July 20th, 2025

Adapting Constrained Devices for Post-Quantum Cryptography

Status Update

Authors: *T. Reddy.K, D. Wing, Ben S, K. Kwiatkowski*

Presenter: *Kris Kwiatkowski*

Draft Evolution

- Original draft: `draft-reddy-pquip-pqc-hsm-00` (IETF 122, Bangkok)
    - Now adopted as WG item: `draft-ietf-pquip-pqc-hsm-constrained-01`
- Title and scope updated to cover:
    - resource-constrained devices (RoT, HSMs, etc.)
- New content and restructured layout for clarity

Why This Draft Matters

- Constrained secure hardware is key to IoT, industrial, automotive
  - Typically 8-16KB RAM (see `draft-ietf-iotops-7228bis` for detailed classification)
- Post-quantum algorithms require more RAM, CPU, storage and power

Memory usage [KB]

	Keygen	Sign	Verify
EdDSA	7.5	7.5	3
ML-DSA-65 (small)	4.5	**11.8**	4.6
ML-DSA-65 (fast)	60.8	**68.8**	9.8
ML-KEM-768 (small)	9	9.1	12.8
ML-KEM-768 (ref)	19.6	19.9	26.7

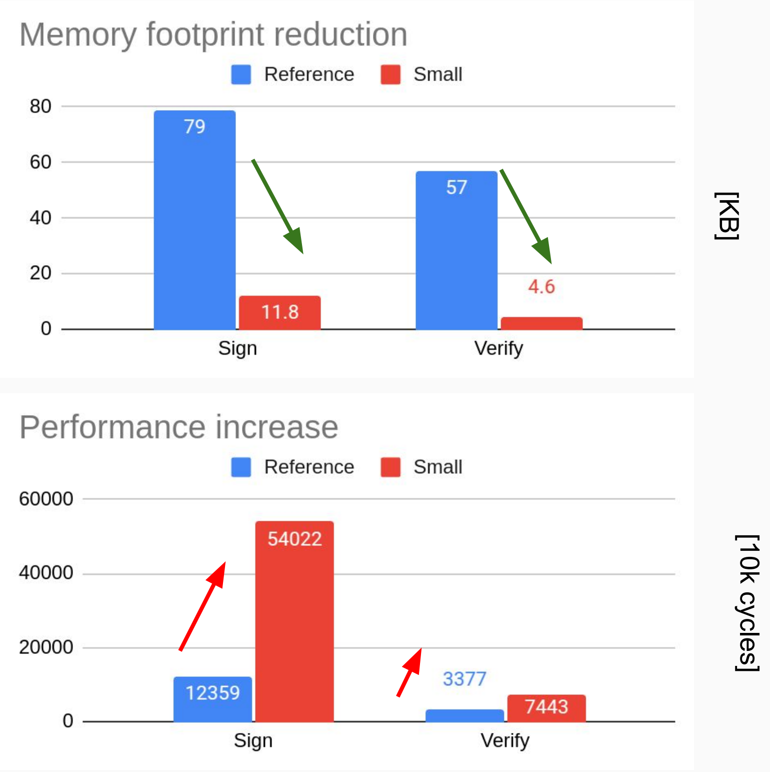
Performance (10k cycles, Cortex-M4)

	Keygen	Sign or Encaps	Verify or Decpas
ML-DSA-65 (small)	3412	**24421**	5732
ML-DSA-65 (fast)	2516	**6193**	2415
ML-KEM-768 (portable C)	988	1138	1387
ML-KEM-768 (small-opt)	644	664	714

Why This Draft Matters

- Real-world adoption needs
  - Practical implementation guidance
  - Trade-off analysis
  - Lifecycle management strategies

- The key challenge for PQ deployments is to finding the right balance between scheme, application and implementation technique



ML-DSA-65

draft-ietf-pquip-pqc-hsm-constrained-01

Focus Areas

- Algorithm selection & optimization for constrained systems
  - RAM/CPU/storage efficiency
- Key generation and usage lifecycle
  - Secret key expansion from the seed
- Post-Quantum Firmware authentication and updates
  - Impact on secure boot
- Power utilization

What’s New in Version -01

- **Renamed** draft to make a scope clearer
- **Pre-hashing interface** emphasized for performance
- **Firmware signing** section revised with concrete suggestions
- **RAM usage guidance** added
- **Algorithm comparison:** SLH-DSA vs ML-DSA vs HSS-LMS
- **Ephemeral key** lifecycle guidance added

Key Technical Guidance

Seed-Based Key Generation

- Store only the seed; derive keys on-the-fly

Impact of PQC Authentication on Constrained Devices

- For example, usage of ML-DSA may require significant amount of RAM
- This affects: TLS session establishment, firmware authentication
- Understanding RAM profiles per algorithm, design that balances memory constraints and performance

Pre-hash Interface

- Reduces data transfer and memory use

Firmware Signing Considerations

Recommended

- HSS-LMS, XMSS, SLH-DSA for long-term authentication
- ML-DSA only when speed outweighs future-proofing

Discussion around boot-time RAM constraints during validation

- Use seed-based keys where flash is tight but CPU is acceptable

Next Steps

- Discussion on side-channel security
- Concentrate on practical implementation issues
  - Design strategies for devices with constrained memory
  - Performance trade-offs
- Impact of new schemes and/or new parameterizations (SLH-DSA with smaller params, FN-DSA, HAWK…)
- Update document structure to clearly reflect revisited scope

Call for Feedback

- Please review and comment on the draft
- Share deployment insights, use cases, or benchmarks
- Suggest reference implementations
    - Maybe good topic for hackathon