

7th **INTERNATIONAL CONFERENCE ON** **PUBLIC KEY INFRASTRUCTURE AND ITS** **APPLICATIONS (PKIA 2026)**

SEPTEMBER 10-11th, 2026

Design and Comparative Evaluation of a Hybrid Post-Quantum Key Exchange Protocol for Quantum-Resistant Onion Routing Networks

Subhash Kumar | MIT-WPU, Pune
Dr. Kishor R. Kolhe | MIT-WPU, Pune

Outline

01The problem

Harvest-Now-Decrypt-Later and the anonymity-specific risk

02The constraint

ML-KEM key material against Tor's fixed 509-byte cell

03Contributions

What we built, and how each number in the talk is grounded

04Protocol design

Hybrid key schedule, combiner, and why degradation is safe

05Fragmentation

Reassembly DoS, atomicity under loss, traffic-analysis burst

06Measured cost

Per-operation timings against the X25519 baseline

07Composed cost

Per-hop compute against deterministic cell counts

08Where it lands

Circuit-build breakdown, memory footprint, positioning

09Hardware

ARM64/NEON cross-build and a synthesised FPGA datapath

10Conclusion

What we found, and what remains open

Introduction

WHAT ONION ROUTING RESTS ON

- Tor protects millions of daily users by relaying encrypted traffic through volunteer relays, so no single relay learns both the origin and the destination
- The security of that construction rests almost entirely on public-key cryptography: every circuit layer is negotiated with the ntor handshake over Curve25519
- Confidentiality of every relayed cell ultimately depends on the secrecy of the keys those handshakes produce

WHY THE EXPOSURE IS SPECIFIC

- Shor's algorithm on a fault-tolerant quantum computer solves the elliptic-curve discrete logarithm problem, recovering the ephemeral keys of every recorded ntor handshake
- Resource estimates keep lowering the projected cost of that attack, which is why standards bodies began migrating well ahead of any demonstrated machine

THE ASYMMETRY

For most Internet protocols the urgency of post-quantum migration is debatable. For an anonymity network it is not.

The value Tor provides is the unlinkability of a user from their activity, and that property is supposed to hold not only during a session but indefinitely afterwards.

This is why the Tor Project has treated post-quantum confidentiality of the circuit-extension handshake as a migration priority, while post-quantum authentication can reasonably be deferred.

Motivation: Harvest-Now-Decrypt-Later

WHY AN ANONYMITY NETWORK CANNOT WAIT FOR A QUANTUM COMPUTER TO EXIST

1 RECORD

A passive AS, IXP or state actor archives circuit-setup cells today



2 STORE

Retention is cheap. The archive simply waits, for years if needed



3 BREAK

A CRQC runs Shor and solves the ECDLP behind Curve25519



4 DE-ANONYMISE

Every recorded ntor handshake, and the traffic under it, opens

Why anonymity is the acute case

- Unlinkability is supposed to hold indefinitely, not merely for the duration of a session
- A source, a dissident or a whistleblower is not made safe by the passage of a decade
- Retrospective decryption does not expose stale data; it retroactively destroys the guarantee the user relied on

What follows for scope

- The circuit-extension handshake needs post-quantum confidentiality now
- Post-quantum authentication can be deferred: live impersonation needs a quantum computer at handshake time
- HNDL is defeated by confidentiality alone, this is transitional security

Literature Survey

APPENDIX · PAPER SECTION II-B, II-E, VII-E

Work	Primitive	Contribution	Gap this paper addresses
Ghosh & Kate, ACNS 2015	ring-LWE	PQ forward-secure onion routing; lattice-augmented one-way AKE	Pre-standard primitive; no fragmentation treatment
Schanck, Whyte, Zhang · (prop. 269 / PoPETs 2016)	modular KEM	Transitionally secure hybrid; safe degradation to plain ntor	Framework only no measured integration
Lovecraft & Schwabe · (proposal 270)	concrete choices	“RebelAlliance” hybrid handshake	Superseded by the finalised NIST standards
Tor Project (proposal 355)	ML-KEM / ML-DSA	Revisits PQ circuit handshakes post-standardisation	Design direction, not an evaluation
ePrint 2025/479	ML-KEM	PQ migration of Tor; $\approx 1.25\text{--}1.28\times$ handshake processing	Aggregate cost; compute and wire not separated
Giacon–Heuer–Poettering (PKC 2018) · Bindel et al. (PQCrypto 2019)	KEM combiners	IND-CCA security of concatenation combiners; hybrid AKE	Security foundation this design inherits
IETF draft-ietf-tls-ecdhe-mlkem	X25519MLKEM768	Deployed hybrid combiner for TLS 1.3	The construction we mirror, for the same reason

