Navigaing the quantum readiness journey, with Quantum Safe Cryptography

Tomas Gustavsson

Chief PKI Officer KEYFACTOR


~~Dilithium~~
ML-DSA / FIPS 204
~~Kyber~~
ML-KEM / FIPS 203
~~SPHINCS+~~
SLH-DSA / FIPS 205

KEYFACTOR

KEYFACTOR

ML-DSA / FIPS 204	Dilithium
ML-KEM / FIPS 203	Kyber
SLH-DSA / FIPS 205	SPHINCS+

What if Quantum Computers become capable enough to…

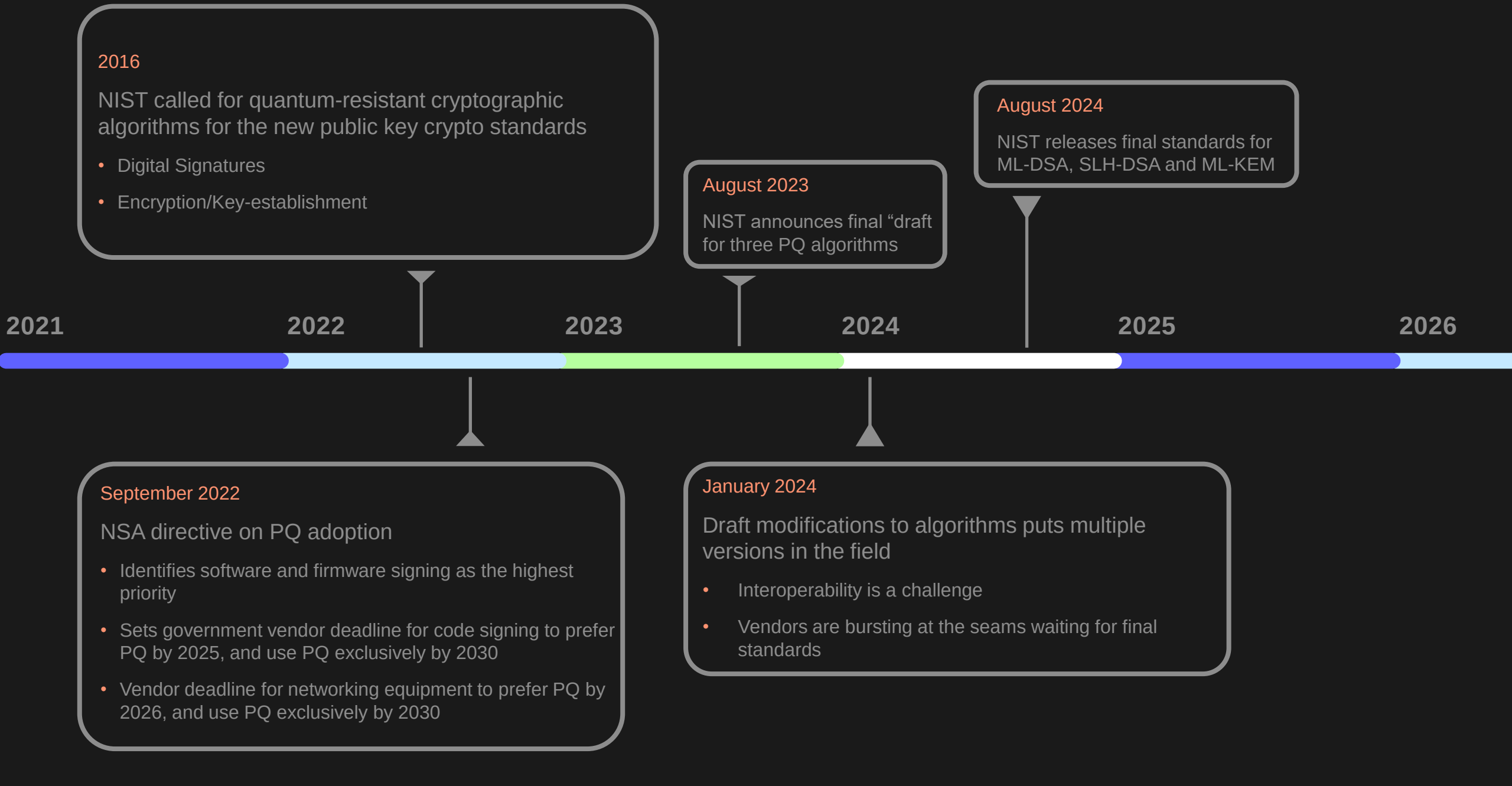
threaten cryptographic algorithms?

