

Post-Quantum Authentication: Preparing Digital Identity for the Quantum Era

John Gray

Entrust

September 3rd, 2026

We are a CA vendor and operator that provides authentication systems that have long deployment lifecycles that span many years, and delayed action may result in significant security, operational, and compliance challenges in the future. Organizations should prioritize the transition to post-quantum authentication now rather than waiting for large-scale quantum computers to become available.

As part of the process of updating our PKI infrastructure to support PQ authentication, we faced numerous challenges.

- We use TLS stacks to communicate between devices, and almost all of them rely on TLS client authentication. We found that most 3rd party software stacks do not yet support even the most basic ML-DSA authentication, let alone the more conservative choices of SLH-DSA or Composite ML-DSA.
 - Many of our web-based applications are Java based or use components built on Java. Unfortunately, Java doesn't support ML-DSA authentication in any LTS release currently.
 - These systems also communicate with Web browsers or other information services, which use different cryptographic libraries. Even the latest versions did not support ML-DSA authentication. This makes configuring a stable PQ secure environment very difficult.
- We make use of multiple different HSM systems. Regarding the basic cryptographic algorithm support, we did find good support of ML-DSA, but SLH-DSA and Composites are not uniform across HSM systems.
 - Regarding PKCS#11, many HSM early adopters make use of vendor defines which make interoperability a challenge.
 - When used in Certificates, the ML-DSA pure mode is required, which presents challenges when signing large data. Not all HSMs vendors support the "external mu" hashing for deployment efficiency or streaming API's.

- We also make extensive use of CMS (Cryptographic Message Syntax) and found it was not too difficult to upgrade our internal software stacks to support the new PQ algorithms.
 - In some systems we were able to create a sort of classical TLS authentication (because of the limitations mentioned earlier), with PQ CMS authentication layer to provide a hybrid tunnel. It was not our preferred approach, but it was the best we could do given the limitations in PQ authentication at this time.

The big takeaway from the initial experience of moving to PQ secure authentication was that we could not rely on 3rd parties to have support of PQ authentication in their existing systems at present. In systems where we could internally control the cryptographic layers, we had more success.