ALGORITHM DIVERSITY · § II-E

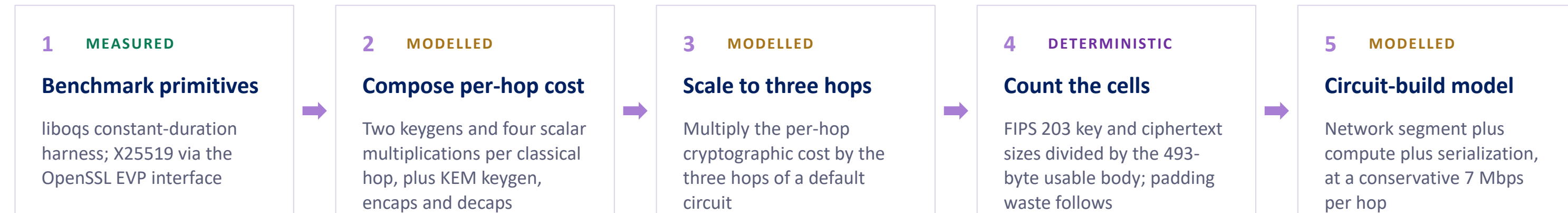
FALCON (FN-DSA, FIPS 206 draft) is the compact signature option at roughly 666 bytes against ML-DSA-44's 2,420. HQC, selected in March 2025, is a code-based backup structurally independent of lattices.

WHAT IS NEW HERE

Not the abstract notion of a hybrid handshake. The contribution is a measured end-to-end integration using standardised ML-KEM, with the fragmentation cost separated from the computational one.

Methodology

APPENDIX • PAPER SECTION V–VI



Not yet run. Step 6 is the Shadow discrete-event simulator running patched Tor over a simulated topology. The configuration ships as an artefact; the analytical model is designed to be superseded by it.

MEASUREMENT ENVIRONMENT

One Intel Xeon core at 2.80 GHz, Ubuntu 24.04, gcc 13.3.0. liboqs 0.15.0 in release mode with AVX2 and AVX-512 active; X25519 via OpenSSL 3.0.13. Five-second windows per KEM operation; X25519 averaged over 200,000 iterations.

FAIR BASELINE

The classical comparison is the same code path with the KEM operations and fragmentation disabled, so differences are attributable to the hybrid components. Opportunistic KEM keypair caching is switched off, so every handshake is a full keygen encapsulate-decapsulate cycle.

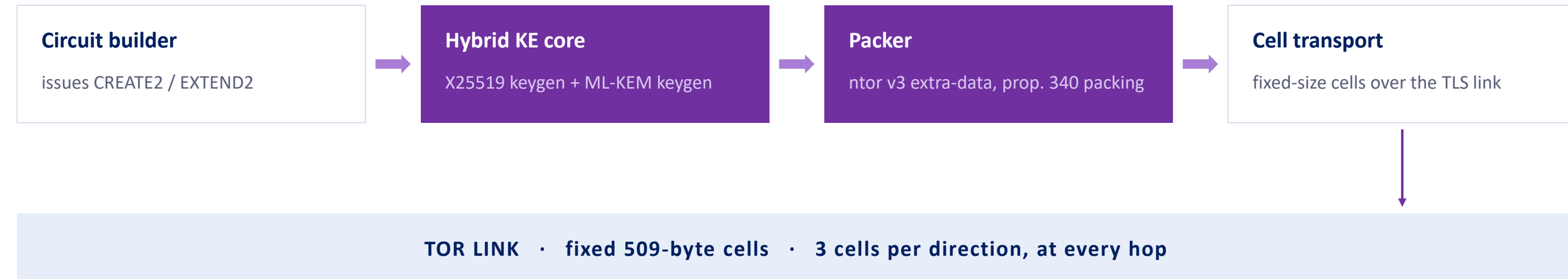
DISCLOSED VERSION SKEW

Memory measurements and the ARM64 cross-build were taken in a later round against liboqs 0.16.0. ML-KEM itself did not change between versions, and no comparison in the paper spans the two rounds we flag it rather than imply one monolithic run.