Symmetric algorithms such as AES, and Secure Hash functions such as SHA-2, are largely unaffected.

Asymmetric, or public-key algorithms effectively drop to zero strength and are un-salvageable

- RSA, ECC, and ECDSA
  - SSL/TLS, SSH, all Digital Signatures (code, firmware, documents), S/MIME, Bitcoin / blockchain participants….

KEYFACTOR


2016
NIST called for quantum-resistant cryptographic algorithms for the new public key crypto standards
• Digital Signatures
• Encryption/Key-establishment
August 2023
NIST announces final “draft for three PQ algorithms
August 2024
NIST releases final standards for ML-DSA, SLH-DSA and ML-KEM
2021
2022
2023
2024
2025
2026
September 2022
NSA directive on PQ adoption
• Identifies software and firmware signing as the highest priority
• Sets government vendor deadline for code signing to prefer PQ by 2025, and use PQ exclusively by 2030
• Vendor deadline for networking equipment to prefer PQ by 2026, and use PQ exclusively by 2030
January 2024
Draft modifications to algorithms puts multiple versions in the field
• Interoperability is a challenge
• Vendors are bursting at the seams waiting for final standards

Are These Algorithms Different?

SLH-DSA

ML-KEM

ML-DSA

LMS

XMSS

In general terms, the signature algorithms are a lot like what we are used to, just with bigger keys, bigger signatures, and having greater demands on memory and CPU.

Bigger keys

Bigger signatures

KEMs

KEYFACTOR

Key Encapsulation Mechanisms

The PQC answer to Diffie-Hellman key agreement and RSA key transport are quite different to what we are used to (while maintaining the overall feeling of bigness...).

While KEMs have public and private keys, they do their work by passing around secrets contained in encapsulations, which are generated using the same process that generates the secret.

This will mean updating protocols for secret key sharing and key transport as things will not always be done the same way.

KEYFACTOR

The algorithms are coming

Lots of things will need to change

> Everything will change a little; **some things will change a lot**.

- Protocols and formats
  New algorithm identifiers, e.g. TLS, PKCS#11
- HSMs, TPMs, secure elements
- Any product that communicates securely on a network, or use cryptography...
- Cryptographic libraries
- Inventory and prioritization

More Use Cases – More CAs

- Zero Trust
- TLS/mTLS
- Code / Container signing
- Service Mesh / SPIFFE
- SBOM Attestations
- IoT / Manufacturing
- National ID / eID
- ePassport

Any RSA, EC, Ed25519, DH, ECDH will change.

Avoid sub-optimization!


Migration
Plan?
KEYFACTOR

Hybrid TLS

Symmetric Key Exchange at risk.

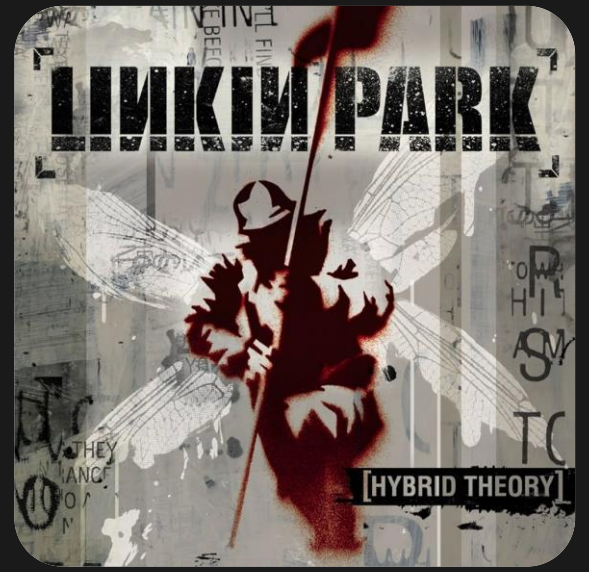
Harvest now Decrypt later

Hybrid TLS Handshake:

1. One classic DH (RSA/EC)
2. One PQ KEM (Kyber)

Belts and suspenders:

- 1 protects for failure in 2
- 2 protects for failure in 1



KEYFACTOR

Hybrid certificates

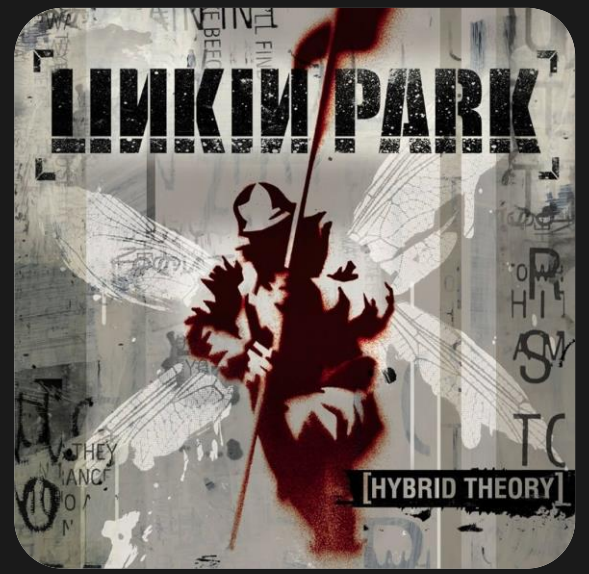
A.K.A "Catalyst", or

X.509 Alternative

Three extensions added to X.509:

1. Alternative Public Key
2. Alternative Signature Algorithm
3. Alternative Signature

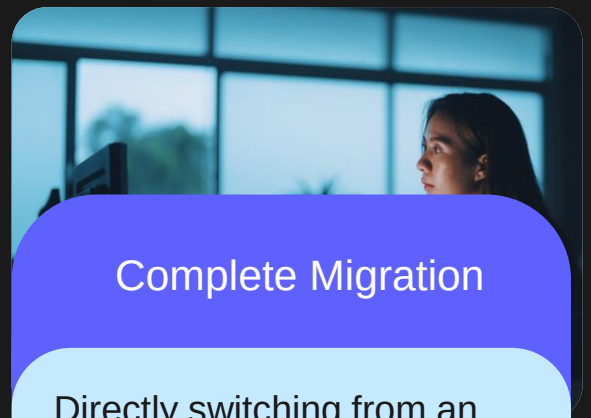
Extensions are non-critical



KEYFACTOR

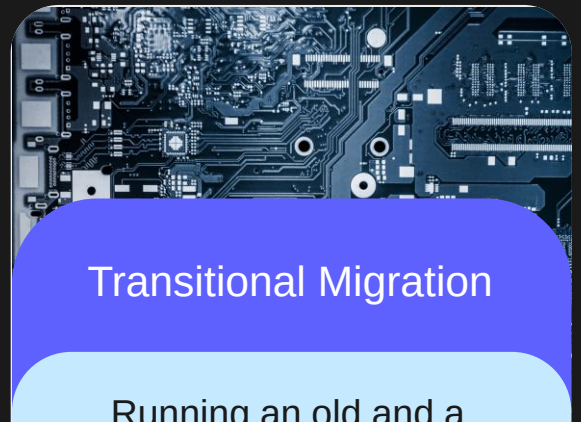
PKI Migration Alternatives

Migrating PKI hierarchies typically fall into one, or a combination of the following alternatives.



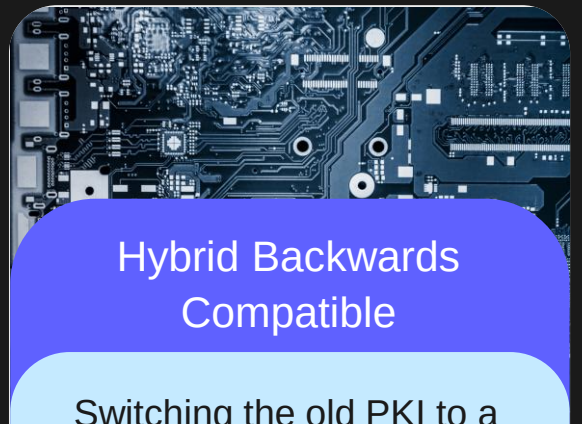
Complete Migration

Directly switching from an old PKI to a quantum-safe PKI.



