

Migrating to quantum-safe PKI

A tale of two approaches

Stjepan A. Kovac, Quantum Resistant Cryptography (EU, US, global) May 9, 2024

Hybrid v.s. direct

Dragging feet or jumping ahead?

- In the migration from current quantum-unsafe mechanisms, 2 approaches are proposed: -hybrid (both classical crypto-enabled key exchanges & pqc/qrc)
-direct (pqc/qrc only)
- From a performance standpoint, the only approach that makes sense is the direct approach: indeed no matter the efficiency of the chosen pqc algorithm, combining it with classical ones makes the overall construction heavier.
- From a security standpoint, if the right (most secure) mechanisms are used, it also makes most sense.
- Choose wisely!

Non-Internet uses of PKI

Towards quantum-safe tokens/MFA

- Aside from TLS, PKI is used in many scenarios to this date as a means of auth, where classical passwords and biometrics **cannot** be trusted, even more so in presence of gen. AI.
- One evolutive way to go about it is to upgrade whenever feasible tokens to use quantum-safe PKI.
- Another way is to replace in such cases the physical token with one's own memory, using image-based passwords. We integrated quantum-resistant symmetric cryptography with such a system provided by our partners MIS from the UK. It is being used in high-security access systems for more than a decade in the Far East.

Future use cases for PKI and cryptography

The “final” frontier

- Brain computer interfaces provide one of the most interesting and frightening at the same time opportunities to do things “right” security-wise.
- Especially having in mind non-invasive BCIs, knowledge-based authentication may be facilitated (log in with your brain), yet gets challenged too.
- That in turn creates the need for cyber-physical/physiological protection systems, where the credential storage system, in this case our brain, is protected from undue interference and spying (or psying, as I mistyped!).
- Actual post-quantum (i.e. beyond quantum) tech can be an ally here. To be continued!