



Technische Herausforderungen der PQC Migration

Dr. Falko Strenzke, MTG AG

heise devSec() | 2026-09-22 | Marburg

MTG AG



- Gegründet 1995
- Darmstadt, Athen
- ERS Produkte: PKI, KMS, Lifecycle Management, HSM-Integration
- PQC Forschung
- DIN ISO 27001 & ISO 9001 zertifiziert



www.mtg.de/de/unternehmen/historie/



MTG

theben®



swissbit®

iskraemeco
BY ELSEWEDY ELECTRIC



Stromnetz
Hamburg



ciena

schleupen

devolo
The Network Innovation



TREMONDI

Stadtwerke
Saarbrücken

VIVAVIS
DECODING THE FUTURE



Syna

utimaco®
Creating Trust in the Digital Society



thyssenkrupp

bdr
BUNDESDRUCKEREI

inno2grid

Knappschaft Bahn See

LEAR
CORPORATION

EU PQC Transition Roadmap^a

- ▶ Anfang 2026: Finalisierung der wichtigsten PKI-Standards für PQC
- ▶ bis Ende 2026: “First Steps” in Mitgliedsstaaten umgesetzt
 - ▶ Cryptographic asset management
 - ▶ Quantum risk analysis (Administration und Kritis)
 - ▶ ...
- ▶ bis Ende 2030:
 - ▶ Abschluss PQC-Migration für Hochrisiko-Anwendungen
 - ▶ PQC Firmware Updates
 - ▶ Start Migration Medium-Risk Anwendungen
- ▶ bis Ende 2035:
 - ▶ Abschluss PQC-Migration für Medium-Risk Anwendungen

^a“A Coordinated Implementation Roadmap for the Transition to Post-Quantum Cryptography” Part 1, Version: 1.1, EU PQC Workstream, 11.06.2025

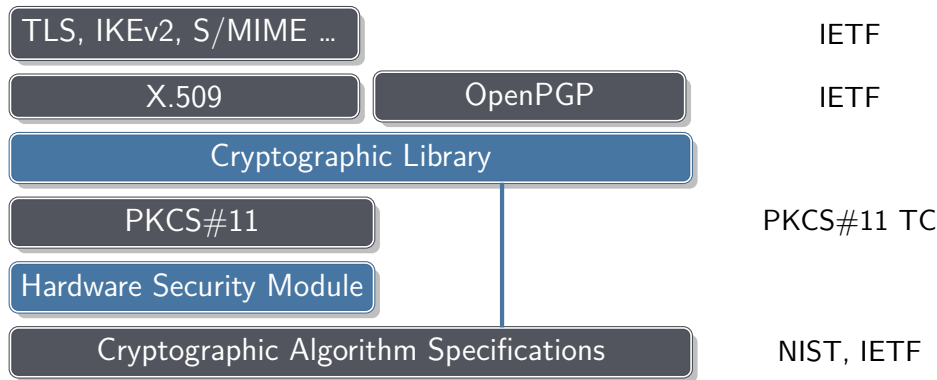
Technische Aspekte der PQC Migration

- ▶ Algorithmen
 - ▶ PQ-Signaturen
 - ▶ HSM-Unterstützung
 - ▶ Hybride (multi-algorithm) Signaturen
 - ▶ “Interface-Verhalten” der PQC Verfahren
 - ▶ PQ-Verschlüsselung
- ▶ Aspekte
 - ▶ Bekannt: Artefakt-Größen, Performanz
 - ▶ Stateful Keys
 - ▶ Signature Count Limit bei NIST SP 800-230
 - ▶ Weniger bekannt: Konsequenzen der Abwendung von Hash-then-sign Paradigma
 - ▶ Migration komplexer PKI
 - ▶ Strategischer Umgang mit technischen Herausforderungen