Transitional Migration

Running an old and a quantum-safe PKI in parallel during the migration phase.



Hybrid Backwards Compatible

Switching the old PKI to a backwards compatible PKI with hybrid certificates.

Composite Non-Backwards Compatible

Switching the old PKI to a non-backwards compatible PKI with composite algorithms.

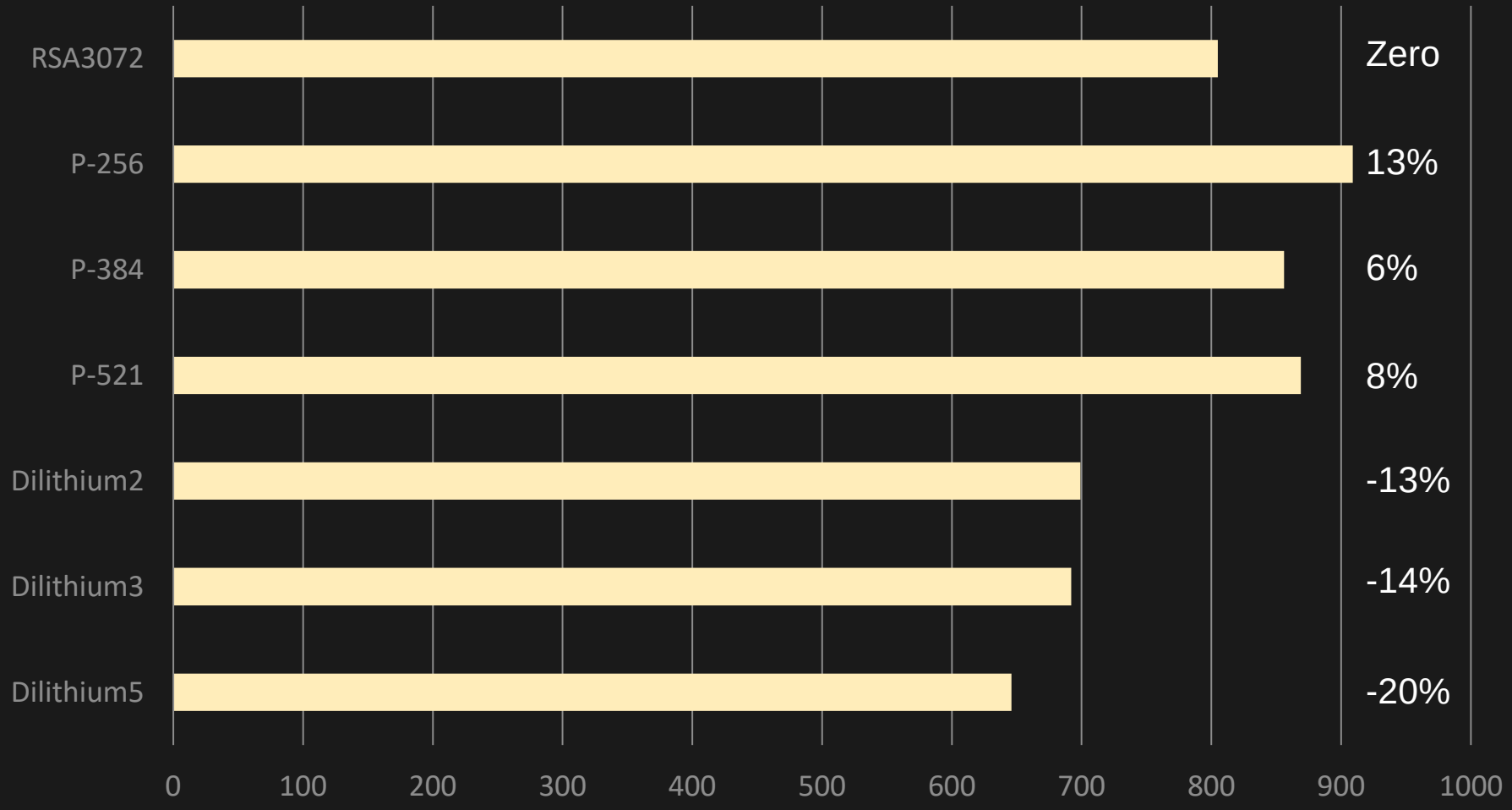
PQC

in Practice

Speed!