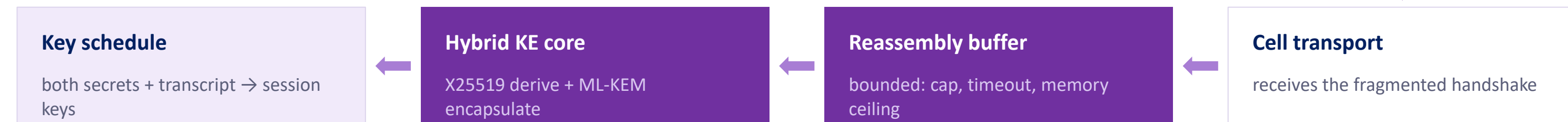
Architecture and Data Flow

WHAT THE MODIFICATION TOUCHES — AND WHAT IT DELIBERATELY DOES NOT

CLIENT



RELAY



■ Added by this work

■ Modified

□ Unchanged Tor code path

Only circuit construction changes. Steady-state relay-cell cryptography is untouched, so the cost appears once per hop at build time and never again while the circuit is in use.

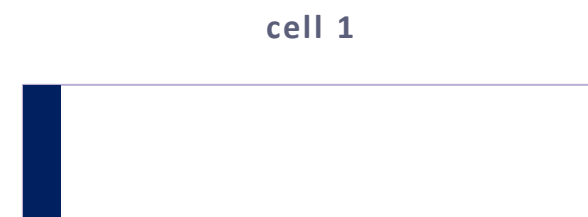
The Constraint Is the Cell, Not the Cryptography

ML-KEM KEY MATERIAL IS THIRTY TO FIFTY TIMES LARGER THAN THE X25519 IT AUGMENTS

PACKED-AND-FRAGMENTED CELLS

X25519

public key · 32 B



1 cell

ML-KEM-768

encapsulation key · 1,184 B



3 cells

ML-KEM-768

ciphertext · 1,088 B



3 cells

Drawn to scale against the 493-byte usable body of the proposal 340 relay cell. The 509-byte payload is fixed; uniform cell size is itself a privacy feature.

Fragmentation becomes mandatory

Neither the key nor the ciphertext fits in one cell, and the legacy format cannot span cells.

Padding is unavoidable

295 B is wasted in the final cell, the price of keeping every cell the same size.

New coupling appears

Reassembly now interacts with flow control, congestion, and traffic analysis.

This Work

A MEASURED END-TO-END INTEGRATION, WITH EVERY NUMBER LABELLED BY HOW IT WAS OBTAINED

01

Threat framing

HNDL stated specifically for onion routing. Confidentiality migrates now; post-quantum authentication is deferred, and we say why.

02

Hybrid handshake

ntor v3 plus ML-KEM-768 under the concatenate-then-derive combiner the IETF specifies for X25519MLKEM768, with fragmentation aligned to Tor proposals 340 and 355.

03

Measurement

liboqs and OpenSSL benchmarks, real resident memory, deterministic cell counts, a circuit-build breakdown, an ARM64/NEON build and a synthesised FPGA datapath.

HOW EACH NUMBER IN THIS TALK IS GROUNDED

● MEASURED

liboqs 0.15.0 with AVX2, X25519 via OpenSSL 3.0.13, one Intel Xeon core at 2.80 GHz. At least five seconds of iterations per operation.

● DETERMINISTIC

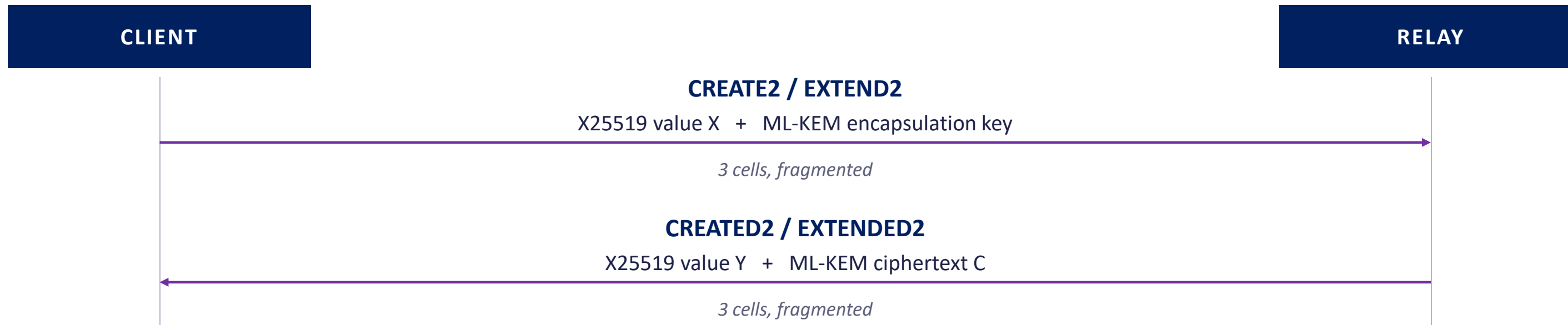
Cell counts and padding follow exactly from the FIPS 203 sizes over a 493-byte body. Arithmetic, not samples.