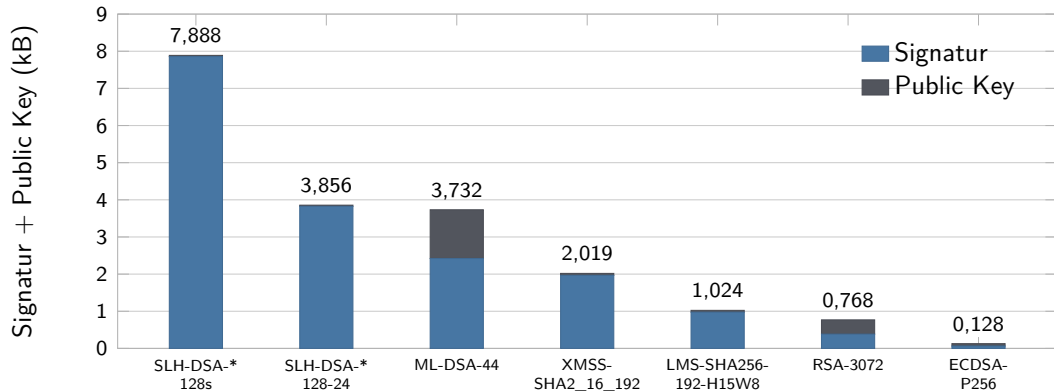
Die Standardisierten PQC-Verfahren

- ▶  KEM (Verschlüsselung)
 - ▶ ML-KEM 
 - ▶ ML-KEM-composite  
- ▶  Signaturen
 - ▶ ML-DSA (FIPS-204) 
 - ▶ ML-DSA-composite  
 - ▶ SLH-DSA (FIPS-203) **#**
 - ▶ SLH-DSA-2²⁴ (NIST SP 800-230 IPD) **#** 
 - ▶ XMSS / LMS **#** 

“Kryptografie-Stack”



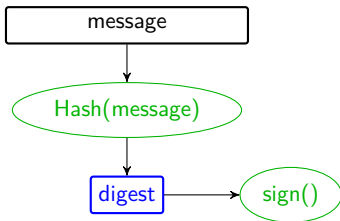
Signaturverfahren - Quantitative Größeneffekte



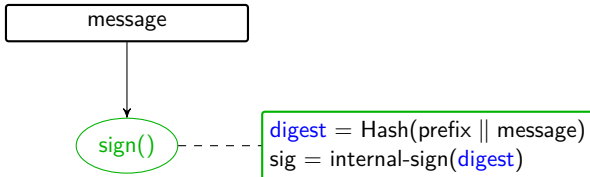
Ungefähre Größen für das 128-Bit-Sicherheitsniveau; XMSS- und LMS-Parameter mit 192-Bit-Sicherheit.

Signatures: From “hash-then-sign” to “pure signing”

- ▶ traditional algorithms RSA, ECDSA:
 - ▶ build by Hash-then-sign paradigm
 1. hash the message stream
 2. sign the hash



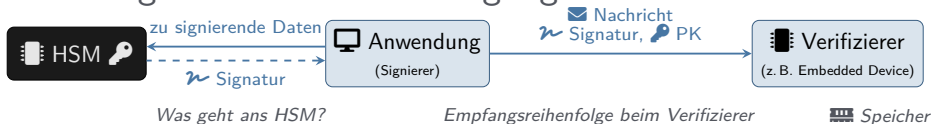
-
- ▶ modern / PQC design paradigm is not hash-then-sign
 - ▶ advantage: increased collision resistance of signature scheme
 - ▶ PQC signature algorithms receive the message stream directly



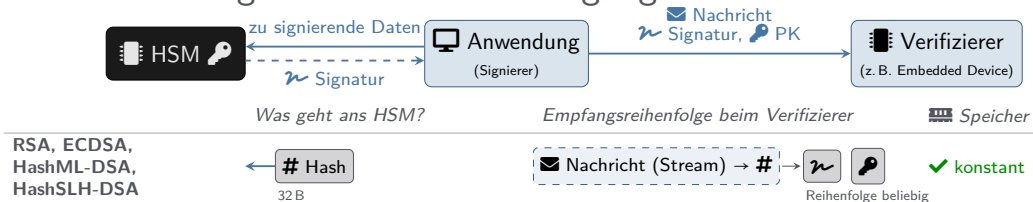
Signaturverfahren – Qualitative Größeneffekte

- ▶ Bisherige Signaturverfahren RSA, ECDSA
 - ▶ Hash-then-sign (→ “Hash-then-verify”)
- ▶ PQ-Signaturverfahren
 - ▶ “Direct Signing”
 - ▶ 2 Effekte:
 - ▶ Übertragung ganzer Nachricht statt nur Hash beim Signieren
 - ▶ Solange “prefix” nicht bekannt ist, kann der Verifizierer nicht mit dem Hashen beginnen

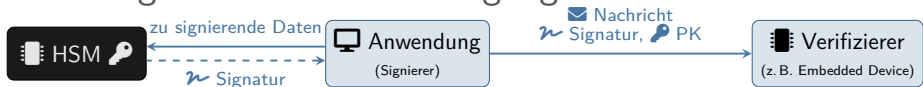
Signaturanwendungen und Datenübertragung



Signaturanwendungen und Datenübertragung



Signaturanwendungen und Datenübertragung

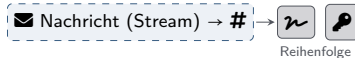


Was geht ans HSM?