KEYFACTOR

Certificate Issuance - Software



KEYFACTOR

Public Key and Signature Size

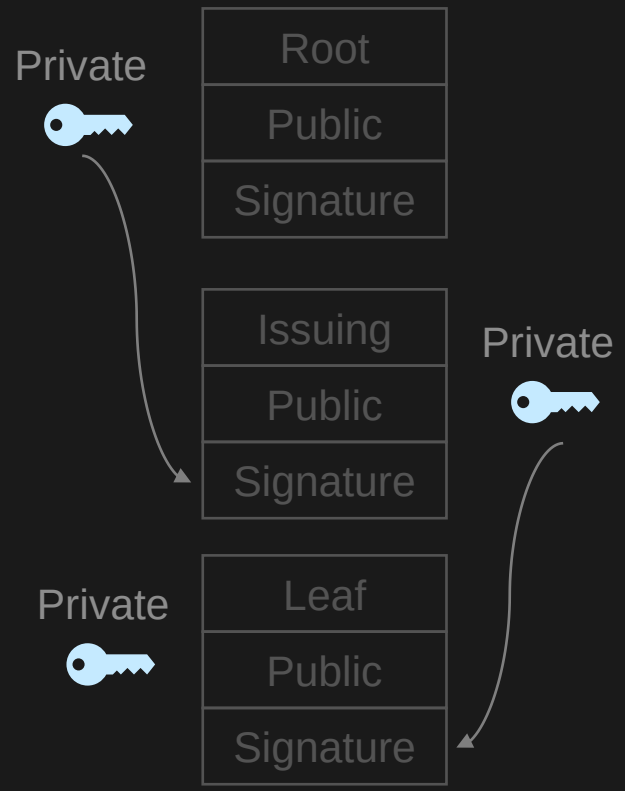
Table 2. Sizes (in bytes) of keys and signatures of ML-DSA

	Private Key	Public Key	Signature Size
ML-DSA-44	2560	1312	2420
ML-DSA-65	4032	1952	3309
ML-DSA-87	4896	2592	4627

Table 2. SLH-DSA parameter sets

	n	h	d	h'	a	k	lg_w	m	security category	pk bytes	sig bytes
SLH-DSA-SHA2-128s SLH-DSA-SHAKE-128s	16	63	7	9	12	14	4	30	1	32	7 856
SLH-DSA-SHA2-128f SLH-DSA-SHAKE-128f	16	66	22	3	6	33	4	34	1	32	17 088
SLH-DSA-SHA2-192s SLH-DSA-SHAKE-192s	24	63	7	9	14	17	4	39	3	48	16 224
SLH-DSA-SHA2-192f SLH-DSA-SHAKE-192f	24	66	22	3	8	33	4	42	3	48	35 664
SLH-DSA-SHA2-256s SLH-DSA-SHAKE-256s	32	64	8	8	14	22	4	47	5	64	29 792
SLH-DSA-SHA2-256f SLH-DSA-SHAKE-256f	32	68	17	4	9	35	4	49	5	64	49 856

