

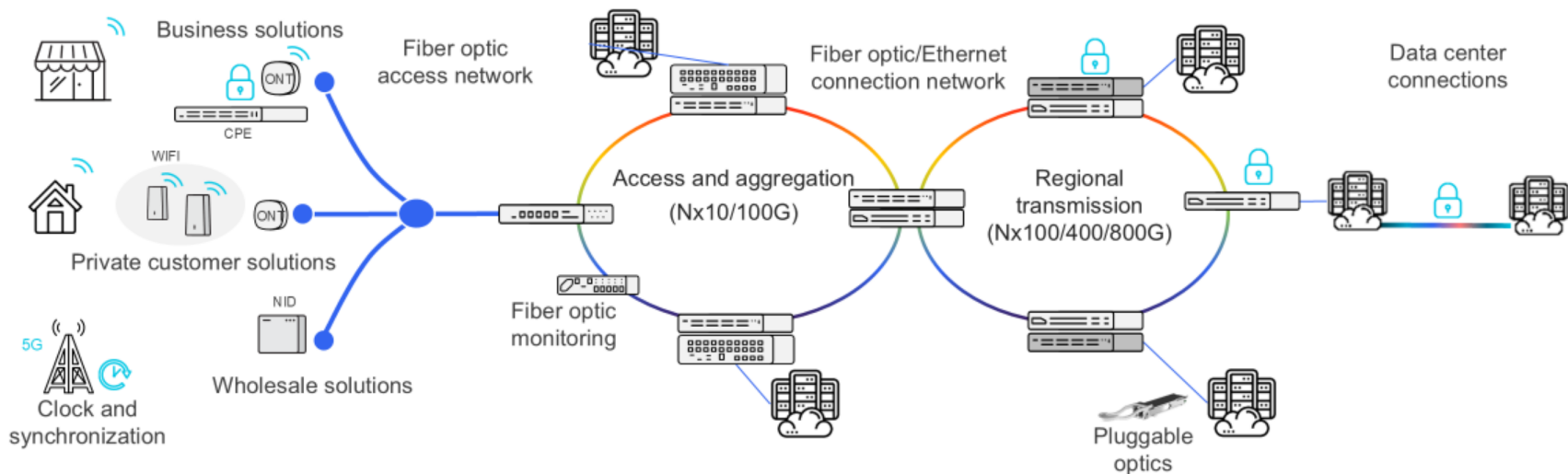
PQ Authentication in Optical Transport Networks: Deployment Experience and Open Questions

Ciaran Mullan, Richard Zink

IAB Workshop on Accelerating the Deployment of Post-Quantum Authentication (pqws)

Prague, October 11 + 12, 2026





What we build

Layer 1 (OTN) and Layer 2 (MACsec) encryptors

Where they run

Carriers, DCI, critical infrastructure, government and defence

Certified

BSI: VS-NfD, VS-V
NATO RESTRICTED
FIPS

Long-lived

20+ years in service;
updates follow long release cycles

Data plane

Done

AES-256-GCM
(but following NIST's wide-block cipher work)