Empfangsreihenfolge beim Verifizierer

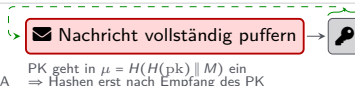
Speicher

RSA, ECDSA,
HashML-DSA,
HashSLH-DSA



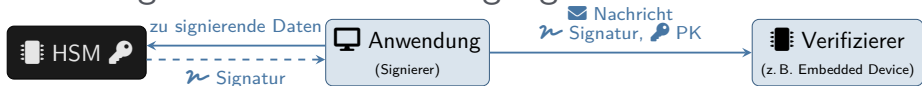
✓ konstant

ML-DSA



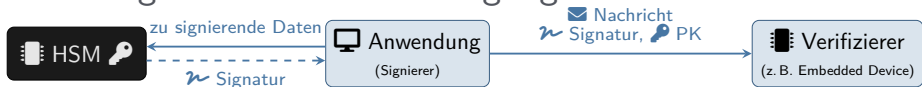
⚠ ganze Nachricht
im RAM
gestrichelt (🔑 vor ✉️):
Streaming, ✓ konstant

Signaturanwendungen und Datenübertragung



	Was geht ans HSM?	Empfangsreihenfolge beim Verifizierer	Speicher
RSA, ECDSA, HashML-DSA, HashSLH-DSA	Hash 32 B	Nachricht (Stream) → → Reihenfolge beliebig	konstant
ML-DSA	μ 64 B; $\mu = H(H(pk) \parallel M)$ extern berechenbar oder HashML-DSA	Nachricht vollständig puffern → $PK \text{ geht in } \mu = H(H(pk) \parallel M) \text{ ein}$ $\Rightarrow$ Hashen erst nach Empfang des PK	ganze Nachricht im RAM gestrichelt (vor): Streaming, konstant
XMSS, LMS	ganze Nachricht MB ... GB; im HSM 1-pass streambar	Nachricht vollständig puffern → Randomizer aus der Signatur geht in den Hash ein $\Rightarrow$ Hashen erst nach Empfang der Signatur	ganze Nachricht im RAM gestrichelt (vor): Streaming, konstant

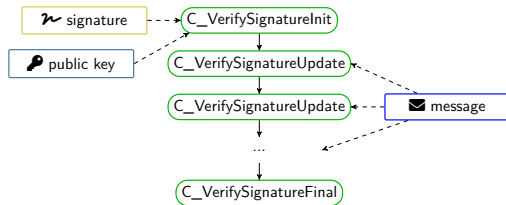
Signaturanwendungen und Datenübertragung



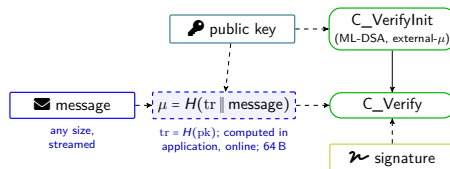
	Was geht ans HSM?	Empfangsreihenfolge beim Verifizierer	Speicher
RSA, ECDSA, HashML-DSA, HashSLH-DSA	# Hash 32 B	Nachricht (Stream) → # → Reihenfolge beliebig	✓ konstant
ML-DSA	μ 64 B; $\mu = H(H(pk) \parallel M)$ extern berechenbar oder HashML-DSA	Nachricht vollständig puffern → PK geht in $\mu = H(H(pk) \parallel M)$ ein $\Rightarrow$ Hashen erst nach Empfang des PK	⚠ ganze Nachricht im RAM gestrichelt (vor): Streaming, ✓ konstant
XMSS, LMS	ganze Nachricht MB ... GB; im HSM 1-pass streambar	Nachricht vollständig puffern → Randomizer aus der Signatur geht in den Hash ein $\Rightarrow$ Hashen erst nach Empfang der Signatur	⚠ ganze Nachricht im RAM gestrichelt (vor): Streaming, ✓ konstant
SLH-DSA (ohne Pre-Hash)	ganze Nachricht 2-pass: $R = \text{PRF}(\text{SK.prf}, M)$, dann $H_{\text{msg}}(R, \dots, M)$ ganze Nachricht auch im HSM-RAM	Nachricht vollständig puffern → wie XMSS / LMS	⚠ ganze Nachricht im RAM gestrichelt (vor): Streaming, ✓ konstant