Chain Public Key and Signature Size

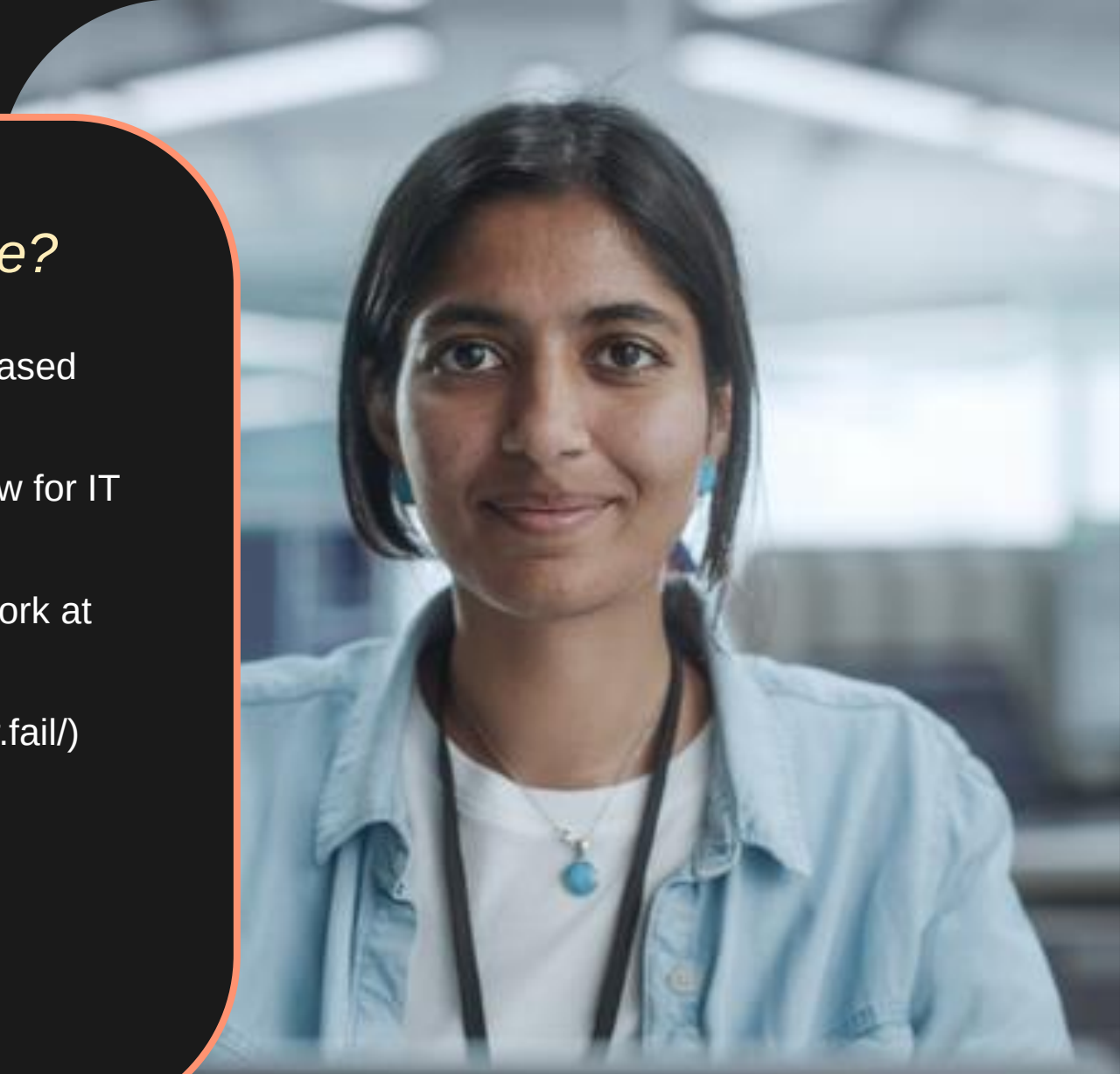


Algorithms	Incl Root	Excl Root
P384 P256 P256	368	224
RSA 4096 RSA 2048 RSA 2048	2304	1280
Falcon-1024 Falcon-512 Falcon-512	6813	3740
Dilithium3 Dilithium2 Dilithium2	13582	8337

KEYFACTOR

Ok, so what does this mean to me?

- You can start ***now***. Hybrid systems enable phased rollout.
- Signing and verification will not be horribly slow for IT systems
- Signing and verification may be slow, or not work at all, for constrained devices
- Some TLS connections may break (https://tldr.fail/)
- Database size increase
  - For example; signed transactions and logs
- Many upgrades
- Many measurements still to be done



KEYFACTOR

Quantum-ready solutions

Now with EJBCA 8.0 and SignServer 6.0

Keyfactor Command

Get an inventory of keys, certificates, and algorithms in use today.

Supports basic inventory of Dilithium certificates

Inventory

Bouncy Castle with Keyfactor

Build applications with post-quantum capable crypto APIs.