Control plane

Deployed

Hybrid authenticated key exchange

Software authentication

In progress

Code signing: HSM, PKI, RSA verifiers in the field

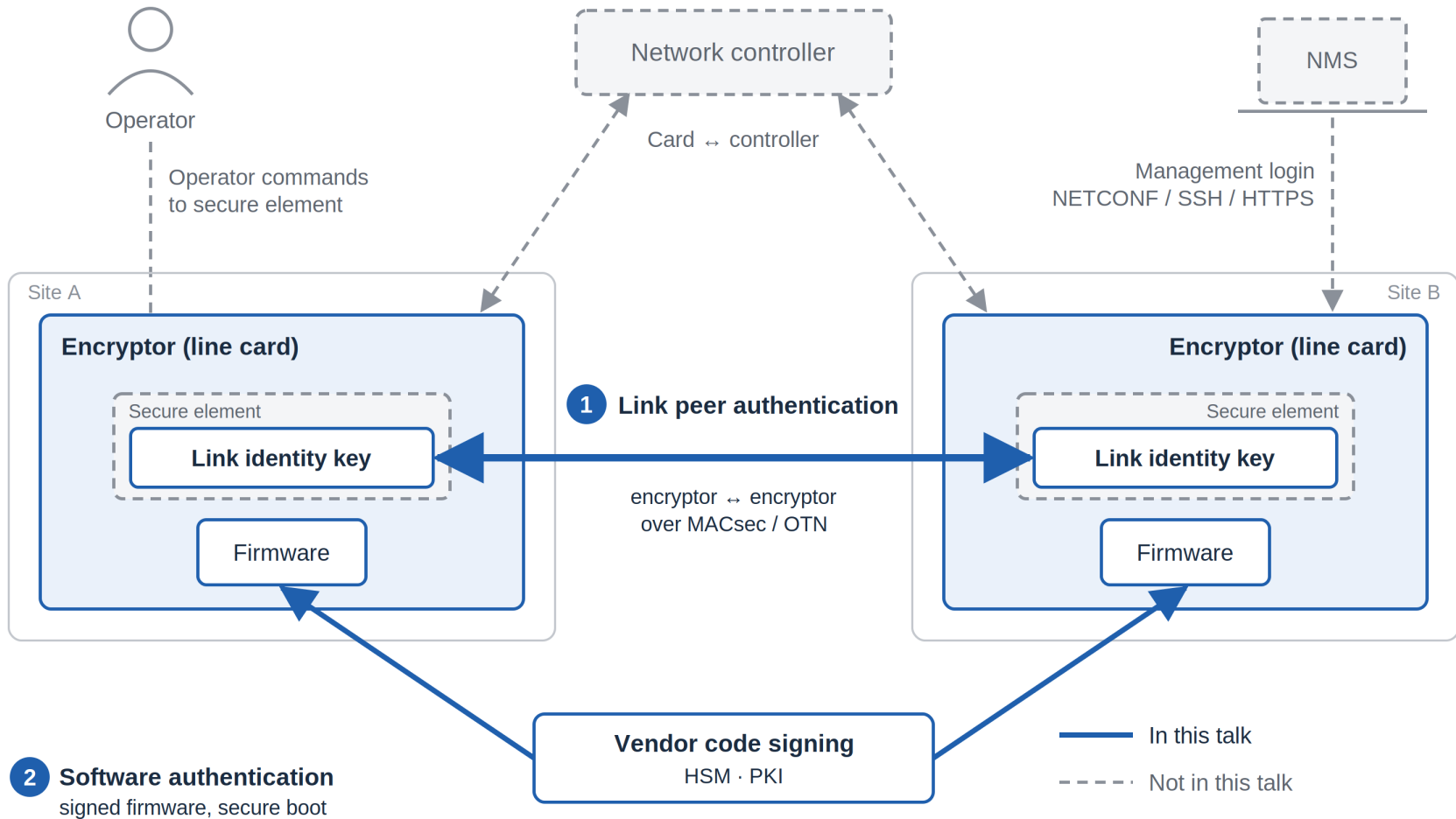
Management plane

In progress

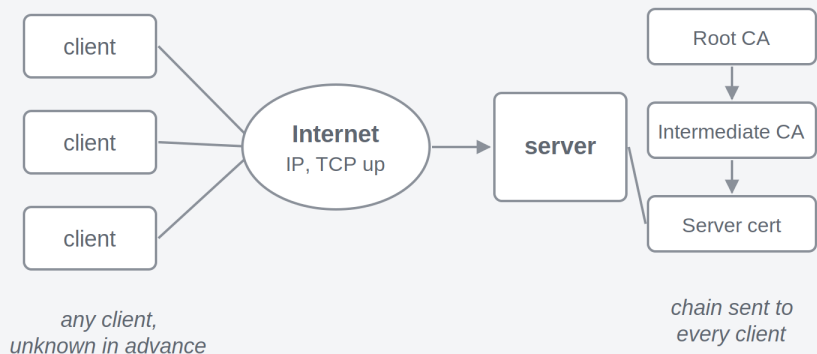
More to do

NETCONF, SSH, HTTPS, ...: migrate as and when stacks do

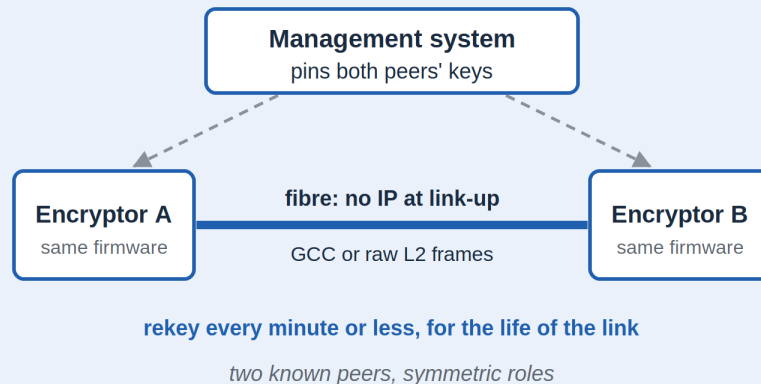
This talk: mainly PQ link authentication (deployed); code signing in brief (in progress)