Umsetzung in PKCS#11 v3.2

PKCS#11 v3.2: C_VerifySignature

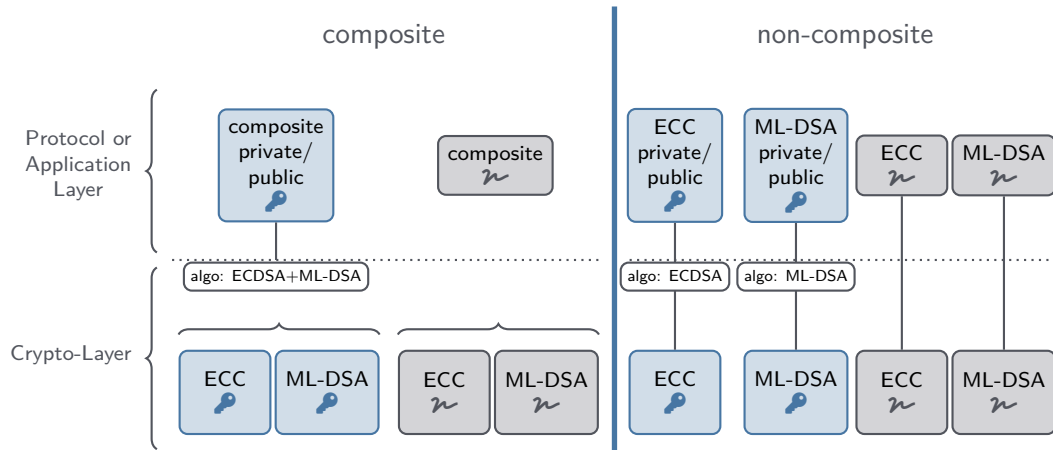


hypothetisch (v3.3): external- μ

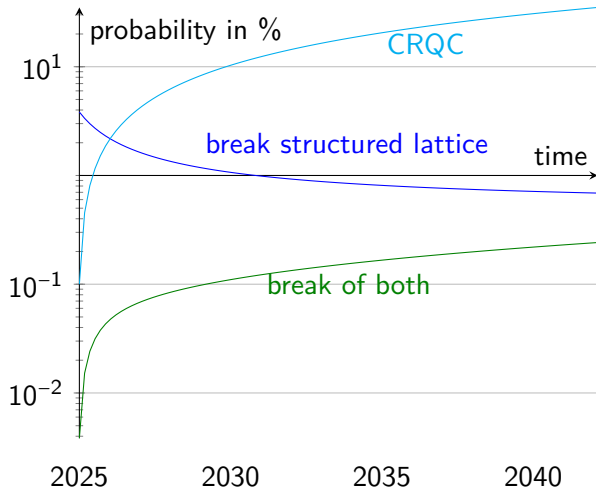


Further reading: F. Strenzke: Scaling of Memory and Bandwidth Requirements of Post-Quantum Signatures with Message Size, <https://eprint.iacr.org/2026/617>

PQ/T Hybride bzw. Composite Verfahren



Hybride Structured-Lattice-Verfahren



- ▶ ML-DSA+ECDSA
- ▶ Problem: Langzeitsicherheit
- ▶ CRQC: RSA/ECC fällt weg
- ▶ ML-* ausreichend?
- ▶ Risiko: Hybrid muss nach Schwächung einer Komponente ausgetauscht werden

Hybride Structured-Lattice-Verfahren

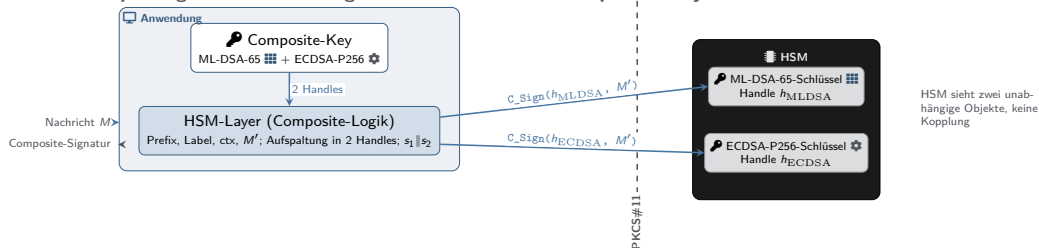
- ▶ “No-regret move” / BSI-Empfehlung
- ▶ ECC Komponente vernachlässigbar bzgl. Bandbreite und Performanz
- ▶ Risikoreduktion für relativ effiziente ML-*
- ▶ Unterschiedliche Bedrohung der beiden Komponenten
 - ▶ Annahmen:
 - ▶ Auch nach dem “Q-Day” bleiben CRQC-Angriffe teuer
 - ▶ Neuer Angriff auf ML-* könnte viel besser skalieren
- ▶ Aber
 - ▶ bietet nicht unbedingt Langzeitsicherheit
 - ▶ zukünftige Unterstützung in Komponenten unklar
 - ▶ Hybride Signaturen finden weniger Unterstützung als hybride Verschlüsselung
 - ▶ Auch Kritik an der Art dem kryptografischen Design der ML-DSA-Hybride für X.509
 - ▶ Neue Vorschläge existieren bereits in der IETF^b
 - ▶ HSM-Verwendung erfordert softwareseitige Kombination der Verfahren^c

