

Post-Quantum Authentication for BGPsec/RPKI, SRv6, and Agentic AI NetOps in Carrier Networks

Jaypal Baviskar · Technical Leader, Cisco CX, Service Provider Technology Group (SPTG) · jay.22kar@gmail.com

Afshan Mulla · Technical Leader, Cisco CX, Professional Services · aafshan.mulla@gmail.com

Position paper- IAB Workshop on Accelerating the Deployment of Post-Quantum Authentication (pqws), Prague, 11–12 October 2026

Position. The carrier control plane - BGPsec/RPKI route validation, SRv6 network programming, and increasingly autonomous agentic AI NetOps actions - is, in our view, among the least-examined post-quantum authentication deployment surfaces, because most public discussion is framed around TLS and the Web PKI; early RPKI-specific work has only recently appeared [22], [23]. Three things combine.

First, ML-DSA [1] and SLH-DSA [2] signatures are roughly 10x to several hundred times larger than the RSA and ECDSA signatures they replace, and carrier infrastructure - tight convergence and fast-reroute budgets, hard per-packet MTU ceilings, embedded route-processor crypto - was not built around that. Adoption is still early: a 2025 survey of 1,042 security leaders found only 5% had deployed any quantum-safe encryption [10].

Second, each surface hits a different wall for the same underlying reason: BGPsec/RPKI signs per hop and per signed object, SRv6 steers every packet with 128-bit segment identifiers, and agentic NetOps needs an action to be verifiable offline, potentially years later, by an auditor who was never in the session.

Third, all three yield to one design principle, applied through different mechanisms: aggregate the signature, don't multiply it.

This paper shows that is not just a proposal. Using today's global RPKI dataset as a realistic scale proxy - from one up to 996,690 validated ROA payloads (VRPs) - a single ML-DSA-65-signed Merkle root replaces 996,690 individual signatures with one, with mean root-signing time flat at ~9–11 ms regardless of batch size and a per-object verification bundle (signature plus inclusion proof) of 3,969 B (Table 1). No single RPKI signer holds all VRPs: in deployment a tree would be built per CA, or by a validator signing the VRP set it distributes, so the gain per tree depends on how many objects each signer holds. We propose extending the same aggregate-don't-multiply principle to SRv6 and agentic NetOps, and raise where each needs the workshop's input.

1. How carrier authentication actually works, and why it is different

RPKI [7] signs every certificate, ROA and manifest, and each signed object such as a ROA or manifest carries two signatures: one on its embedded end-entity certificate and one on its CMS content [17]. Today these use RSA-2048 (256 B signatures) [11], which ML-DSA-65 (3,309 B) inflates ~13x, and relying parties re-validate the whole repository on every sync. BGPsec [6] additionally signs each UPDATE at every AS hop with ECDSA P-256 [16] (~70–100 B per hop); ML-DSA-65 makes that ~3.3 KB per hop. A short 5-hop path then carries ~16.5 KB of signatures, past the classic 4,096-byte UPDATE limit and into extended messages [8], and 20 or more signing hops exceed even the 65,535-byte extended-message ceiling. BGPsec itself is barely deployed - only two BGPsec public keys were available in the RPKI as of October 2025 [25] - but any per-hop public-key path-validation scheme hits this wall.

SRv6 [9] encodes each segment as a 128-bit identifier, carried in the IPv6 destination address and, for multi-segment paths, in a Segment Routing Header [12]; compressed SIDs [19] shrink this but do not remove it. There is no way to attach a multi-kilobyte post-quantum signature per packet without breaking a 1,500-byte Ethernet MTU. SRv6 is also designed to run inside a trusted SR domain [12], so the post-quantum exposure lies mainly in the control plane: the controller-to-head-end policy channel and the keys that protect it.

Agentic NetOps - autonomous AI agents proposing or directly executing configuration and remediation actions with no per-action human approval - raises a different question: not just which key holder signs an action, but whether that action can be proven, offline, to an auditor who was never in the original session. This is the same long-lived-verification problem this workshop's call already scopes in for firmware and software signing, applied instead to AI-issued network changes.

2. What a post-quantum Merkle-aggregated attestation costs

Key and signature sizes are from FIPS 204 [1]. We built an ML-DSA-65 Merkle-tree aggregation prototype on liboqs/oqs-provider and measured it against a real RPKI VRP dataset (~1M validated ROA payloads, fetched live from RPKI publication points via Routinator), at batch sizes from $n=1$ to $n=996,690$. Every run - 50 total - independently verified both the root signature and the generated inclusion proof before being recorded. Signing times are means as measured by the prototype; verification time, signing-time variance and tree-build time were not measured and are part of question 1.

