

The Drill

What a Cryptographic Swap Actually Costs

Mohamad Amin Hasbini

Independent researcher · Paris, France

Published October 13, 2026

ABSTRACT

Four drills were run on one reference stack on 5 October 2026: a receipt-signature swap from Ed25519 to ML-DSA-65, an agent-identity rotation with its TLS handshake cost, a channel re-termination and a cascade containment. Each engine signature and verification call, and each handshake, took milliseconds, and whole-ledger operations took seconds, mostly the engine's process start-up. The failures came from rules and settings that no component of the stack held, from a verifier policy to the definition of isolation. The paper reports the measurements, frames each drill as an exercise a regulated firm can run on a copy of its estate, and publishes the runbook template.

KEYWORDS: AI agent security, crypto-agility, post-quantum migration, ML-DSA, receipt ledger, key rotation, TLS termination, cascade containment, DORA testing, crisis exercise

CITE AS

Hasbini, M. A. (2026). The Drill: What a Cryptographic Swap Actually Costs. Running Regulated Agents Series, Paper #2.

DOI <https://doi.org/10.5281/zenodo.23322699>

Available at <https://mahasbini.org/papers/s3-02-the-drill/>

PDF <https://mahasbini.org/publications/papers/s3-02-the-drill.pdf>

TL;DR

- **While each signing, verification and handshake completed within milliseconds, the failures that mattered came from policies and settings that fell between the components of the stack, where no one had been asked to own them.** On one reference stack, starting the move of a signed receipt ledger from Ed25519 to ML-DSA-65 took 96.9 milliseconds of machine time in a single timed run, from new-key generation to the first verified post-quantum record, and whole-ledger operations took seconds. The drills did not measure how long settling those policies would take.
- **In a ledger whose record hashes cover the signatures, do not re-sign old receipts in place.** Re-signing the 500 classical records in place broke the hash chain at all 500 links, although every new signature verified. Keeping the classical public key trusted let a holder of the retired private key forge new records that verified, and removing it from the trust store failed 501 legitimate records. A verifier policy keyed to the swap point, held outside the ledger, closed both sides.
- **An ML-DSA-65 certificate chain pushed the server's first flight past the initial window.** On minimal test chains, that flight was 14,880 bytes with classical key exchange and 15,954 with the hybrid group, both over the 14,600-byte upper bound RFC 6928 sets for TCP's initial window. ML-DSA-44, which some national guidance does not accept, stayed under it. What a production chain adds was not tested and could use up its margin of about 2.6 kB. Three older clients failed against ML-DSA-only servers, and a dual-certificate server kept all three working.

- **Moving the termination point downgraded all the hybrid-capable clients tested that could still connect, while the health checks stayed green.** On a local equivalent of the stack's terminator, an edge carrying a legacy group template did this in 5 of 5 runs. A posture probe that offers only the hybrid group caught that step but later passed while the hop behind the edge was classical, and the self-check endpoint reported whichever hop it terminated.
- **In the reference fleet, isolation was only a status flag.** Containing an agent interrupted all of its delegations in a median of 60.7 milliseconds. Its key still signed receipts that verified, its own ledger recorded nothing, and a restart released it.
- **A drill is an exercise day a regulated firm can run.** Four drills on a copy of the estate produce measured results, a list of decisions without an owner and a filled runbook, whose template is the paper's appendix.

A swap that started in a tenth of a second

On the morning of Monday 5 October 2026, a receipt ledger on the author's reference stack changed its signature scheme. Five hundred records signed with Ed25519 sat on the ledger. A transition record was then appended, carrying the hash of the last classical record and signed with the old key and a new ML-DSA-65 key side by side. ML-DSA is NIST's module-lattice post-quantum signature standard, FIPS 204, and ML-DSA-65 is its middle parameter set. In the drill's single timed run, the first post-quantum record was signed and verified 96.9 milliseconds after the new key began to be generated.

Then the verifier was asked to stop trusting the retired key, as every migration eventually asks. At that point 501 of the ledger's 1,001 records failed verification together. Putting the public key back in the trust store restored them. It also let anyone holding the classical key's private half append a new classical record after the swap, and the engine accepted that forged ledger as valid. Without a rule saying when the classical key stops counting, neither keeping its public key trusted nor removing it was acceptable.

A verifier policy of about twenty lines, which knows the swap point, the rule for every record after it and the ledger's expected length, closed both sides. It flagged all four forged ledgers the engine had accepted after the swap. The ledger, the engine and the trust store had each done their own job correctly, and the policy belonged to none of them, which on a stack run by one operator is a design gap.

The same kind of gap turned up in each of the four drills run that day. [Paper II.2](#) ended its five Monday actions with the line that agility you have not rehearsed is agility you do not have. This paper rehearses it on a reference stack, reports what was measured, and turns each drill into an exercise a regulated firm can run on a copy of its own estate.

Why rehearse, and why measure

Published migration frameworks for post-quantum cryptography are plentiful, and most describe the same sequence of inventory, prioritisation, planning and migration. Measured rehearsals are fewer, although the handshake cost of post-quantum certificate chains has been measured in public since 2019. In payments, the BIS Innovation Hub's Project Leap, in its second phase, replaced the classical signature on liquidity transfers in a test environment of the Eurosystem's T2 with a post-quantum one. Its December 2025 report gives verification times, and it found that hybrid signatures were not feasible in the existing configuration and would need substantial changes to the system. In the sources reviewed for this paper, much less is published on what a swap breaks in hash-chained records and agent identities, or on who owns the decisions it exposes.

[Paper III.1](#) gave an estate a way to measure itself, and the agility ladder inside its model already placed "a scheme swap you have rehearsed" at the top. Running a drill like the four below is what that rung asks for.

The author's own registers of the post-quantum and agent-security markets, compiled from public sources and not yet checked entry by entry, suggest the same gap. In the post-quantum register, migration advisory and discovery each have more than 80 entries, while migration execution and managed operation have at most 7 documented offers each, and none of the entries under discovery, agility or migration records a certificate. In the agent-security register, 162 entries offer red teaming or testing of AI systems or agents, and none mentions TLPT or TIBER-EU. One post-quantum entry lists a tabletop exercise, on the cryptographic inventory rather than the swap. These counts describe the registers, not the whole market. In them, planning and discovery are widely offered, while an offer to rehearse a swap hardly appears.