Web PKI (TLS)



Our managed optical link



Algorithms

Negotiated per connection
client offers suites, server picks

Transport at start

IP and TCP already up
bandwidth is not the constraint

Session pattern

One handshake per connection
then the session ends

Announce and match

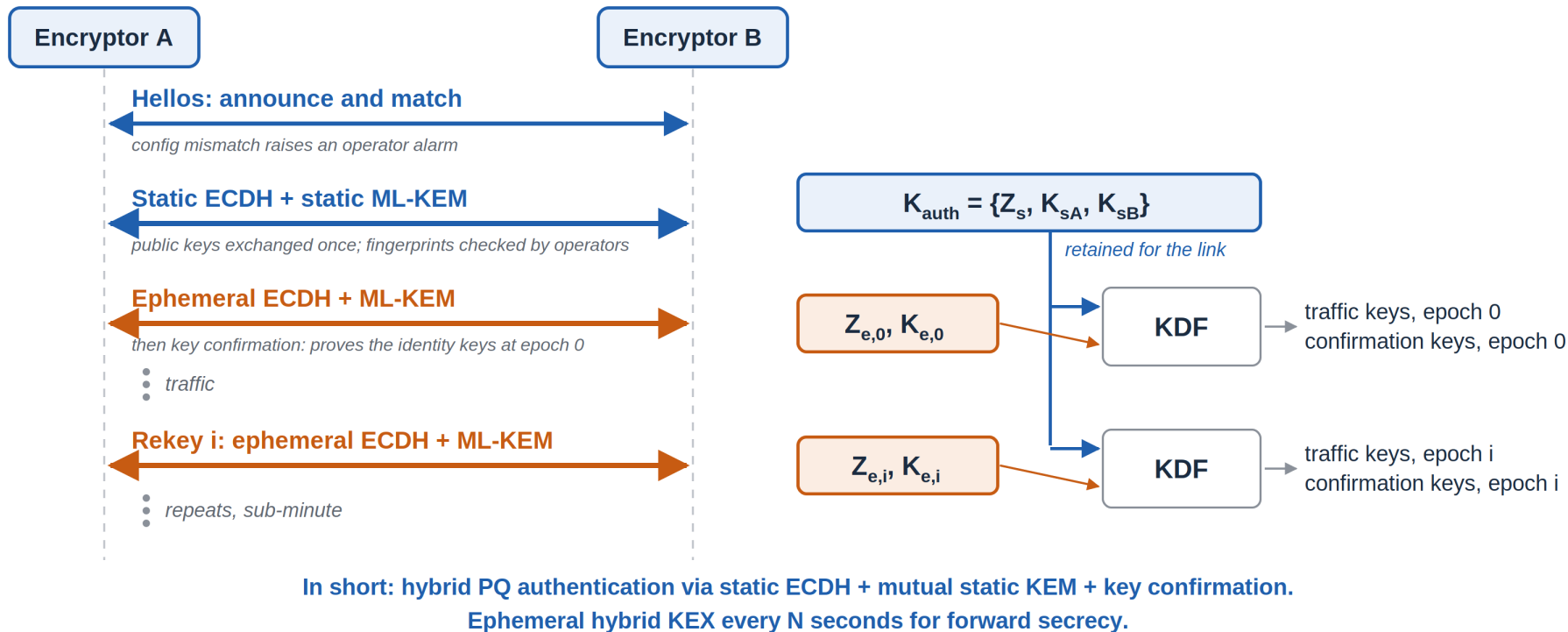
one provisioned suite; mismatch fails closed, no fallback