^bdraft-devevey-cfrg-silithium

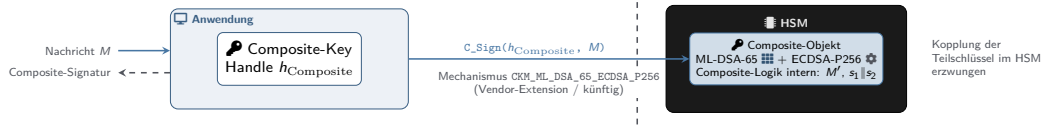
^cZukünftige PKCS#11 Versionen

Composite Verfahren auf HSM

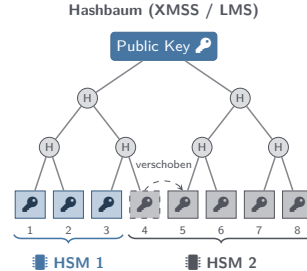
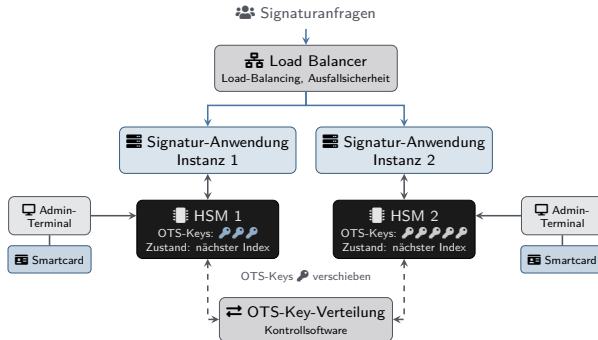
A: Aufspaltung in der Anwendung – HSM kennt nur die Component-Keys



B: Composite nativ in PKCS#11 – HSM führt das Composite-Verfahren aus



Einsatz von Stateful-hash-based



OTS = One-Time-Signature-Schlüssel (Blätter): jeder nur einmal verwendbar
 ⇒ Zustand (nächster Index). Verteilung der Blätter auf die HSMs
 = verteilter Zustand; die Kontrollsoftware verschiebt ganze Blätter.

State-Handling: Backup und Hochverfügbarkeit

- ▶ Schlüssel zustandsbehafteter Hash-Signaturen (SHBS) müssen in kryptografischen Modulen liegen (NIST SP 800-208)
 - ▶ Export nicht erlaubt (NIST)
- ▶ Das State-Handling betrifft Backup- und Hochverfügbarkeits-Setups
- ▶ Zwei Standards:
 - ▶ NIST SP 800-208 (von 2020, Aktualisierung erwartet)
 - ▶ IETF draft-ietf-pquip-hbs-state (in der WG abgeschlossen)

State-Handling: NIST SP 800-208

- ▶ NIST SP 800-208 (2020) spezifiziert verteilte Multi-Trees
 - ▶ HSS oder XMSSMT mit 2 Ebenen
 - ▶ Wurzelbaum (Public Key) von allen HSMs gemeinsam genutzt
 - ▶ signiert die Teilbäume
 - ▶ Ein Teilbaum pro HSM
 - ▶ signiert die Nachrichten
 - ▶ Erscheint nach außen als ein einziges Schlüsselpaar
- ▶ — Größere Signaturen durch den Multi-Tree
- ▶ Umsetzung ...
 - ▶ Für HSS mit Instanzen von Standard-LMS realisierbar
 - ▶ Für XMSSMT ist eine angepasste Implementierung nötig

XMSS^{MT} Merkle Tree Structure (Two Tree Levels)