That gap matters more now that the transition dates are published. The coordinated European roadmap asks Member States to have their first steps in place by the end of 2026, and high-risk use cases migrated by the end of 2030. In the United States, an executive order of June 2026 directs federal agencies, through OMB guidance, to move their high value assets and high impact systems to post-quantum digital signatures by 31 December 2031. NIST's draft transition guidance lists EdDSA, the family Ed25519 belongs to, among the algorithms disallowed after 2035.

Some records in regulated finance carry multi-year retention rules, for example five years for an investment firm's order and transaction data under Article 25(1) of MiFIR. [II.3](#) argued that agent receipts should be kept on the same supervisory horizon. Where they are, and where the firm or its national roadmap classes the receipt stream as a high-risk use case, a receipt signed today will meet the European roadmap's end-2030 date inside its own retention horizon, since five years from now already runs past it.

Every result below comes from the author's reference stack on one workstation. It is a baseline of that stack, never a benchmark of any product or estate. Keys, signatures, verifications, hash chains, TLS handshakes and chain validation are real cryptographic operations, while the four-agent fleet and its request stream are simulated and labelled as such. All drill traffic stayed on the machine, so network round trips were not measured. The method and the failure classes should transfer to another estate, while the milliseconds describe this stack only.

Drill 1: the receipt swap

What a firm drills. Take one signed record stream the firm relies on as evidence, for example an agent's action receipts or a signed audit trail. On a copy, move it from its current signature scheme to a post-quantum one. Then ask the three questions an inspector would ask later: whether the old records still verify, whether they should be re-signed, and whether the chain of evidence survives the boundary.

What was measured. The drill built a hash-chained receipt ledger of 1,001 records, 500 under Ed25519, one transition record signed by both keys, and 500 under ML-DSA-65. A second, hybrid ledger started from a copy of the same 500 classical records and continued, after its own transition record, with 500 new records signed with both ML-DSA-65 and Ed25519. Each record carries the SHA-256 hash of the full previous record, signatures included. The engine accepts a record when every signature present on it verifies against a trusted key, and it requires no particular algorithm, which the stripping forgeries below exploit.

An Ed25519 signature is 64 bytes and an ML-DSA-65 signature 3,309. The ledger stores signatures as base64 text, 88 and 4,412 characters, so the median record grew from 1,226 to 5,552 bytes, about 4.5 times, and a thousand such records from 1.23 MB to 5.55 MB. Verifying the drill's ledger took 9.5 seconds in a single run, and the hybrid one 18.2 seconds. Most of that time is the engine starting one library process per operation, about 8 milliseconds each, so it says more about the engine than about the algorithms.

What broke. Old records verified after the swap, all 500 of them, for as long as the classical public key stayed in the verifier's trust store. Removing it from the trust store failed 501 records, the 500 classical records and the

transition record. A classical-only verifier build also failed 501 records on both ledgers, this time the transition record and the 500 after it. Every signature on a record must verify, and that build could not check an ML-DSA-65 signature, while the 500 classical records still verified under it.

On copies, replacing the signatures in place and adding a countersignature in place both broke the hash chain at all 500 links, while every new signature verified. Each record's hash covers its signatures and the next record commits to that hash, so to this engine's verifier a re-signed ledger was indistinguishable from a tampered one. Rebuilding the whole chain under the new key produced a valid chain, at the price of changing all 1,001 record hashes, which breaks every external reference to any record.

The forgery cases used the real retired Ed25519 private key as a stand-in for a future break of the classical scheme. The transition record protected history, since it carries the hash of the last classical record under its ML-DSA-65 signature and every later record commits to the one before it. A rewrite of any classical record therefore broke the chain, at the next record or, if the forger re-linked every later classical record, at the transition.

Four forgeries of three kinds still verified as valid while the classical key was trusted. One appended a classical record after the swap to the post-quantum ledger. Two stripped the post-quantum signature from the newest record of the hybrid ledger, with or without a rewrite. The fourth cut the post-quantum ledger back to its 500 classical records, removing the transition record and everything after it, and rewrote records 250 to 499. Truncation is the subtle one, because a ledger that simply ends at the boundary looks complete unless something outside it records how long it should be.

What had no owner. No component of the stack held the verifier's swap policy. It covers the sequence number of the swap, the rule that every later record must carry a post-quantum signature, the classical key's validity window, and a ledger length or head hash anchored outside the ledger. In the policy written for the drill, the post-swap signature rule caught the appended record and the two stripped records, and the anchored length caught the truncated ledger. The policy did not flag a rewritten control ledger that had never been swapped, which only an anchored head hash would catch.

How long retired public keys stay trusted was not held anywhere either. In this ledger design, the cheapest option that keeps the evidence intact appears to be not re-signing at all. That holds provided the ledger length or head hash is recorded at each check in a store the ledger's operator cannot rewrite, with its own access control and retention. The drill assumed such an anchor and did not build one.

