

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) · jbaviska@cisco.com

Afshan Mulla · Technical Leader, Cisco CX, Business Critical Services (BCS) · afsmulla@cisco.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 the least-examined post-quantum authentication deployment surface, because nearly all public discussion is framed around TLS and the Web PKI. Three things combine.

First, ML-DSA and SLH-DSA signatures are one to two orders of magnitude larger than the classical mechanisms they replace, and carrier infrastructure - sub-second BGP convergence, hard per-packet MTU ceilings, embedded route-processor crypto - was not built around that.

Second, each surface hits a different wall for the same underlying reason: BGPsec/RPKI signs per hop and per certificate, SRv6 carries a fixed 16-byte segment identifier in every packet, 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 can be addressed the same way: aggregate the signature, don't multiply it.

This paper shows that is not just a proposal. Aggregating real RPKI certificates - from a single certificate up to 996,690, the practical scale of today's global RPKI dataset - into a single ML-DSA-65-signed Merkle root reduces total signature overhead by up to 830,951.7x versus signing each certificate individually, with signing time flat at 9–12 ms regardless of batch size (Table 1). 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

BGPsec (RFC 8205) and RPKI (RFC 6480) validate routes through per-hop and per-certificate signatures, checked on every route change across a full-table BGP session. An ECDSA signature adds ~70–100 B per AS hop; an ML-DSA-65 signature adds ~3.3 KB, and a 30–100-hop AS_PATH multiplies that linearly, pushing past the classic 4,096-byte BGP UPDATE limit into RFC 8654's extended message support.

SRv6 (RFC 8986) carries a 16-byte segment identifier inside a Segment Routing Header in every packet. There is no way to attach a multi-kilobyte post-quantum signature per packet without breaking a 1,500-byte Ethernet MTU, so authentication has to move to the control plane rather than the data plane.

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 signed 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 RIR repositories 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.

Batch size (n)	Root sign time	Aggregated total	Classical total (all leaves signed)	Reduction factor
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. individual per-certificate signing, measured against a real RPKI dataset.

- **Signing cost is flat, not growing.** Root-signing time stays 9–11 ms across seven orders of magnitude of batch size, because the mechanism always signs one 32-byte root - never once per certificate.
- **The reduction factor grows with scale, not shrinks.** 93.5x at 100 certificates, climbing to 830,951.7x at 996,690 - the expected O(n) vs. O(log n) relationship between classical and Merkle-aggregated signing. Larger repositories benefit more, not less.
- **Tree-build time is the only cost that scales with n, and it stays small.** Even at nearly one million real certificates, building the tree takes well under a second; the constant is the flat signing cost above.

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 and per-certificate signing; ML-DSA-65 adds ~3.3 KB per signature vs. ~70–100 B for ECDSA	Yes - aggregate	Measured 830,951.7x reduction at real dataset scale (Table 1)
SRv6 SID / SRH	16-byte segment ID carried in 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 BGPsec/RPKI attestations into a Merkle tree, in the spirit of Merkle Tree Certificates [3] already referenced in this workshop's scope.** A relying party verifies one compact root plus a short inclusion proof, instead of one full signature per hop or certificate - the mechanism measured in Table 1.
- (2) **Sign SRv6 policies once, at the controller, never per packet.** Distribute the signed policy through the existing BGP-LS/PCEP or controller channel, and have the data plane carry only a short cached validity token - open question: how long that 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** binding each action to the issuing agent's identity and authorization scope, independently re-verifiable offline by an auditor outside the original session. The underlying trust-scoring mechanism is kept generic here; it is the subject of separate, ongoing work.

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

1. What is the measured convergence-time impact of Merkle-aggregated BGPsec/RPKI attestation on a full-table eBGP session, beyond the signing-cost numbers already measured here?
2. 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?
3. 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?
4. 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?
5. 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 who owns carrier control-plane post-quantum authentication end to end. The pieces do have homes: Merkle Tree Certificates and composite signatures sit with LAMPS and PLANTS, RPKI operational practice sits with SIDROPS, BGP mechanics sit with IDR, SRv6 sits with SPRING, and a per-action attestation telemetry model sits with OPSAWG. 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, Merkle Tree Certificates, IETF PLANTS WG, work in progress.
- [4] draft-ietf-lamps-pq-composite-sigs, Composite ML-DSA for X.509, IETF LAMPS WG, work in progress.
- [5] draft-celi-wiggers-tls-authkem, KEM-Based Authentication for TLS 1.3, 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, Quantum Readiness Outlook (Propeller Insights survey), May 2025.