● MODELLED

Per-hop and circuit-level composition over a representative 95 ms segment. The Shadow harness ships as an artefact, not yet run.

One Key Schedule, Two Independent Secrets

THE HYBRID RUNS BOTH EXCHANGES IN PARALLEL AND BINDS THEM INTO THE ntor v3 TRANSCRIPT



KEY SCHEDULE INPUT

`s_dh || s_kem || ID || B || X || Y || C || PROTOID`

Identical in structure to the X25519MLKEM768 group the IETF defines for TLS 1.3, we adopt it for the reason the TLS working group did, namely that it has a published security reduction.

WHY DEGRADATION IS SAFE

Either primitive suffices. The concatenation combiner is IND-CCA secure if either input KEM is, and the KDF is a sound PRF.

The ciphertext is bound. C sits inside the transcript, so substituting it invalidates the relay's authentication tag.

No silent downgrade. Domain separation forces both secrets to be present and correct before any tag verifies.

Fragmentation Introduces Three New Problems

NONE OF THEM CRYPTOGRAPHIC • ALL OF THEM PROPERTIES OF THE TRANSPORT

3

fragments per 1,216 B client handshake

295 B

of padding wasted in the final cell of every key

3×

cells per direction, at every hop of the circuit

1 Reassembly-state DoS

WHAT BREAKS

A relay must buffer partial fragments before it can compute, so an adversary who never sends the last one inflates that state at will.

WHAT BOUNDS IT

Per-source cap on incomplete handshakes, a short eviction timeout, a global memory ceiling. Policy-local, like half-open TCP defences.

2 Atomicity under loss

A one-cell handshake either arrives or does not. A three-cell handshake can fail partially and stalls until retransmission completes the set.

Genuinely a new failure mode. It surfaces in the tail of the build-time distribution, which is a question for Shadow, not for a closed-form model.