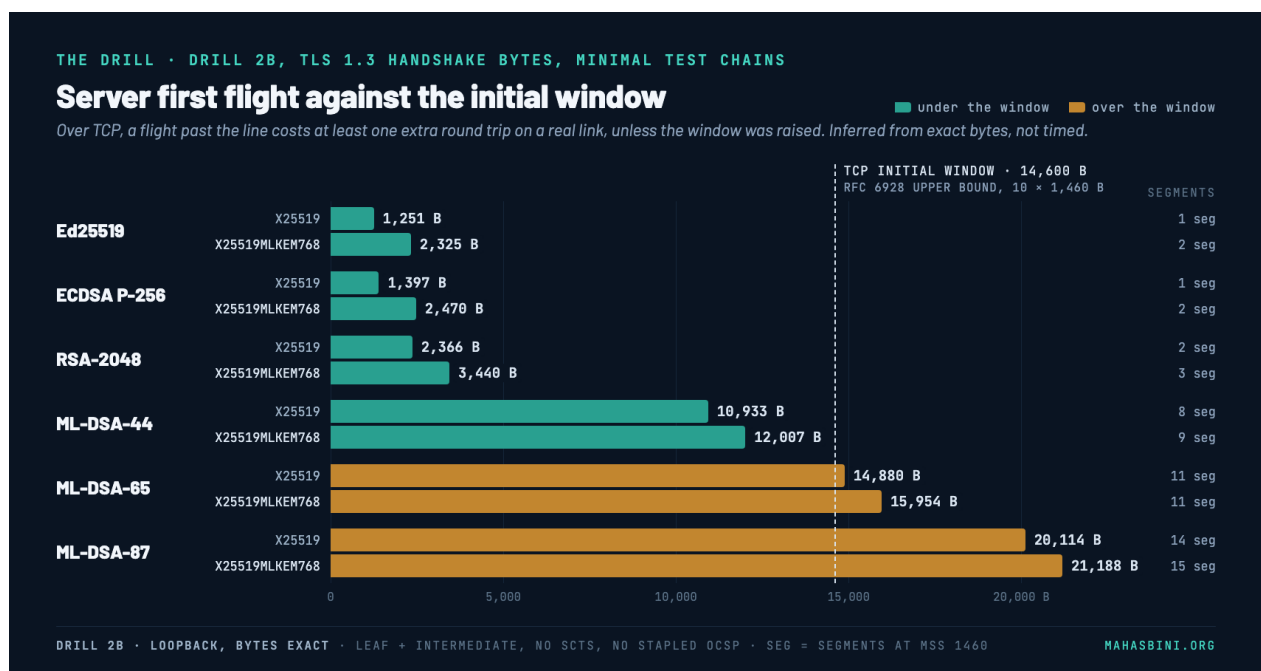
Drill 2: the identity rotation

What a firm drills. Rotate the signing keys of a small group of agents to a post-quantum scheme while they keep serving, each agent keeping its identifier. Separately, measure what a post-quantum certificate chain does to the TLS handshakes those agents and their callers depend on. Drill 2a tests the rotation procedure, and Drill 2b the network and the clients.

What was measured, the fleet (Drill 2a). In a simulated four-agent fleet with real keys and receipts, each agent rotated from Ed25519 to ML-DSA-65 while a request stream kept running. A rotation meant a new keypair, a rotation receipt signed by the old and the new key and sealed on the agent's own ledger, its verification, and a first post-quantum receipt. Over 7 runs and 28 agent rotations, the median time to swap one agent was 89.2 milliseconds, with a range of 63.6 to 117.4 milliseconds. The whole fleet rotated in a median window of 357 milliseconds, and all four agents ended on ML-DSA-65 in all 7 runs. No cryptographic failure occurred, although in every run exactly one request was refused inside a rotation window and the simulated caller did not retry it.

What was measured, the handshake (Drill 2b). Three-level test chains, a root, an intermediate and a leaf, were built locally for six signature algorithms, Ed25519, ECDSA P-256, RSA-2048, ML-DSA-44, ML-DSA-65 and ML-DSA-87, and the server sent the leaf and the intermediate. With no transparency timestamps, no stapled status response and no certificate compression, these are minimal test chains, and production chains are usually larger. The client offered either classical X25519 or the hybrid X25519MLKEM768 share, which pairs X25519 with the post-quantum ML-KEM-768.

The threshold is the upper bound that RFC 6928 sets for TCP's initial window, 10 segments and at most 14,600 bytes. A server first flight above it cannot be sent in full before the client acknowledges the first segments, so on a real link it costs at least one extra round trip, unless the server's initial window has been raised. The drills inferred that round trip from exact byte counts without timing it, and QUIC was not tested.



In these minimal test chains, the signature algorithm decided whether the first flight fitted the window.

CHAIN SIGNATURE	SERVER FIRST FLIGHT WITH X25519 (BYTES)	WITH X25519MLKEM768 (BYTES)	FITS THE 14,600-BYTE WINDOW
Ed25519	1,251	2,325	yes
ECDSA P-256	1,397	2,470	yes
RSA-2048	2,366	3,440	yes
ML-DSA-44	10,933	12,007	yes
ML-DSA-65	14,880	15,954	no
ML-DSA-87	20,114	21,188	no

The hybrid group added about 1.1 kB to the server's first flight, and grew the client's first message from 239 to 1,417 bytes.

In the hybrid case the ML-DSA-44 flight took nine segments, leaving a margin of 2,593 bytes on this minimal chain. What a production chain adds, such as more names and extensions or the timestamps and stapled response these chains lacked, could consume that margin, and none of it was tested.

On the machine itself the processing cost was small, a median of 1.35 milliseconds per handshake for ECDSA with X25519 against 1.97 for ML-DSA-65 with the hybrid group. On a real link, the extra round trip is likely to cost more than the processing.

What broke. Three older clients, an OpenSSL 3.0 command-line client and two curl builds, one of them on LibreSSL, failed the handshake against both ML-DSA-only servers tested, one with an ML-DSA-44 chain and one with an ML-DSA-65 chain, one attempt per client and server, 6 of 6. A server holding two certificates, ML-DSA-65 and ECDSA P-256, kept all three working in 10 of 10 attempts each.

To a current OpenSSL client at its default settings, the same server still sent the ML-DSA-65 chain in 10 of 10 attempts, whichever certificate was configured first. TLS 1.3 only requires the certificate to fit a signature algorithm the client offers (RFC 9846, Section 4.5.1.2), so which one a server prefers when both fit is up to the implementation, and this result describes OpenSSL on both ends, not TLS in general.

What had no owner. The retry of requests refused during rotation had no owner, because the fleet drops them and leaves the decision to the caller, and the simulated caller had no retry rule. The certificate profile, meaning which ML-DSA level, for which link of the chain, single or dual certificate, sits between whoever runs the PKI and whoever runs the terminator. The initial congestion window is a network setting that neither of them owns. Drill 4 adds one more item, the rotation of an agent that was already isolated when a rotation ran. The rotation skips that agent, and nothing schedules its rotation before the agent is released.