Batch size (n)	Root sign time	Per-object bundle	Classical total (all leaves signed)	Reduction factor (bound)
1	~10.7 ms	3,309 B	3,309 B	1.0x
100	~9.9 ms	3,540 B	330,900 B	93.5x
10,000	~9.0 ms	3,771 B	33,090,000 B	8,774.9x
100,000	~9.9 ms	3,870 B	330,900,000 B	85,503.9x
996,690	~10.0 ms	3,969 B	3,298,047,210 B	830,951.7x

Table 1. ML-DSA-65 Merkle-tree aggregation vs. one signature per VRP, measured against a real RPKI VRP dataset. The per-object bundle grows by exactly 33 B per tree level ($\lceil \log_2 n \rceil$ levels). The reduction factor divides the total for n individual signatures by one bundle, so it bounds repository-wide signature volume, not per-object verification cost.

- **Signing cost is flat, not growing.** Root-signing time stays ~9–11 ms across six orders of magnitude of batch size, because the mechanism always signs one 32-byte root - never once per object.
- **Signatures collapse from n to one.** The reduction factor grows from 93.5x at $n = 100$ to 830,951.7x at $n = 996,690$, because individual signing grows as $O(n)$ while each object's proof grows only as $O(\log n)$ (20 levels, 660 B at full scale). Larger trees benefit more, within one signing domain. By calculation, shipping a proof with every object still cuts signature-and-proof bytes ~5x (~3.3 GB to ~0.66 GB). A relying party holding the complete signed set can instead recompute the root and needs no proofs, but then one missing or stale object blocks verification of the whole batch, unlike RPKI's per-object failure isolation.
- **Tree build scales linearly with n .** It needs $2n-1$ hash operations per build, once per publication cycle rather than per change; we did not time it separately.

3. Where the limits sit in a carrier control plane

Where	What it is	Can it be changed?	What it means
BGPsec / RPKI signatures	Per-hop (BGPsec) and per-object (RPKI) signing; ML-DSA-65 is ~3.3 KB per signature vs. 256 B (RSA-2048, RPKI) or ~70 B (ECDSA, BGPsec)	Yes - aggregate	One signature per tree instead of one per object (Table 1)

Where	What it is	Can it be changed?	What it means
SRv6 SID / SRH	128-bit segment IDs steer every packet; no room for a multi-KB signature	No - that is SRv6's point	Signing must move to the control plane, not the packet
Embedded route-processor crypto	HSM / crypto-accelerator capacity sized for classical signatures	Yes, over a hardware refresh cycle	Carrier-scale signing may need firmware or hardware updates
Agentic NetOps actions	Autonomous AI-issued config/remediation actions, no per-action human approval	Architectural	Needs long-lived offline verification, the same problem as firmware signing
Classical/PQC coexistence	Multi-year migration; mixed AS_PATH hops during transition	Yes - the open question	No consensus yet on hybrid validation duration or fallback rules

Table 2. Carrier-specific pressure points. The last row is the open migration question this workshop is well placed to help resolve.

4. What we propose

(1) Aggregate RPKI (and any future path-validation) attestations into a Merkle tree. This follows Merkle Tree Certificates [3] and RPKI manifests [18], which already sign one list of object hashes per CA; pqRPKI [22] develops a related Merkle-based design inside manifests. A relying party verifies one signed root plus a short inclusion proof - or recomputes the root from the full set it already fetches - instead of one full signature per object or hop. Trees can be built per CA, or by a validator signing the VRP set it serves to routers, which today is protected, if at all, only by transport security [21].

(2) Sign SRv6 policies once, at the controller, never per packet. Distribute the signed policy through BGP SR Policy [13], PCEP [20] or another controller channel, and have the data plane carry only a short symmetric token derived from it, in the spirit of the SRH HMAC TLV [12]; symmetric MACs already resist quantum attack, so the open problems are key distribution and how long the token should remain valid on a node that cannot always reach the controller.

(3) Wrap agentic NetOps actions in a compact, JOSE/COSE-based, post-quantum-signed envelope (e.g. COSE_Sign1 with ML-DSA [14], every security-relevant field in the signed protected header) binding each action to the issuing agent's identity and authorization scope, independently re-verifiable offline by an auditor outside the original session. Agent identities could be batch-issued post-quantum credentials such as MTAC [24], and append-only action logs with receipts fit SCITT [15]; authorization-scope semantics are policy-layer and left generic here.