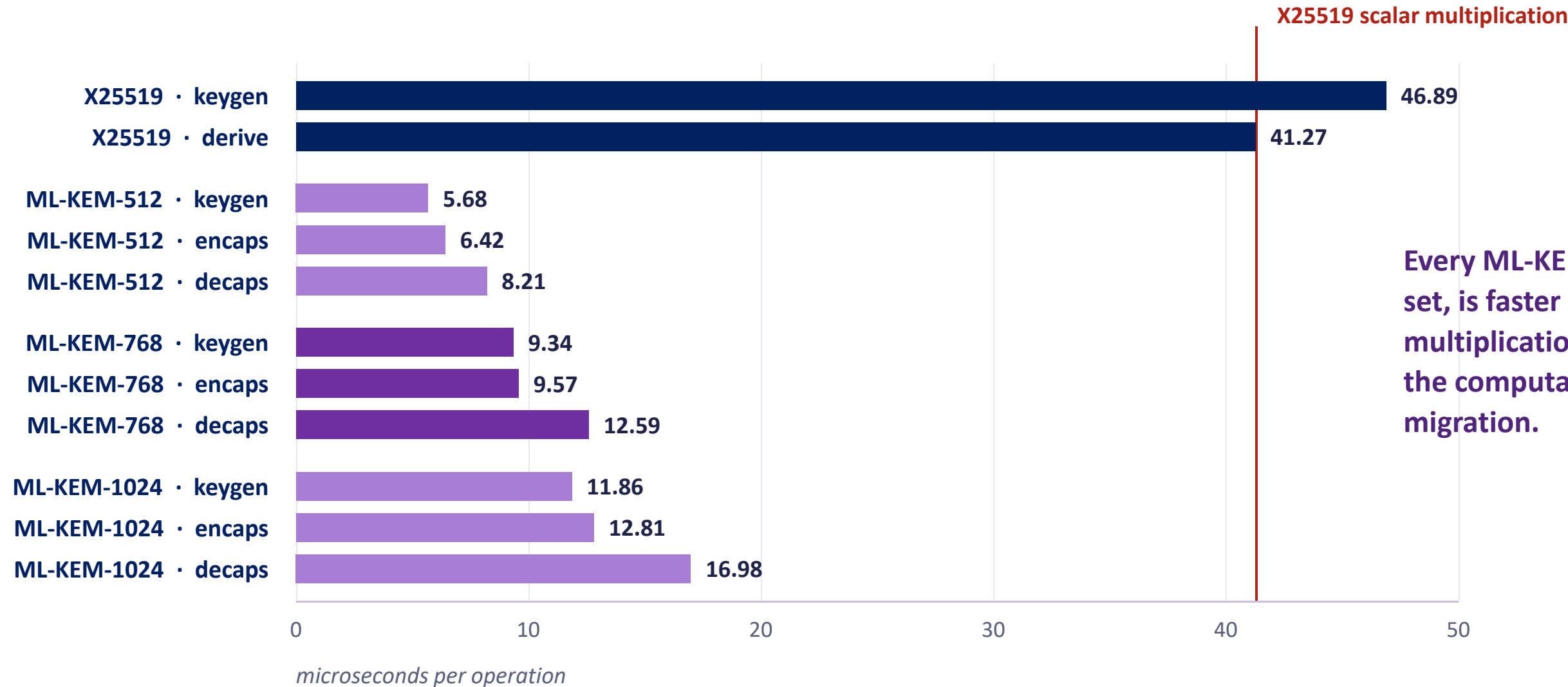
3 Traffic-analysis burst

Three consecutive cells per direction is a distinguishable burst shape even though every cell stays full-size and encrypted.

Interleave cover cells; standardise the fragment count across parameter sets so the KEM choice is not itself a signal. Neither is exercised here.

Every ML-KEM Operation Beats One X25519 Scalar Multiplication

MEASURED • LIBOQS 0.15.0 AND OPENSSSL 3.0.13 • INTEL XEON 2.80 GHZ • UBUNTU 24.04



Every ML-KEM operation, at every parameter set, is faster than a single X25519 scalar multiplication. Post-quantum security is not the computationally heavy part of this migration.

Vectorised SHA-3 with AVX2 makes the lattice arithmetic outrun the elliptic-curve operation it augments.

4.3× cheaper

ML-KEM-768 encaps vs. X25519 derive

Composed Cost: 1.12× Compute, 3× Cells

MODELLED OVER MEASURED PRIMITIVES • DETERMINISTIC CELL COUNTS OVER A 493-BYTE BODY

HANDSHAKE COMPUTATION

Scheme	per-hop (μs)	3-hop (μs)	× base
Classical ntor v3	258.9	776.6	1.00
+ ML-KEM-512	279.2	837.5	1.08
+ ML-KEM-768	290.4	871.1	1.12
+ ML-KEM-1024	300.5	901.5	1.16

Two ephemeral key generations and four scalar multiplications per classical hop. The hybrid adds one KEM keygen and one decapsulation at the client, and one encapsulation at the relay.

WIRE COST

Scheme	pk (B)	pk cells	ct (B)	ct cells
X25519	32	1	32	1
ML-KEM-512	800	2	768	2
ML-KEM-768	1,184	3	1,088	3
ML-KEM-1024	1,568	4	1,568	4

Exact values from the FIPS 203 sizes and the proposal 340 cell body. The final cell of an ML-KEM-768 public key wastes 295 B of padding.

Compute grows 1.12×. Cells grow 3×. The price of this migration is paid on the wire, not in the CPU.