Drill 3: the channel re-termination

What a firm drills. Move the point where TLS terminates on purpose, from the origin to an edge and back, and watch what each class of client actually negotiates at each step. Termination moves in real estates through CDN onboarding, load-balancer changes and cloud migrations, and [II.1](#) argued that agent traffic usually terminates on infrastructure the entity does not control. The drill asks what such a move costs to make and to undo.

What was measured. The drill ran on a local equivalent built for the purpose, which reproduces the termination shape and group list of the author's own post-quantum video-conferencing deployment, Meetade, open by invitation only. Its TLS endpoint is public, and anyone with OpenSSL 3.5 or later can see the hybrid handshake with `openssl s_client -connect meetade.mahasbini.org:443 -groups X25519MLKEM768`. The equivalent also reproduces the deployment's self-check endpoint, the plain-HTTP hop behind the terminator, and graceful reload. It is not the deployment's own proxy or TLS stack, and it runs without DNS or a public certificate authority. With an edge in front, the hop behind the edge, also called the edge-to-origin or inner hop, is a different hop from the plain-HTTP one behind the origin's terminator. A seven-step sequence ran five times, with a graceful reload for each group change. The step medians ran from 7.5 to 17.2 milliseconds on loopback, and a restart at the last step, used there in place of a reload, cost about 27 milliseconds of unavailability. They are not a deployment estimate, since they leave out DNS lifetimes, certificate issuance and change approval.

What broke. Moving termination to an edge carrying a legacy group template, hardened before hybrid groups existed and so limited to classical curves, downgraded all the hybrid-capable clients tested that completed a handshake through it, three configurations (the default client, the health check and an API call), to classical X25519 in 5 of 5 runs, while the health check returned 200 at every observation point. The deployment's own posture probe, a script that connects offering only the hybrid group and passes if the server accepts it, failed against the edge at that step, also in 5 of 5 runs. The certificate's issuing CA changed with the move. A test client that trusted only the origin CA's root therefore failed, and after the move back a client that trusted only the edge CA's root failed.

A connection opened before a graceful reload kept its original group for every request it carried after the reload, and a connection opened before the switch back kept being terminated at the edge. That persistence is inherent to TLS, whose parameters are fixed at the handshake. It was built into the local equivalent to match graceful reload, so it was not observed on the live deployment.

Once the edge sat in front, the origin's self-check endpoint reported the edge-to-origin hop and no longer the visitor's. When the origin was then rolled back to classical groups, a client connecting through the edge stayed on the hybrid group. The same probe passed against the edge in 5 of 5 runs while the inner hop was classical. Pooled edge-to-origin connections kept the old group for the whole six seconds observed, until the edge itself reloaded.

What the live estate showed. In place of re-running the sequence on the deployment's own terminator image, the drill added a read-only, client-side observation of the author's live estate, which already runs both termination shapes. The video-conferencing deployment terminates at its own origin, and the author's other public hosts terminate at a content-delivery edge. The observation used 92 connections to six hosts from one home connection in Paris over about two minutes, two of them runs of the deployment's probe script. On both shapes, a current OpenSSL client at its default settings got the hybrid group in 18 of 18 connections.

Every connection from a classical-only or older client, 72 of 72, negotiated X25519 and got HTTP 200. Nothing in those exchanges warned that the channel was classical. The group appeared only in the body of the self-report a client has to request, and whether the servers logged it could not be seen from the client side. Each self-report described only the client-facing hop, and nothing on the estate reports the hop behind the edge, which could not be observed from outside.

A browser request to one edge host shortly before the run went over HTTP/3 with an encrypted client hello, while every connection in the run used TCP, so these were separate handshakes at the edge, and the posture a probe reports depends on the transport it uses. The whole observation is one vantage point and one snapshot. It shows that classical negotiation completes on a live estate without any warning, and it did not observe a hybrid-capable client being downgraded there.

What had no owner. The edge's certificate and issuer, its client-facing group template, its upstream group list and its connection pooling all lacked an owner in the drill. So did the origin's direct exposure once an edge sat in front of it, what the self-check endpoint reports, and what the posture probe proves. A probe that offers only the hybrid group proves that the server can negotiate it, without showing that a default client gets it or anything about the hops behind the endpoint.

Drill 4: cascade containment

What a firm drills. Treat one agent as compromised, isolate it, interrupt every delegation that touches it, and measure what keeps running, what stops, and what "isolated" means in the system that enforces it. [Series I](#) spent a paper on delegation, and this drill is the operational test of its unwinding.

What was measured. In the simulated four-agent fleet, a peer agent sealed an isolation order, which means that it signed the order as a receipt and appended it to its own hash-chained ledger. Every outstanding delegation was interrupted by a receipt sealed the same way, and a one-second observation window then ran under the request stream. Over 13 single-containment runs across all four agents, the time from issuing the isolation order to sealing the last interruption receipt had a median of 60.7 milliseconds, with a range of 24.0 to 90.8, and it grew with the number of delegations. All 38 seeded delegations were interrupted, in 13 of 13 runs. The other three agents kept serving, while the isolated agent's share of the requests was refused and not rerouted.

Ten further instrumented runs timed when isolation took effect. The agent's status flipped as soon as the order had been sealed, a median of 16.4 milliseconds after the order was issued. The scheduler served exactly one request in the drill window in every run in which it was the target (10 of 10 runs over HTTP, 5 of 5 instrumented runs). In the five instrumented runs where the scheduler was the target, that request had entered before the order was issued, but in one of the five its receipt was sealed after the isolation order, so the ledgers show the agent acting after its isolation order. A platform needs to say whether containment is judged from the moment an order is sealed or the moment it takes effect.

What broke. The same ten instrumented runs showed that isolation in this reference fleet was a status flag in the controller process. The isolated agent refused new work, while its private key stayed on disk. A receipt it signed after isolation verified with public keys alone in 10 of 10 runs, including in the full fleet verification, because verification had no notion of isolation. Its own ledger carried no trace of its isolation, which was recorded only in the ordering peer's ledger, the counterparts' ledgers and the drill ledger. A restart booted every agent as active, in 10 of 10 runs.