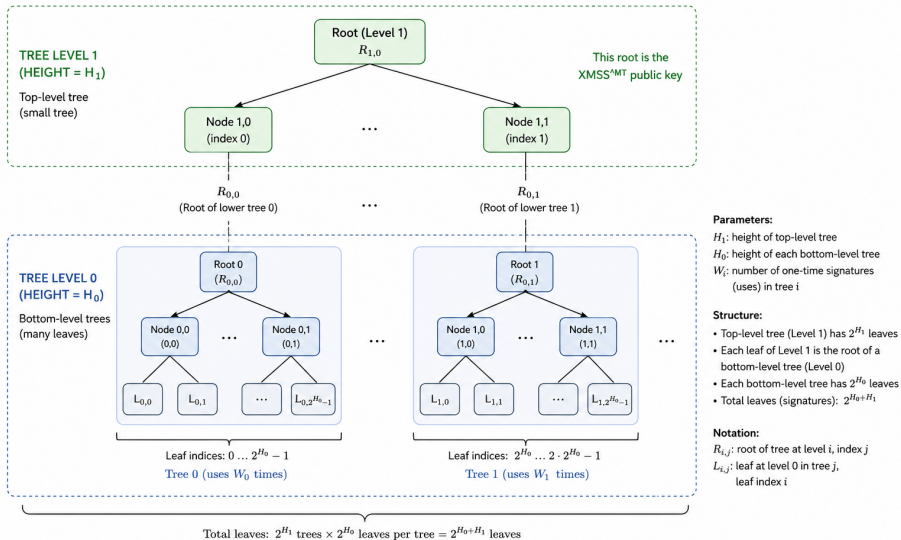
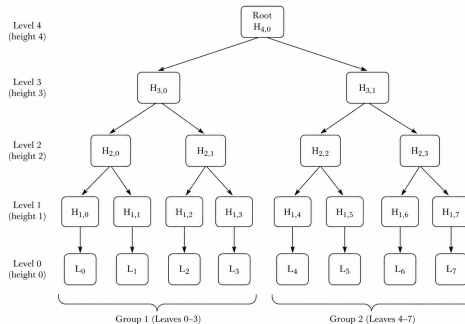


Figure: XMSS-Multi-tree

State-Handling: draft-ietf-pquip-hbs-state I

- ▶ IETF draft-ietf-pquip-hbs-state mit zwei Autoren vom BSI
- ▶ Weitere Verfahren:
 - ▶ *Sektorisierung*
 - ▶ ähnlich zu *Distributed Multi-Trees*
 - ▶ benötigt keine mehreren Bäume: die OTS-Keys werden in Sektoren aufgeteilt und auf verschiedene HSMs verteilt
 - ▶ + keine größeren Signaturen, anders als bei *Distributed Multi-Trees*



State-Handling: draft-ietf-pquip-hbs-state II

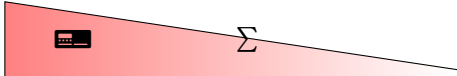
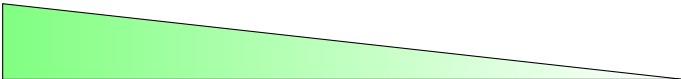
- ▶ *Partial Key Transfer*: dynamische Sektorisierung + sicherer Transfer von OTS-Keys im laufenden Betrieb
 - ▶ Von einigen HSMs implementiert
 - ▶ HSMs merken sich die gelöschten OTS-Keys
 - ▶ Import von OTS-Keys nur auf einem einzigen, vorab festgelegten Gerät möglich
 - ▶ (die Geräte werden vorher kryptografisch verknüpft)
- ▶ Andere HSMs haben eine ähnliche Lösung
 - ▶ Geräte einer HA-Gruppe tauschen Teilbäume automatisch aus
- ▶ 💡 Solche herstellergestützten Lösungen sind die einzigen praktisch relevanten
- ▶ Hinweis: Beide verstoßen gegen das Exportverbot für Schlüssel in der aktuellen NIST SP 800-208

State-Handling: draft-ietf-pquip-hbs-state III

Weitere Verfahren

- ▶ *Zeitbasierte, intervallbasierte, vorab zugewiesene Zustände*
 - ▶ Gefährlich oder unpraktikabel
- ▶ *Mehrere Public Keys, Public-Key-Rotation*
 - ▶ Verlagert die Last auf die PKI
- ▶ draft-ietf-pquip-hbs-state beschreibt außerdem ein Verfahren für ein dediziertes externes Backup der untersten Baumebene

Kriterien für PQC Signaturen^e

	ML-DSA hybride	XMSS / LMS	SLH-DSA-2 ^{24d}	SLH-DSA
key-mgmt-complexity				
Long-term security				
Performance				
Signature size				

^d SLH-DSA mit kleineren Parametern mit begrenzter Signaturanahl. NIST SP 800-230 (IPD) .

^e Mit Ausnahme von SLH-DSA-2²⁴ sind dies die Signaturverfahren mit erwarteter kurzfristiger Verfügbarkeit in X.509.

PQC Verschlüsselung

- ▶ Nur ein “Europa-taugliches” standardisiertes PQC-Verfahren: ML-KEM hybride
 - ▶ RFC vermutlich in einigen Monaten
- ▶ Store-now-decrypt-later erfordert schnelles Handeln
- ▶ Standardisierung von weiteren PQC Verfahren für X.509:
 - ▶ in der IETF
 - ▶ FrodoKEM ?
 - ▶ Classic McEliece ?