Supports all finalist NIST PQC algorithms

Keyfactor EJBCA

Create a post-quantum CA and issue PQC certificates.

Supports FALCON and Dilithium certificate issuance

Keyfactor SignServer

Sign code and artifacts with post-quantum certificates.

Supports SPHINCS+ and Dilithium signing

Test (today) & Transition (in the future)

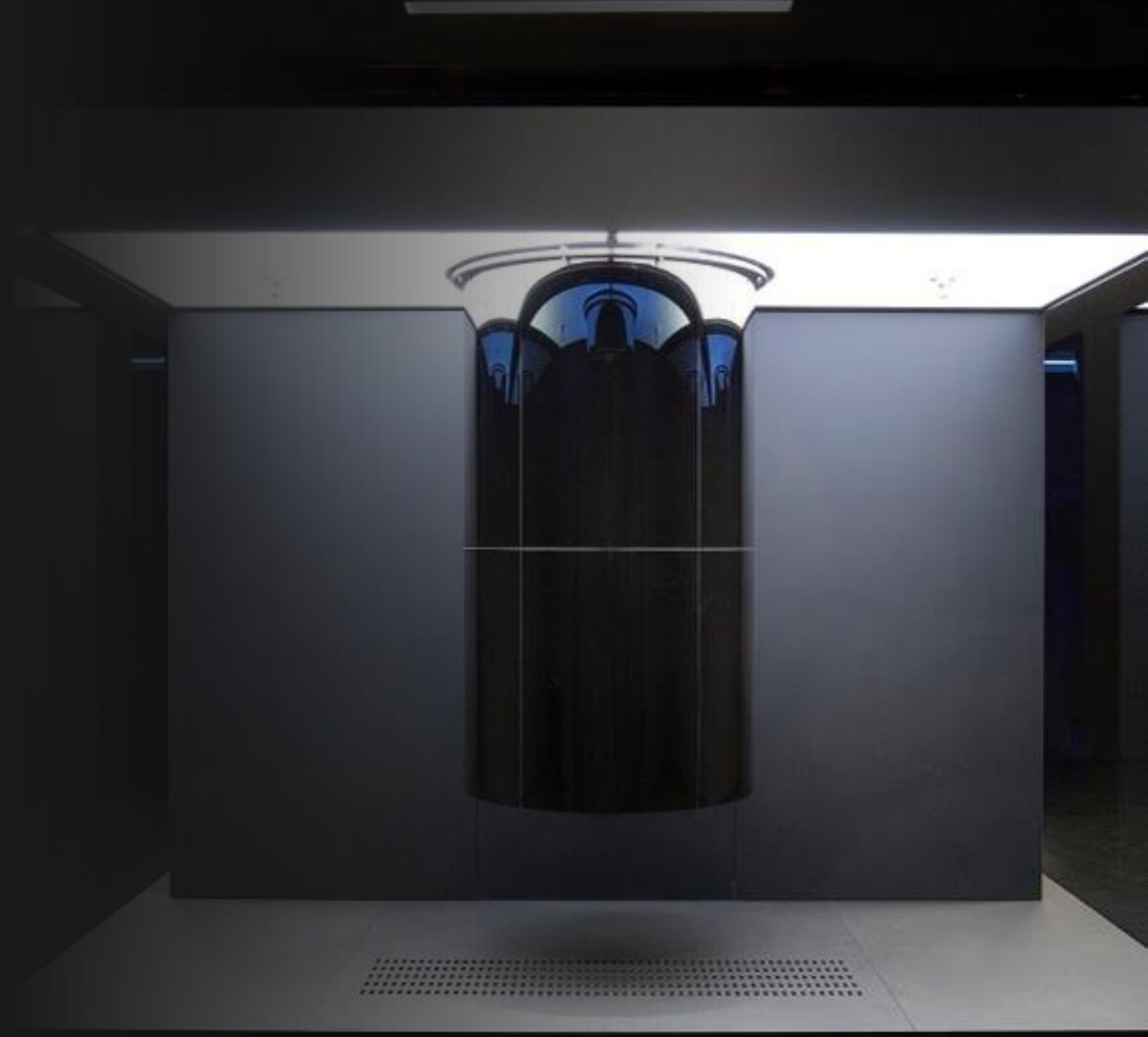
Keyfactor Command

Automate migration and re-issuance from a new post-quantum PKI.