In a separate sequence, run once, that contained the agents one after another, the fourth agent could not be contained once three were isolated. The authority to isolate belonged to peers, and no active agent remained to seal the order. In another sequence, run three times, an agent was contained first and the rotation that followed skipped it, leaving it the only one in the fleet still signing with an Ed25519 key.

What had no owner. Rerouting of the isolated agent's work had no owner, and neither did a verifier rule that rejects what the agent signs after its isolation while keeping its earlier receipts valid. The same was true of persistence of isolation across restarts, of the authority to isolate when no healthy peer remains, and of a marker on the isolated agent's own record. These results come from one reference application with a simulated fleet, and a production agent platform is likely to implement isolation differently. The drill is still worth running elsewhere, since each of its questions maps to a check on a copy, and on this stack none of the answers had been written down.

The four drills side by side

THE DRILL · FOUR DRILLS, ONE REFERENCE STACK

The four drills side by side

Here "no owner" means that no component of the reference stack held the policy or setting and no written rule assigned it.

MACHINE TIME MEASURED

1

Receipt swap

96.9 ms, ONE TIMED RUN
NEW-KEY GENERATION TO FIRST
VERIFIED POST-QUANTUM RECORD

2

Identity rotation

89.2 ms PER AGENT
357 ms PER FLEET (MEDIAN)

3

Channel re-termination

7.5 TO 17.2 ms PER STEP
ON LOOPBACK (MEDIAN), NOT
A DEPLOYMENT ESTIMATE

4

Cascade containment

68.7 ms, ORDER TO LAST
INTERRUPTION (MEDIAN)

WHAT BROKE

Re-signing in place broke the hash chain

At 500 of 500 links. Removing the classical public key from the trust store failed 501 of 1,001 records. Keeping it trusted let four forgeries verify.

Older clients failed the handshake

Three, against ML-DSA-only servers (6 of 6). ML-DSA-65 test chains pushed the first flight past the 14,600-byte window. One request refused per run (7 of 7), not retried.

Silent downgrade, health checks green

A legacy edge template downgraded every hybrid-capable client to X25519 (5 of 5). Clients trusting only one CA's root failed. Self-check and probe described only their own hop.

Isolation was only a status flag

The isolated agent's key still signed receipts that verified (10 of 10). Its work was refused, not rerouted. A restart released it (10 of 10).

WHAT HAD NO OWNER

The verifier's swap policy

Swap point, rule for later records, the classical key's validity window, a length or head hash anchored outside the ledger. Also the trust-store lifecycle, the re-signing decision.

The certificate profile

Which ML-DSA level, for which link, single or dual certificate, between PKI and terminator. Also request retry, the initial window, rotation of isolated agents.

The edge, on both of its hops

Its certificate and issuer, client-facing and upstream group templates, connection pooling, the origin's direct exposure, what the self-check reports, what the probe proves.

A post-isolation verifier rule

Rejecting what the agent signs after its isolation, earlier receipts still valid. Also rerouting, persistence, isolation authority, the agent's own record.

DRILL RECORD · MON 5 OCT 2026 · REAL CRYPTOGRAPHY, SIMULATED FLEET · A BASELINE, NEVER A BENCHMARK

MAHASBINI.ORG

DRILL	MACHINE TIME MEASURED	WHAT BROKE	WHAT HAD NO OWNER
1. Receipt swap	96.9 ms from new-key generation to first verified post-quantum record (one run)	In-place re-signing broke 500 of 500 chain links. Removing the classical public key from the trust store failed 501 of 1,001 records. Keeping it trusted let four forgeries of three kinds verify	The verifier's swap policy and the anchored ledger length, the trust-store lifecycle, the re-signing decision
2. Identity rotation	89.2 ms per agent, 357 ms per fleet (medians)	One request refused per run, not retried. ML-DSA-65 chains over the initial window. Three older clients failed against ML-DSA-only servers	Request retry, the certificate profile, the initial window, rotation of agents held in isolation
3. Channel re-termination	7.5 to 17.2 ms per step on loopback (medians of 5 runs each), not a deployment estimate	Silent downgrade of all the hybrid-capable clients tested, with health checks green. Clients trusting only one CA's root failed. Self-check and probe described only their own hop	Edge certificate, group templates on both hops, connection pooling, origin exposure, what self-checks report
4. Cascade containment	60.7 ms, order to last interruption (median)	Work refused, not rerouted. The isolated key still signed valid receipts. A restart released the agent	Rerouting, a post-isolation verifier rule, persistence, isolation authority, the agent's own record

Here "no owner" means that no component of the reference stack held the policy or setting and no written rule assigned it.

The owner gaps cluster at the boundaries between components that each behaved correctly, the ledger and its verifier, the PKI and the terminator, the edge and the origin, the controller and the trust store. In a firm, each team can make its own component pass, while the boundary between two teams usually belongs to neither of them until somebody assigns it. At this scale the cryptography was cheap by comparison, since even the hybrid receipt and the ML-DSA-65 handshake that crosses the window cost milliseconds and kilobytes per record or per handshake on this machine. The decisions in the last column are likely to cost more, because each of them needs a person, a written rule and, usually, a change to something another team runs. The drills did not measure how long a firm takes to make them.

The drill as an exercise day

On the reference stack, whose receipt engine and simulated fleet already existed, all four drills ran on one morning. For a firm, one working day is a reasonable target for running them on a copy that already exists. Building the copy, approving the exercise and remediating afterwards fall outside that day and were not timed here. The day is a format a regulated firm can run itself or commission. It resembles the crisis exercises many regulated firms already hold, except that those usually rehearse people responding to an incident, whereas this one has the estate change its cryptography and its containment under controlled conditions. It then asks the people in the room who owns what each change exposed.

Before the day. Choose the surface with the smallest blast radius and the oldest signature, usually one receipt or audit stream, one agent group and one terminated endpoint. Build the copy and confirm that it is one. It should have its own test keys, certificates and trust roots, test credentials only, synthetic or approved data, and no route by which it can sign, call or publish into production. Check that the original is untouched, for example with file hashes taken before and after, as the drills did. Write down the scope, the stop rule and who can halt the day, then, before anything runs, the planned time for each drill and the result that would count as success.