No IP at link-up

OTN GCC: 1.3 Mbit/s (10G) to about 14 Mbit/s (100G)
Ethernet: raw L2 frames, 1,500-byte payload limit
What counts: time to provision N Secure Associations, not one handshake

Continuous for the life of the link

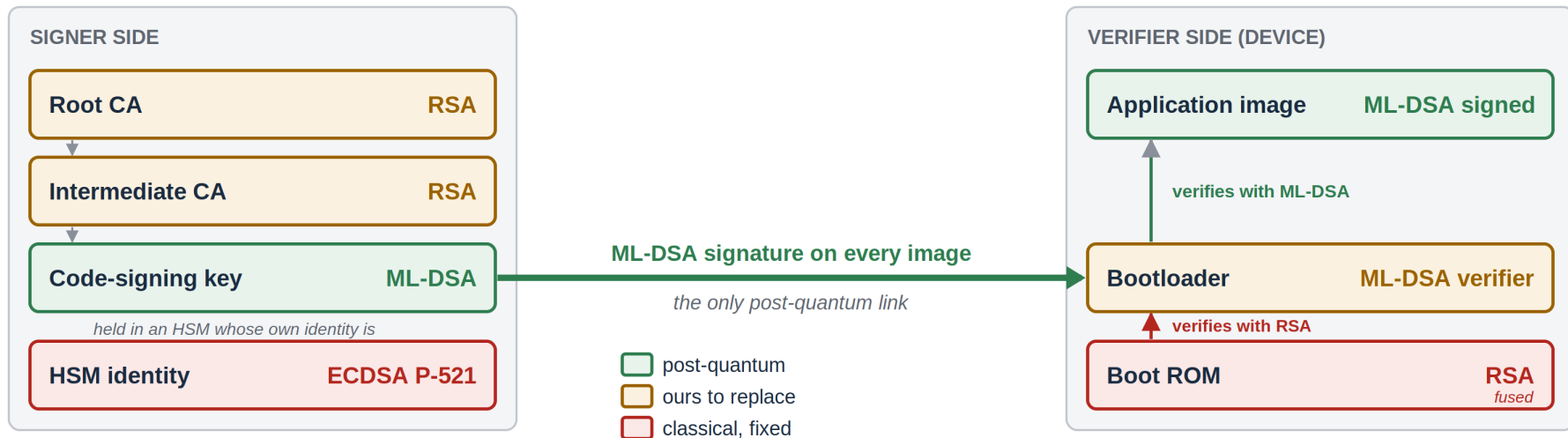
sub-minute rekeying at 100 GbE (AES-GCM usage limits)



Engineering considerations (per risk assessment)

1. Identity keys are proven once, at epoch 0. Rekeys prove only knowledge of K_{auth} .
2. Reboot or link down/up always reauthenticates and refreshes K_{auth} .
3. Our patch window is long. PQ + classical, or PQ + PQ + classical?

End to end: a PQ signature between two classical anchors

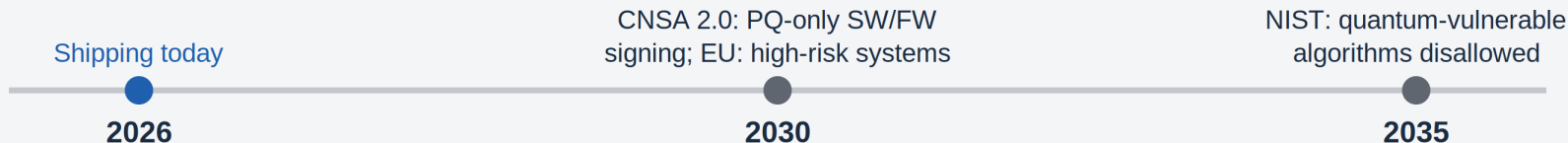


Discussion (held yesterday)

1. Device boot ROMs, ~20-year lifetimes
 - Current: fused hash of an RSA public key as root of trust
 - Future: fused hash of an ML-DSA public key instead? Alternatives?
2. Certificate chains: ML-DSA all the way vs mixed-algorithm chains
3. Our HSMs currently use vendor-defined PKCS#11 mechanisms for PQ schemes

Where this leaves us

TIMELINES



Lots more PQ work to do

Device identity (DevID)

Management logins

Intra-shelf links

Operator commands

...