Circuit Build, Memory, and Positioning

OF THE MODELLED 3.61 MS INCREASE, ROUGHLY 97% IS SERIALISING THE EXTRA CELLS

Component	Classical	Hybrid-768	Share of Δ
Network (telescoping)	570.00 ms	570.00 ms	0%
Cryptographic compute	0.78 ms	0.87 ms	$\approx 3\%$
Serialization (cells)	1.76 ms	5.27 ms	$\approx 97\%$
Total (modelled)	572.53 ms	576.14 ms	100%

Measured memory footprint

3.400 KB per in-flight ML-KEM-768 keypair

- VmRSS delta over 1,000 concurrent live keypairs 2.9% under the theoretical sum
- 10,000 in-flight handshakes ≈ 34 MB, negligible for a busy guard relay

Against prior hybrid work

- Earlier designs predate the standard and use ring-LWE or NTRU, reporting $\approx 1.25\text{--}1.28\times$ handshake processing
- We measure $1.12\times$ with the finalised ML-KEM (FIPS 203)
- The reassembly buffer, not keypair storage, is the quantity a relay must actually bound

WHERE THE 3.61 MS GOES



3% ML-KEM computation

97% serialising two extra cells per direction

Network delay is unchanged by construction: neither primitive alters the number of round trips, only the bytes carried within them.

Hardware Feasibility: ARM64/NEON and FPGA

DOES THE FINDING HOLD OFF SERVER-CLASS X86? WE CHECKED RATHER THAN EXTRAPOLATED

ARM64 / NEON

Cross-compiled and functionally verified

- liboqs built for AArch64 against the native backend with hand-written NEON kernels, not compiler auto-vectorisation
- A genuine aarch64 ELF: disassembly confirms 8,559 instructions on NEON vector registers
- The emulated run reproduces the exact FIPS 203 sizes and reports AES, SHA-2, SHA-3 and NEON as active
- All three ML-KEM parameter sets were cross-built and exercised, not only the recommended one

Boundary. QEMU dynamic binary translation does not preserve timing, so no ARM performance figure is claimed. A ready-to-run benchmark ships as an artefact.

FPGA · LATTICE ECP5-45K

Verified Cooley–Tukey butterfly datapath

- The most-instantiated primitive in the NTT 896 of them per forward transform over one polynomial
- Barrett reduction with $m = 5,039$; 5,005 test vectors, zero mismatches; Yosys and nextpnr, open toolchain end to end

Metric	Combinational	3-stage pipelined
LUT4 (logic)	64	148
Hard multipliers	4	4
Flip-flops	0	84
Achieved Fmax	-	45.8 MHz

Under 1% device utilisation. A correct but latency-unoptimised baseline: the critical path is the Barrett reducer's ripple carry chain, which is precisely what a retiming pass should target first.

Conclusion and Future Work

TRANSITIONAL POST-QUANTUM CONFIDENTIALITY AT A MODEST AND WELL-LOCATED COST

What we found

- ML-KEM runs faster than the X25519 it augments, so per-hop handshake computation grows only about twelve percent
- The genuine cost is structural: key material thirty to fifty times larger than a Curve25519 key, carried through a network built on 509-byte cells
- Roughly 97% of the modelled circuit-build increase is serialisation, not cryptography
- Effort belongs in cell packing, reassembly and tail behaviour under congestion, not in optimising lattice arithmetic that is already cheap
- Aligning with proposal 340 keeps the number of distinct cell formats down and lets the migration ride on work already underway

What remains open

- **Shadow end to end.** Run the patched handshake under real congestion and loss, where the atomicity penalty will actually surface. Highest priority.
- **Traffic analysis.** Quantify whether the three-cell burst measurably improves circuit-construction fingerprinting against real background traffic.
- **PQ authentication.** FALCON becomes the right default once FIPS 206 is finalized roughly 666-byte signatures against ML-DSA-44's 2,420.
- **Algorithm diversity.** HQC is a structurally independent code-based backup to ML-KEM; a three-way hybrid would be strictly more conservative.
- **Real silicon.** An ARM64 figure on physical hardware, and a full NTT with Keccak/SHAKE alongside the verified butterfly.