Those written predictions are part of the evidence, since the gap between plan and result can be one of the most useful findings of the day.

The morning. Run the four drills on the copy, each with the inject set on its page of the runbook template in the last appendix. The injects run from a classical signing key assumed broken that morning to an agent assumed compromised. Each drill records its machine times, what broke, and every item that needed a decision nobody present could take.

The afternoon. The afternoon walks the last column of the drill record. Each item without an owner gets a name, a date and a written rule, or an explicit decision to accept the risk. The day's output is a filled runbook, an owner map, the drill record with its measured results, and remediation lines that feed straight into the scorecard of III.1. The room needs the people listed in the runbook header, from the CISO or a delegate to someone from compliance who will later need the record.

The guard on the day. Everything that changes runs on a copy, and no production private key or credential is used for an inject. Observation of the live estate, where it is useful, stays read-only and client-side, as it did in Drill 3. If a step would change production, the drill stops there, and at the end of the day the copy's test keys are destroyed or archived under the firm's rules.

Where the drills sit in the regulation

None of the instruments below names agents, scheme-swap rehearsals or post-quantum cryptography as a test type. Their duties are technology-neutral and still reach drills of this kind, most clearly in regulated finance, where DORA is directly applicable.

DORA requires financial entities other than microenterprises to establish, maintain and review a digital operational resilience testing programme (Regulation (EU) 2022/2554, Article 24(1)). It also lists the kinds of tests that programme may include, in an open list naming scenario-based tests and compatibility testing among others (Article 25(1)). A drill of this kind can be run as a scenario-based test of the estate's ability to change, although DORA does not prescribe it. If a firm counts it as a test of that programme, it must be undertaken by independent parties, internal or external, without conflicts of interest, and the issues it reveals must be prioritised, classified and remedied (Article 24(4) and (5)). Entities that the competent authorities identify for it must also run threat-led penetration testing at least every three years (Article 26(1) and (8), third subparagraph). These drills do not replace such testing, since threat-led tests run on live production systems supporting critical or important functions (Article 26(2)), and every change in these drills was made on a copy.

DORA also requires the ICT business continuity plans and the ICT response and recovery plans to be tested at least yearly. The same clause requires a test in the event of any substantive changes to ICT systems supporting critical or important functions (Article 11(6)(a)). Where a receipt ledger supports such a function, a scheme swap under the signatures of its records is the kind of substantive change that clause appears to anticipate. For entities under its full framework, the technical standard on ICT risk management requires the policy on encryption and cryptographic controls to include provisions "for updating or changing, where necessary, the cryptographic technology on the basis of developments in cryptanalysis". An entity that is not able to update or change it must adopt mitigation and monitoring measures. The policy must also require their adoption to be recorded with a reasoned explanation (Commission Delegated Regulation (EU) 2024/1774, Article 6(4) and (5)). II.2 read that article as an agility duty in substance. A firm that has never swapped a scheme would find it hard to show it holds that capability, and a drill record is probably the most direct evidence it can produce.

For financial entities identified as essential or important under NIS2, DORA is the sector-specific act under NIS2 Article 4 (Regulation (EU) 2022/2554, Article 1(2)). NIS2's Articles 21 and 23 and its supervision and enforcement chapter therefore do not apply to them, and its recital 28 asks Member States not to apply those

provisions to any financial entity DORA covers (Directive (EU) 2022/2555, Article 4(1) and recital 28). NIS2 carries the same families for the other essential and important entities, among them the cloud, data-centre and managed service providers a bank may rely on. It binds them through national transposition, which as of early October 2026 was still incomplete in France, Ireland and Spain. Its risk-management measures include business continuity and crisis management, policies and procedures to assess the effectiveness of those measures, and policies and procedures regarding the use of cryptography (Article 21(2)(c), (f) and (h)). For cloud computing, data-centre and managed service providers, among other digital providers, Commission Implementing Regulation (EU) 2024/2690 sets the technical content of those measures. Its cryptography section asks for protocols and algorithms chosen "following, where appropriate, a cryptographic agility approach" (Annex, point 9.2(b)). Its effectiveness section asks the entity to determine "who is responsible for monitoring and measuring the effectiveness" of the measures (point 7.2(d)). Where NIS2 applies, a drill is one way to carry out that effectiveness assessment for the cryptography measure.

The cybersecurity standard being written for the AI Act, prEN 18282, was at enquiry stage this summer, as II.3 described, and its testing clause addresses resistance to the attack types it lists. The author's enquiry comments argued that containment of a propagating compromise and substitution of the cryptographic mechanisms behind records and credentials should be tested as capabilities in their own right. The four drills above show one way such tests could be run, on one reference stack.

Five things to do Monday

THE DRILL · SERIES III.2

Five things to do Monday

- 1 Pick the smallest surface and drill it on a copy.**

The one with the oldest signature. One receipt or audit stream is enough for a first drill, with the planned time and the success criterion written down before it runs.

written predictions are part of the evidence
- 2 Write the verifier's swap policy before you swap.**

The swap point, the rule for records after it, the retired key's validity window, a ledger length or head hash anchored outside the ledger. Then decide re-signing deliberately.

where record hashes cover the signatures, re-signing in place reads as tampering
- 3 Count the bytes of your planned certificate profile.**

Compare the server's first flight with your initial congestion window, and test your oldest client against the profile. If the flight crosses the window or the client fails, check the lowest level your policy accepts, then try a smaller parameter set for the window, or a dual-certificate server for the older clients.

on minimal test chains, ML-DSA-65 pushed the first flight past the window
- 4 Probe every re-termination with default clients.**

From both sides of the edge. Check which hop your self-check endpoint describes. Alert when a client that could negotiate the hybrid group ends up on classical key exchange.

in the drill, health checks stayed green
- 5 Define what isolation does to the agent's key, then test it.**

Verifiers reject what it signs after the isolation boundary the platform defines and keep accepting earlier receipts. Its private key disabled or out of reach, isolation that survives a restart, an authority to isolate when no healthy peer remains, a marker on its own record.

in the reference fleet, isolation was only a status flag

AMIN HASBINI · MAHASBINI.ORG

1. **Pick the smallest surface with the oldest signature, and drill it on a copy.** One receipt or audit stream is enough for a first drill, with the planned time and the success criterion written down before it runs.
2. **Write the verifier's swap policy before you swap anything.** It needs the swap point, the rule for records after it, the retired key's validity window, and a ledger length or head hash anchored outside the ledger.