5. Questions for the workshop, and where the work belongs

- What is the end-to-end impact of Merkle-aggregated RPKI attestation on relying-party validation time and publication-to-router latency under realistic ROA churn, beyond the signing-cost numbers already measured here?
- For SRv6 cached attestation, what is an operationally sound validity window for a controller-unreachable mid-path node, and how should it interact with fast-reroute timers?
- Is there appetite, inside or outside the IETF, for a lightweight, JOSE/COSE-based action-attestation envelope for autonomous network changes, distinct from existing API tokens?
- What HSM and embedded-route-processor capability gaps most block carrier-scale post-quantum control-plane signing today, and which are solvable in firmware versus requiring a hardware refresh?
- Must a carrier control plane support mixed classical and post-quantum validation indefinitely during migration, and what does that do to relying-party software and composite certificate handling [4]?

Where the follow-up belongs. There is no single IETF working group that owns carrier control-plane post-quantum authentication end to end. The pieces do have homes: Merkle Tree Certificates sit with PLANTS [3] and composite signatures with LAMPS [4]; RPKI operational practice sits with SIDROPS, where post-quantum RPKI work has begun [23]; BGP mechanics sit with IDR and SRv6 with SPRING; signed, transparent per-action records map naturally onto SCITT [15]; and KEM-based authentication [5] may suit interactive control channels such as controller-to-router sessions. So the carrier control plane, as a deployment surface, needs one of them to start.

References

- [1] FIPS 204, Module-Lattice-Based Digital Signature Standard, NIST 2024.
- [2] FIPS 205, Stateless Hash-Based Digital Signature Standard, NIST 2024.
- [3] draft-ietf-plants-merkle-tree-certs-07, Merkle Tree Certificates, IETF PLANTS WG, October 2026, work in progress.
- [4] draft-ietf-lamps-pq-composite-sigs, Composite ML-DSA for use in X.509 Public Key Infrastructure, IETF LAMPS WG, in RFC Editor queue.
- [5] draft-celi-wiggers-tls-authkem-07, KEM-based Authentication for TLS 1.3, May 2026, work in progress.
- [6] RFC 8205, BGPsec Protocol Specification.
- [7] RFC 6480, An Infrastructure to Support Secure Internet Routing (RPKI).
- [8] RFC 8654, Extended Message Support for BGP.
- [9] RFC 8986, Segment Routing over IPv6 (SRv6) Network Programming.
- [10] DigiCert, 2025 Quantum Readiness Study (Propeller Insights survey of 1,042 security leaders), May 2025.
- [11] RFC 7935, The Profile for Algorithms and Key Sizes for Use in the Resource Public Key Infrastructure.
- [12] RFC 8754, IPv6 Segment Routing Header (SRH).
- [13] RFC 9830, Advertising Segment Routing Policies in BGP.
- [14] RFC 9964, ML-DSA for JOSE and COSE.
- [15] RFC 9943, An Architecture for Trustworthy and Transparent Digital Supply Chains (SCITT).
- [16] RFC 8608, BGPsec Algorithms, Key Formats, and Signature Formats.
- [17] RFC 6488, Signed Object Template for the Resource Public Key Infrastructure (RPKI).
- [18] RFC 9286, Manifests for the Resource Public Key Infrastructure (RPKI).
- [19] RFC 9800, Compressed SRv6 Segment List Encoding.
- [20] RFC 9603, Path Computation Element Communication Protocol (PCEP) Extensions for IPv6 Segment Routing.
- [21] RFC 8210, The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1.
- [22] W. Li, Y. Li, T. Chung, pqRPKI: A Practical RPKI Architecture for the Post-Quantum Era, arXiv:2603.06968, March 2026.
- [23] T. Yoshikawa, L. Velindron, draft-yoshikawa-sidrops-pqc-rpki-02, Post-Quantum Signature Experiments and Migration Considerations for the Resource Public Key Infrastructure (RPKI), August 2026, work in progress.
- [24] R. Burls, draft-burls-mtac-00, Merkle Tree Agent Certificates (MTAC): Batch-Issued Post-Quantum Identity Credentials for AI Agents, July 2026, work in progress.
- [25] IETF 124 SIDROPS, Transitive BGPsec is Difficult/Infeasible (slides citing rpki-client data: two BGPsec public keys as of 28 October 2025), November 2025.