Confidentiality vs. Authentication

- ▶ Store-now-decrypt-later sorgt für Confidentiality-Problem
 - ▶ ggf. bereits verlorenes Rennen
 - ▶ Risiko-Abschätzung
- ▶ Sichere Authentisierung oftmals wichtiger
 - ▶ Beispiel: monetäre Transaktionen
- ▶ “Authentisierung muss erst PQC-sicher sein, wenn CRQC existieren”
 - ▶ Gilt nur für kryptografische Authentisierung
 - ▶ Store-now-decrypt-later gefährdet verschlüsselt übertragene Authentisierungs-Tokens (Passwörter)
 - ▶ Zur Migration gehört die rechtzeitige PQ-sichere Erneuerung von Authentisierungs-Tokens

TLS mit PQC-Verschlüsselung

Chromium mit Google Server

Security overview



This page is secure (valid HTTPS).



Certificate - **valid and trusted**

The connection to this site is using a valid, trusted server certificate issued by WE2.

[View certificate](#)



Connection - **secure connection settings**

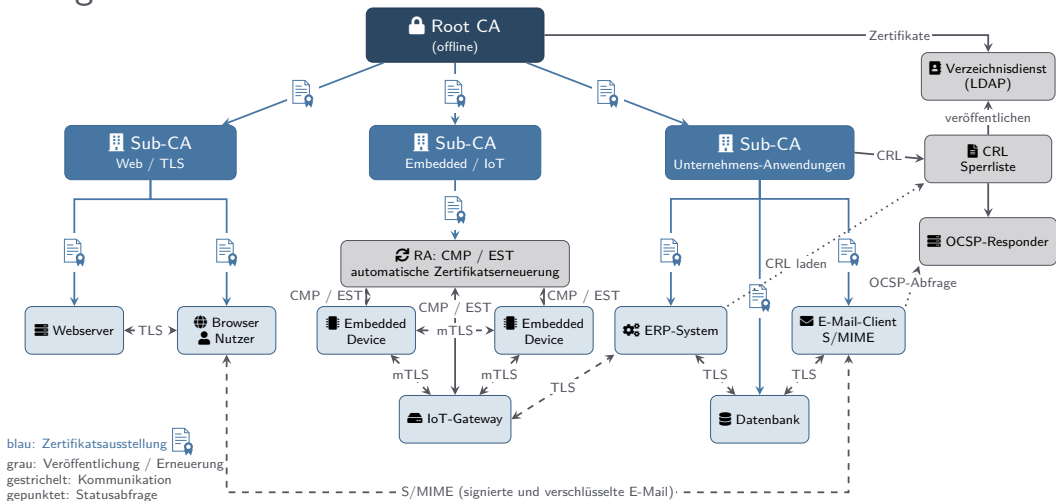
The connection to this site is encrypted and authenticated using TLS 1.3, X25519MLKEM768, and AES_128_GCM.

PQC VPN

- ▶ Proprietäre PQC-Lösungen verschiedener Hersteller
- ▶ IKEv2 with MLKEM in Arbeit bei IETF^f

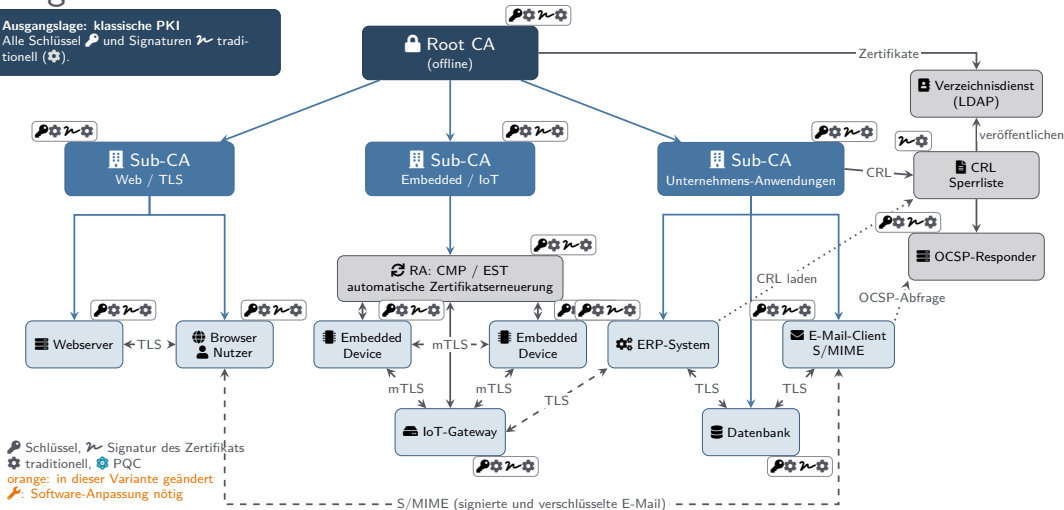
^f [draft-ietf-ipsecme-ikev2-mlkem](#)