Then decide re-signing deliberately, knowing that in a hash-chained ledger whose record hashes cover the signatures, re-signing in place reads as tampering.

3. Count the bytes of the certificate profile you plan to deploy, and test your oldest client against it.

Compare the server's first flight with your initial congestion window. If a post-quantum chain crosses it, or an older client fails, check the lowest level your policy accepts, then try a smaller parameter set for the window, or a dual-certificate server for the older clients, before you change anything else. The dual-certificate server tested here kept the old clients working and still sent the ML-DSA-65 chain to a capable client. ML-DSA-44 sits in NIST security category 2, which some guidance does not accept. The German BSI recommends only ML-DSA-65 and ML-DSA-87, and a post-quantum signature only in combination with a classical one, which neither the ML-DSA-only chains nor the dual-certificate server tested here provides, since each handshake still carries a single signature scheme. CNSA 2.0 requires ML-DSA-87 for US national security systems.

4. Probe every re-termination with default clients, from both sides of the edge. Check what your self-check endpoint actually reports and which hop it describes. Alert when a client that could negotiate the hybrid group ends up on classical key exchange, since nothing else in the estate is likely to tell you.

5. Define what isolation does to the agent's key, then test it. Verifiers should reject receipts the agent signs after the isolation boundary the platform defines, whether the sealing of the order or its taking effect, and keep accepting earlier ones, which is Drill 1's swap-point rule, with its anchored ledger head, applied to one agent. In this engine, removing its key from the trust store would fail its earlier receipts too, as removing the classical public key did in Drill 1. Its private key should be disabled or moved out of reach, the isolation should survive a restart, someone should hold the authority to isolate when no healthy peer remains, and the agent's own record should say it was isolated.

This is the second paper of Running Regulated Agents, after The Assessment and before The Inspection, the day a supervisor asks for the evidence that the first two produced.

Appendix, for method reviewers

On the software and the timings. The receipt engine and the simulated fleet are the author's own reference software and are not published. Repeated measurements are reported as medians, with ranges for the headline timings of Drills 2 and 4, and no figure is a best run picked from several. Drill 1 ran once at full size, so its swap and verification times are single runs, marked as such in the text. The request stream and the workload the receipts describe are simulated, at about 30 requests per second, and so is the client cutover of Drill 3. Engine operations in Drills 1, 2a and 4 start one library process per signature or verification, about 8 ms each, so their timings measure this stack's process cost far more than the algorithms. In the library's own benchmark an ML-DSA-65 signature took 0.49 milliseconds against 0.019 for Ed25519, about 26 times longer, while inside the engine the medians were 9.9 and 8.5 milliseconds, about 1.2 times.

On the window. Byte counts are TLS record bytes, headers included, from the server's first record to its Finished message. With TCP timestamps, which shrink the window to 14,480 bytes, the results were the same. Larger initial windows, packet loss, real round trips and QUIC, which sets its own limits, were not tested.

On the live observation. The connections were read-only, with no retries and no change of any kind. The edge operator and the certificate authorities are not named, since none of the findings depends on which vendor runs the edge or issues the certificates.

On the market figures. In the two registers, an entry is an organisation, such as a vendor, integrator, operator, research lab, alliance, public body or investor. "Documented" means backed by product documentation or

better, up to a recorded certificate, algorithm validation or approval (NIST CMVP or CAVP, BSI). The registers are not published, and figures are aggregate counts only, with no vendor listed or ranked.

On the instruments of law. DORA applies since 17 January 2025, and Article 6 of Commission Delegated Regulation (EU) 2024/1774 sits in Title II, the full framework, while the simplified framework of Title III carries no equivalent clause. Commission Implementing Regulation (EU) 2024/2690 also has continuity and recovery plans tested at planned intervals and after significant changes to operations or risks (Annex, point 4.1.4).

The NIS Cooperation Group's roadmap (2025) is non-binding. The United States date is in section 4(b)(iii) of Executive Order 14412 of 22 June 2026 (91 FR 38483), whose section 4(b)(i) excludes National Security Systems, and OMB Memorandum M-26-15 of 24 June 2026 carries it as a signature migration phase in 2031. The 2035 date is from NIST IR 8547 in its initial public draft, whose Section 4.1.1, Table 2, lists EdDSA at 128 bits of security strength or more as disallowed after 2035, so it should be cited as expected, not final. None of the three is an obligation on European entities beyond what national and sector instruments adopt. DORA's technical standard leaves the retention period of logs to the entity (Commission Delegated Regulation (EU) 2024/1774, Article 12(2)(a)). A firm should therefore compare the retention schedule of each receipt class with these dates. Any restatement of this paper as "DORA requires post-quantum drills" is an overclaim the text does not support.

On the data and the boundary. The drill results, method notes, scripts and raw outputs of 5 October 2026 are held by the author and are not published with this paper. The aggregate results behind every number in the paper, and the source data of the figures, are available to reviewers and supervisors on request. No client estate, no confidential engagement material and no closed-room observation appears in this paper, and the exercise-day format is described from the drills themselves.