References

AS CITED IN THE PAPER

- [1] R. Dingledine, N. Mathewson, P. Syverson. Tor: the second-generation onion router. USENIX Security, 2004.
- [2] I. Goldberg, D. Stebila, B. Ustaoglu. Anonymity and one-way authentication in key exchange protocols. DCC 67(2), 2013.
- [3] C. Gidney, M. Ekerå. How to factor 2048-bit RSA integers in 8 hours using 20 million noisy qubits. Quantum 5:433, 2021.
- [4] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms. SIAM J. Comput. 26(5), 1997.
- [5] L. Chen et al. Report on post-quantum cryptography. NIST IR 8105, 2016.
- [6] Tor Project. Proposal 355: Revisiting post-quantum cryptography for circuit handshakes, 2024.
- [7] I. Lovecruft, P. Schwabe. Proposal 270: RebelAlliance, a post-quantum secure hybrid handshake.
- [8] NIST. Module-lattice-based key-encapsulation mechanism standard. FIPS PUB 203, Aug. 2024.
- [9] IETF. ML-KEM post-quantum key agreement for TLS 1.3. draft-ietf-tls-mlkem, 2025–2026.
- [10] Anonymous. Post-quantum migration of Tor. Cryptology ePrint Archive, Paper 2025/479.
- [11] N. Mathewson. Proposal 332: ntor protocol with extra data, version 3, 2021.
- [12] S. Ghosh, A. Kate. Post-quantum forward-secure onion routing. ACNS, 2015, pp. 263–286.
- [13] J. Schanck, W. Whyte, Z. Zhang, N. Mathewson, I. Lovecruft, P. Schwabe. Proposal 269: transitionally secure hybrid handshakes, 2016.
- [14] J. Schanck, W. Whyte, Z. Zhang. Circuit-extension handshakes for Tor achieving forward secrecy in a quantum world. PoPETs 2016(4).
- [15] Tor Project. Proposal 340: Packed and fragmented relay cells.
- [16] NIST. FIPS 203, 204 and 205, Aug. 2024.
- [17] Cloudflare. The state of the post-quantum Internet. Cloudflare Research, 2025.
- [18] M. Braithwaite. Experimenting with post-quantum cryptography (CECPQ2). Google Security Blog, 2019.
- [19] F. Giacon, F. Heuer, B. Poettering. KEM combiners. PKC 2018, LNCS 10769, pp. 190–218.
- [20] N. Bindel, J. Brendel, M. Fischlin, B. Goncalves, D. Stebila. Hybrid KEMs and authenticated key exchange. PQCrypto 2019, LNCS 11505.
- [21] K. Kwiatkowski, P. Kampanakis, B. E. Westerbaan, D. Stebila. Post-quantum hybrid ECDHE-MLKEM key agreement for TLS 1.3. IETF, 2026.
- [22] P.-A. Fouque et al. Falcon: fast-Fourier lattice-based compact signatures over NTRU. NIST PQC Round 3 submission.
- [23] R. Perlner. FIPS 206: FN-DSA (Falcon). NIST Sixth PQC Standardization Conference, Sept. 2025.
- [24] NIST. FN-DSA (Falcon) draft standard submitted for approval, Aug. 2025.
- [25] NIST. NIST selects HQC as fifth algorithm for post-quantum encryption, Mar. 2025.
- [26] C. Aguilar Melchor et al. Hamming Quasi-Cyclic (HQC). NIST PQC Round 4 submission, 2017–2025.

THANK YOU

BACKUP SLIDES

Supporting material for Q&A - outline, introduction, background, literature survey, architecture, methodology and references.

Background

ntor AND ntor v3

- A one-way authenticated key exchange: the client already knows the relay's long-term identity key from the network consensus, and the relay authenticates implicitly only the holder of the private key can complete the exchange.
- Version 3 (proposal 332) carries additional encrypted setup data inside CREATE2 and EXTEND2, with the explicit caveat that the extra client-to-relay data does not receive forward secrecy.
- That extensibility is the hook on which post-quantum migration hangs: a defined place to attach a KEM public key and ciphertext without redesigning the message types.

THE CELL FORMAT

- Tor multiplexes all traffic into fixed-size cells with a 509-byte payload. Uniform cell size is itself a privacy feature, because it frustrates traffic analysis.
- Proposal 340 introduces a packed-and-fragmented relay cell: the recognised and digest fields move to the front, the usable body becomes 493 bytes, and several logical messages can be packed into one cell or one message spread across several.
- That substrate is exactly what a post-quantum handshake needs, so we align with it rather than inventing a parallel mechanism.

THE STANDARDS

- In August 2024 NIST finalised ML-KEM (FIPS 203), ML-DSA (FIPS 204) and SLH-DSA (FIPS 205).
- ML-KEM derives from CRYSTALS-Kyber; its security rests on the module learning-with-errors problem, and its three parameter sets target roughly AES-128, AES-192 and AES-256.
- Hybrid key exchange has moved from experiment to production: browsers and CDNs now negotiate X25519 with ML-KEM-768 for a growing share of TLS traffic, and earlier trials such as CEC PQ2 surfaced the middlebox and packet-size problems that motivate careful handling of large key shares.