Testing and Validation

By Tooba Qasim

How to Test Post-Quantum Cryptography Without Breaking Things

Once organisations move from planning to deployment, testing becomes the make-or-break stage of post-quantum cryptography (PQC). This is where theory meets reality.

PQC algorithms may be standardised but that does not mean they automatically work well inside existing systems. They behave differently from classical cryptography. Keys are larger. Handshakes look different. Performance characteristics change. And small assumptions baked into systems over many years can suddenly be exposed.

Testing is not a box-ticking exercise here. It is how organisations avoid outages, interoperability failures and false confidence.

Why PQC Cannot Be Tested Like Traditional Crypto Changes

In the past, cryptographic upgrades were often incremental. Moving from one hash function to another, or increasing key sizes, rarely affected system architecture in visible ways. PQC is different.

Post-quantum algorithms introduce changes that ripple across protocols, libraries and infrastructure. A TLS handshake using hybrid PQC looks very different from a classical one. Certificate sizes increase. CPU usage changes. Network latency can behave in unexpected ways.

This is why PQC testing cannot be rushed or treated as a simple regression test. Organisations need to understand not only whether systems still work, but **how they behave under real conditions**.

Interoperability: The First Thing That Breaks

One of the earliest testing challenges organisations encounter is interoperability.

PQC rarely exists in isolation. During the transition period, systems must communicate with a mix of:

- PQC-enabled clients,
- Classical-only systems,
- Hybrid configurations,
- Legacy devices that cannot be upgraded easily.

This is especially visible in protocols like TLS, VPNs and secure APIs. A single unsupported cipher suite, extension, or handshake option can cause silent failures or unexpected fallbacks.

Testing needs to answer practical questions:

Will PQC-enabled servers still accept connections from older clients?
Do load balancers, proxies and inspection tools handle larger handshakes correctly?
Do certificate chains still validate as expected?

This is why protocol-level testing matters as much as algorithm-level testing.

Performance Testing: Numbers Matter

PQC algorithms are secure but they are not free.

Many post-quantum schemes use larger keys and signatures which affects:

- handshake sizes,
- memory usage,
- CPU load,
- connection setup times.

Research on hybrid post-quantum TLS has shown measurable increases in handshake size and latency depending on algorithm combinations and network conditions. These effects may be negligible in some environments and significant in others, especially in high-throughput systems or constrained devices.

Performance testing should not aim for perfection. It should aim for **predictability**.

Organisations need to know:

Where does performance change?

How much overhead is acceptable?

Which systems are most sensitive?

Testing answers these questions before users experience them.

Backward Compatibility and Fallback Behaviour

Backward compatibility is one of the most underestimated risks in PQC deployment.

Many systems include fallback mechanisms designed to “keep things working” if negotiation fails. In a PQC context, these fallbacks can unintentionally downgrade security if they are not understood and controlled.

Testing should deliberately explore failure cases:

What happens if a PQC handshake fails mid-negotiation?

Does the system fall back to classical cryptography automatically?

Is that fallback visible, logged and acceptable?

These scenarios are rarely covered in normal testing but become critical in a hybrid world.

Lab Testing vs Production-Like Testing

Most organisations begin PQC testing in controlled lab environments. This is the right place to start.

Lab testing allows teams to experiment safely, enable new cipher suites, swap libraries and explore protocol behaviour without operational risk. It is ideal for understanding algorithm characteristics and basic compatibility.

However, lab environments rarely reflect production reality.

Production-like testing is where hidden issues surface. Real traffic patterns, real network latency, real device diversity and real integrations behave differently. Systems that work perfectly in isolation can struggle under load or at scale.

The most effective testing strategies combine both:

- labs for learning and experimentation,
- staging or pilot environments for realism.

Security Validation Beyond “It Works”

Testing PQC is not only about functionality and performance. It is also about security assurance.

Post-quantum implementations must be validated against:

- correct algorithm usage,
- proper parameter selection,
- secure randomness,
- correct integration with key management and certificate lifecycles.

Misconfigurations can undermine even the strongest algorithms. This is why many organisations align PQC testing with existing security validation practices, such as code reviews, cryptographic audits and penetration testing.

The goal is not to prove PQC is perfect, but to confirm it is **used correctly**.

Common Testing Pitfalls to Avoid

A few patterns appear repeatedly in early PQC testing efforts.

One is testing only the cryptographic library, without testing the protocols and systems that depend on it. Another is focusing exclusively on performance benchmarks while ignoring interoperability and fallback behaviour.

A third is assuming that vendor support automatically means readiness. Vendor claims still need validation in your own environment, with your own traffic and dependencies.

Testing is where assumptions are challenged.

What “Good” PQC Testing Looks Like

Successful PQC testing is structured, iterative and visible.

It starts small, expands gradually and produces clear evidence that systems behave as expected. Results are documented, lessons are shared and failures are treated as learning opportunities rather than setbacks.

Most importantly, testing feeds directly into deployment decisions. It informs where PQC can be rolled out safely, where hybrid approaches are needed longer and which systems require redesign or replacement.

Why Testing Is an Investment, Not a Delay

There is sometimes pressure to move quickly once deployment begins. Testing can feel like friction.

In reality, testing is what allows organisations to move faster later. It reduces surprises, avoids emergency rollbacks and builds confidence across technical and non-technical stakeholders.

Post-quantum cryptography is a long-term transition. Careful testing ensures that the systems we protect today remain trustworthy tomorrow.