Thank you

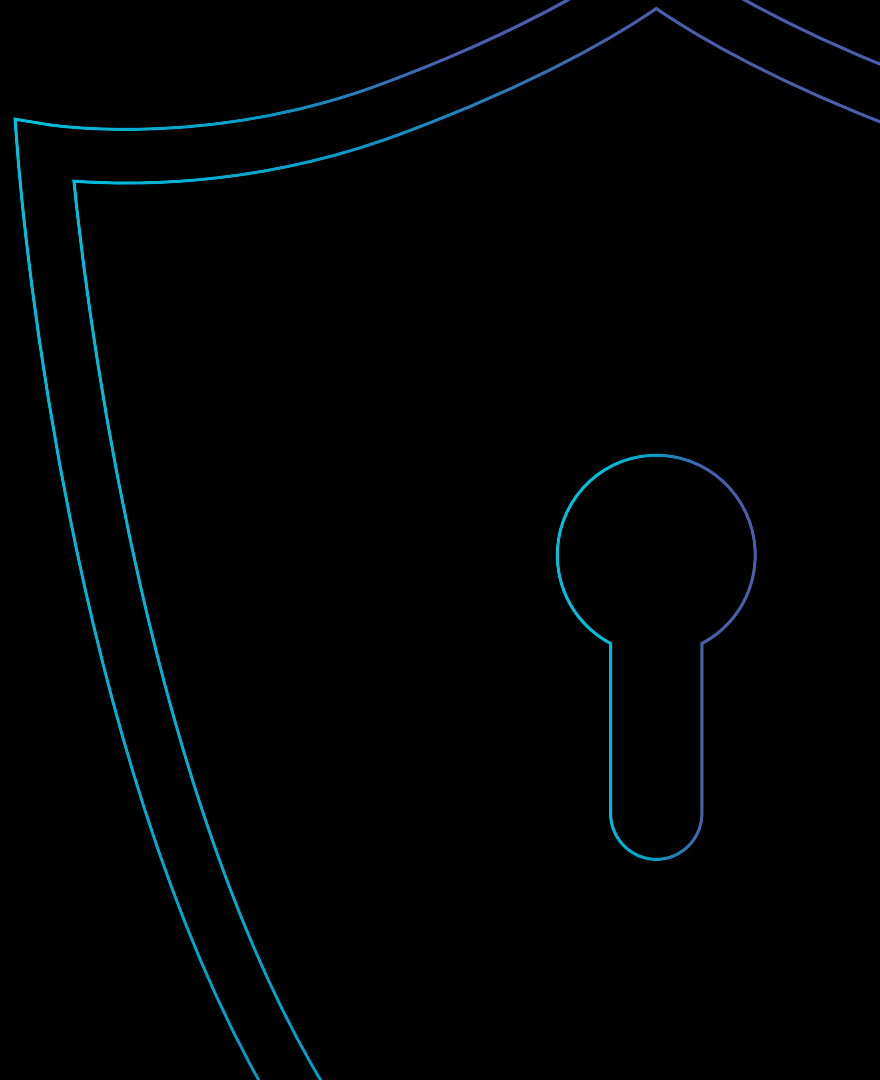
Keen to exchange ideas with anyone doing sub-IP PQ authentication,
or involved in related work: IETF, IEEE 802.1 (e.g. 802.1X, 802.1AR), ITU-T (OTN)

My address:

ciaran.mullan@advasecurity.com

Thank you

www.advasecurity.com



Provisioning the link identity keys: today, and at scale

Today: manual

Static public keys are exchanged once.

Operators at each end compare key fingerprints over the management channel.

Works, but manual and hard to scale: the PGP model.

At scale: certificates?

A KEM key cannot sign, so PKCS#10 proof-of-possession does not apply.

CMPv3 specifies KEM proof-of-possession; BRSKI-AE (RFC 9733) routes enrolment over CMP.

BRSKI (RFC 8995) assumes IP to a registrar at bootstrap; a dark fibre at link-up has none.

Discussion

1. Automated provisioning of peer keys at scale, without certificate machinery?