origin AS (where our results predict the signal is clean), and report precision, recall, and latency. The targeted more-specifics (YouTube, MyEtherWallet) and the large-scale leak (Rostelecom) sit at opposite ends of the impact scale and so jointly exercise recall and the impact ranking. Two further runs complete the operating range: the same pipeline on a low-fanout edge feed, where we expect the novel rate to collapse, and on origin-keyed multi-vantage data, to test whether cross-vantage disagreement separates a hijack from a legitimate re-home. We will also report the detector's own control-plane cost (trie accesses per update under the incremental algorithm) alongside the near-zero FIB-write burst measured here.

VI. Conclusion

A standalone aggregation-shaped observer turns high-impact changes to a network's own forwarding into a cheap, amplified signal: because aggregation concentrates the table into a few large, long-stable blocks, a hijack, leak, or large re-home surfaces as an outsized, rankable disturbance rather than as per-prefix noise, scored by impact (redirected novel space weighted by stability) rather than churn. On two real tier-1 feeds the promise holds, but so do its limits: next-hop keying bounds the signal to edge or origin-keyed deployment, and it ranks impact, not intent, so a verdict needs a second layer. Impact gating still yields a handful of high-impact alerts per day, and validating detection effectiveness against documented historical incidents is the immediate next step.

References

[1] R. Draves, C. King, S. Venkatachary, and B. Zill, "Constructing optimal IP routing tables," in *Proc. IEEE INFOCOM*, New York, NY, USA, 1999.

[2] X. Zhao, Y. Liu, L. Wang, and B. Zhang, "On the aggregatability of router forwarding tables," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010.

[3] Y. Liu, B. Zhang, and L. Wang, "FIFA: Fast incremental FIB aggregation," in *Proc. IEEE INFOCOM*, Turin, Italy, 2013.

[4] Z. A. Uzmi, M. Nebel, A. Tariq, S. Jawad, R. Chen, A. Shaikh, J. Wang, and P. Francis, "SMALTA: Practical and near-optimal FIB aggregation," in *Proc. 7th ACM Int. Conf. Emerging Networking Experiments and Technologies (CoNEXT)*, Tokyo, Japan, 2011.

[5] Y. Liu and G. Grigoryan, "Toward incremental FIB aggregation with quick selections (FAQS)," arXiv preprint arXiv:1812.05520, 2018. [Online]. Available: https://arxiv.org/abs/1812.05520

[6] N. Sarrar, R. Wuttke, S. Schmid, M. Bienkowski, and S. Uhlig, "Leveraging locality for FIB aggregation," in *Proc. IEEE GLOBECOM*, Austin, TX, USA, 2014.

[7] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system," in *Proc. 15th USENIX Security Symp.*, Vancouver, BC, Canada, 2006.

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

[9] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP prefix hijacking on my own," in *Proc. ACM SIGCOMM*, Seattle, WA, USA, 2008.

[10] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti, "BGPStream: A software framework for live and historical BGP data analysis," in *Proc. ACM Internet Measurement Conf. (IMC)*, Santa Monica, CA, USA, 2016.

[11] M. Lepinski and S. Kent, "An infrastructure to support secure Internet routing," IETF, RFC 6480, Feb. 2012. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6480

[12] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of BGP multiple origin AS (MOAS) conflicts," in *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement (IMW)*, San Francisco, CA, USA, 2001.

[13] A. Mitseva, A. Panchenko, and T. Engel, "The state of affairs in BGP security: A survey of attacks and defenses," *Computer Communications*, Elsevier, 2018.

[14] University of Oregon, "The Route Views Project." [Online]. Available: http://www.routeviews.org

[15] RIPE NCC, "YouTube hijacking: A RIPE NCC RIS case study," Mar. 2008. [Online]. Available: https://www.ripe.net/news/study-youtube-hijacking.html

[16] L. Poinsignon, "BGP leaks and cryptocurrencies," The Cloudflare Blog, Apr. 24, 2018. [Online]. Available: https://blog.cloudflare.com/bgp-leaks-and-crypto-currencies/

[17] Qrator Labs, "This is how you deal with route leaks," Apr. 2, 2020. [Online]. Available: https://qrator.net/blog/details/how-you-deal-route-leaks