Sources (accessed 10 October 2026 unless noted): [Regulation \(EU\) 2022/2554](#) (DORA), OJ L 333, 27.12.2022, p. 1, Articles 1(2), 11(6)(a) and 24 to 27 · [Commission Delegated Regulation \(EU\) 2024/1774](#), OJ L, 2024/1774, 25.6.2024, Articles 6(4), 6(5) and 12(2)(a) · [Directive \(EU\) 2022/2555](#) (NIS2), OJ L 333, 27.12.2022, p. 80, Articles 4(1) and 21(2)(c), (f) and (h), recital 28 · [Commission Implementing Regulation \(EU\) 2024/2690](#), OJ L, 2024/2690, 18.10.2024, Article 1, Annex points 4.1.4, 7.2(d) and 9.2(b) · EUR-Lex, [national transposition measures for Directive \(EU\) 2022/2555](#) · European Commission, [press release of 8 July 2026](#), referral of Ireland, Spain, France and the Netherlands to the Court of Justice for failure to transpose Directive (EU) 2022/2555 · [Regulation \(EU\) No 600/2014](#) (MiFIR), Article 25(1) · NIS Cooperation Group, [A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography](#), Part 1, v1.1 (June 2025) · [Executive Order 14412](#) of 22 June 2026, 91 FR 38483, Section 4(b)(iii), FR Doc. 2026-12909 · OMB, [Memorandum M-26-15](#) (24 June 2026) · [NIST IR 8547](#), initial public draft (November 2024), Section 4.1.1, Table 2 · [FIPS 204](#) (August 2024), Section 4, Tables 1 and 2 · [FIPS 186-5](#), Digital Signature Standard, Appendix A.2.3 (Ed25519 signature length) · [RFC 8032](#) (EdDSA), Sections 5.1.6 and 7 · [RFC 6928](#) (TCP initial window, Experimental, April 2013), Sections 1 and 2 · [RFC 9846](#) (TLS 1.3, July 2026, obsoletes RFC 8446), Section 4.5.1.2 · [RFC 10024](#), Post-Quantum Traditional (PQ/T) Hybrid Key Agreement Mechanisms for TLS 1.3 (Proposed Standard, August 2026) · [BSI TR-02102-1](#), version 2026-01, Sections 5.3.4 and 5.3.4.2 · NSA, The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ, U/OO/194427-22, version 2.1 (December 2024) · prEN 18282:2026, Artificial intelligence - Cybersecurity specifications for AI Systems, enquiry draft of May 2026 (CEN/CLC/JTC 21), Clause 11 "Verification and testing requirements", with the enquiry timeline as described in [II.3](#) · BIS Innovation Hub, [Project Leap Phase 2 report](#) (December 2025), pp. 18, 21 and 23 · Sikeridis, Kampanakis and Devetsikiotis, Post-Quantum Authentication in TLS 1.3: A Performance Study, NDSS 2020, [doi:10.14722/ndss.2020.24203](#) · Paquin, Stebila and Tamvada, Benchmarking Post-Quantum Cryptography in TLS, [IACR ePrint 2019/1447](#) · drill results, method notes and raw outputs of 5 October 2026, unpublished (see "On the data and the boundary").

Appendix, the drill runbook template

Version 1.0, October 2026, published with this paper. Pages are filled on a copy, never on production, with any live observation read-only and client-side.

Header, once per exercise day

- Estate, scope, stop rule, who can halt the day.
- Copy confirmed as one (own test keys and trust roots, test credentials only, no route to production), and how it was checked.
- Date and people present: CISO or delegate, PKI owner, edge and network owner, agent-platform owners, compliance.
- Machine or environment, library versions, background load, disposal plan for the copy's test keys.

Fields for every drill

- Before the run: planned time, success criterion, inject.
- During and after: machine times (median, range, runs), what broke, what had no owner, its new owner with a date and a written rule or an explicit risk acceptance, evidence kept, a scorecard remediation line.

Drill 1, receipt swap (one signed record or audit stream)

- Inject: the signing key is assumed broken this morning and must be retired. Measure: swap start to first verified post-quantum record, record and ledger growth, full-verification time.
- Check old records with the old public key trusted, then removed, without the new key, and on a classical-only verifier build.
- Re-signing on copies (none, replace or countersign in place, rebuild the chain): does the chain hold, do external references to record hashes survive?
- Forge with the retired key: append after the swap, strip the newest record's post-quantum signature, truncate at the boundary.
- Owners: verifier swap policy (swap point, rule for later records, retired-key validity window, anchored ledger length or head hash), trust period of retired public keys.

Drill 2, identity rotation (one agent group, one endpoint)

- Inject: the new certificate profile goes live, and the oldest supported client must keep working. Measure: rotation time per agent and group, refused requests, share of agents rotated.
- Handshake: server first flight per certificate profile against the initial window (10 segments, at most 14,600 bytes), older clients against post-quantum-only and dual-certificate servers.
- Owners: retry of refused requests, certificate profile (parameter set, chain link, single or dual), initial window, rotation of isolated agents.

Drill 3, channel re-termination (one endpoint)

- Inject: termination moves to the edge with the network team's real template, then back. Measure at each step: what default, classical-only and older clients negotiate, health check, certificate issuer, what the self-check reports and for which hop, the probe's verdict.
- Check: connections opened before a reload or switch, pooled edge-to-origin connections, direct origin reachability, browser against probe transport.

- Owners: edge certificate and issuer, client-facing and upstream group templates, connection pooling, origin exposure, what self-checks report, a downgrade alert for hybrid-capable clients.

Drill 4, cascade containment (one agent and its delegations)

- Inject: one agent is assumed compromised and must be contained. Measure: order to isolation, order to last interrupted delegation, what kept serving, what was refused, and whether rerouted.
- Check: do post-isolation signatures still verify, and do verifiers reject them while keeping earlier receipts? Does the agent's own record show the isolation, and does it survive a restart? Can anyone isolate with no healthy peer left?
- Owners: rerouting, a verifier rule for post-isolation receipts, private-key custody, persistence, isolation authority, the agent's own record.

Close of day

- Owner map: each "no owner" item gets a name and a date, or a risk acceptance naming who accepted it.
- Drill record kept with raw outputs, remediation lines entered in the assessment scorecard, the copy's test keys destroyed or archived.
- Next drill: the next-smallest surface, the next-oldest signature.

About the author

Amin Hasbini is an AI and cybersecurity executive. Through his firm, HASBINI CONSEIL, he advises leadership teams and CISOs on AI agent security and post-quantum migration, and he carries the post-quantum and crypto-agility roadmap of the European Champions Alliance's cybersecurity group. He previously spent twelve years in threat research, the last seven directing Kaspersky's Global Research and Analysis Team for the Middle East, Turkey and Africa. mahasbini.org