Automate

PQC

for Engineers



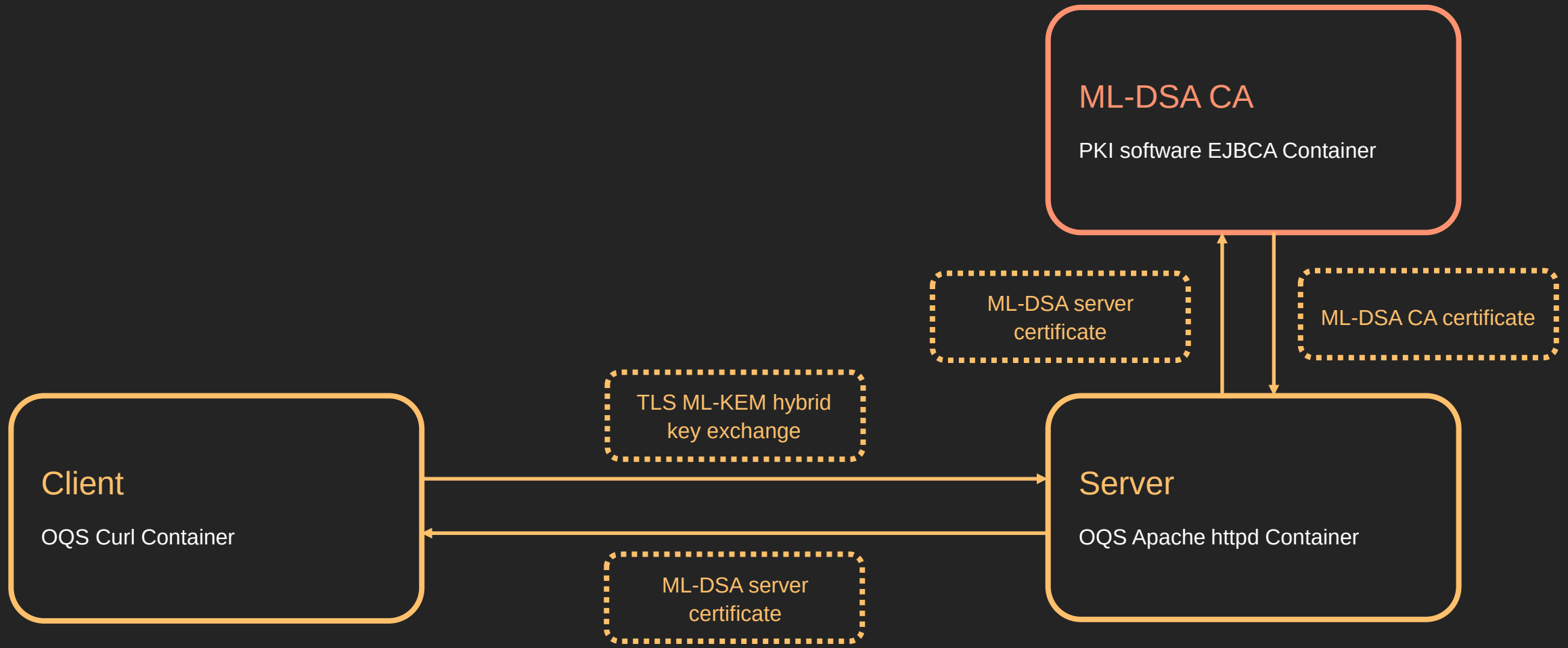
Post-Quantum Cryptography for Engineers:

https://www.ietf.org/archive/id/draft-ar-pquip-pqc-engineers-03.html

KEYFACTOR

Demo

KEYFACTOR


ML-DSA CA
PKI software EJBCA Container
ML-DSA server certificate
ML-DSA CA certificate
TLS ML-KEM hybrid key exchange
Client
OQS Curl Container
Server
OQS Apache httpd Container
ML-DSA server certificate

KEYFACTOR

Tomas Gustavsson

Chief PKI Officer KEYFACTOR

Any Questions?

KEYFACTOR