PQC Migration für PKI



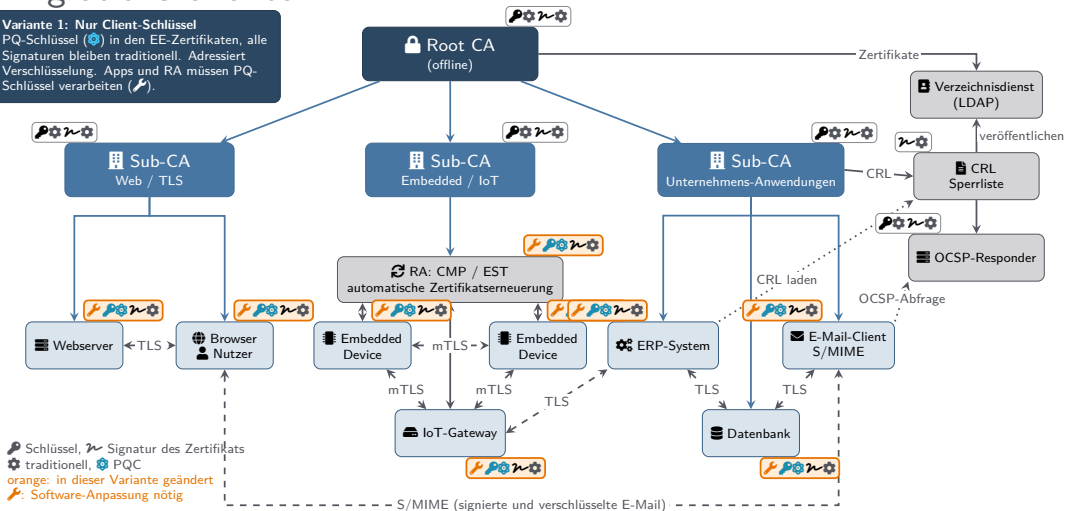
PKI Migrationsvarianten

Ausgangslage: klassische PKI
Alle Schlüssel  und Signaturen  traditionell ().



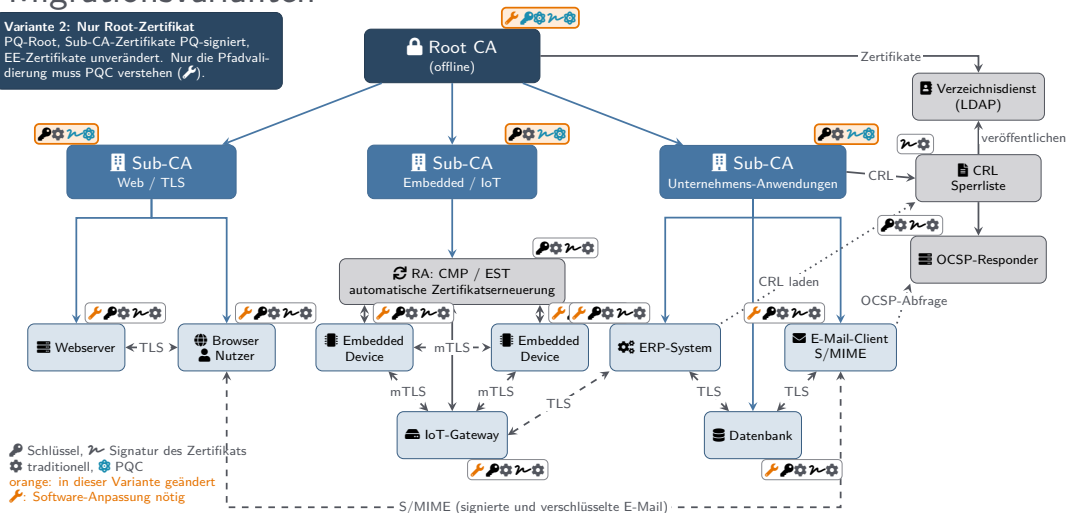
PKI Migrationsvarianten

Variante 1: Nur Client-Schlüssel
PQ-Schlüssel (🔑) in den EE-Zertifikaten, alle Signaturen bleiben traditionell. Adressiert Verschlüsselung. Apps und RA müssen PQ-Schlüssel verarbeiten (🔧).



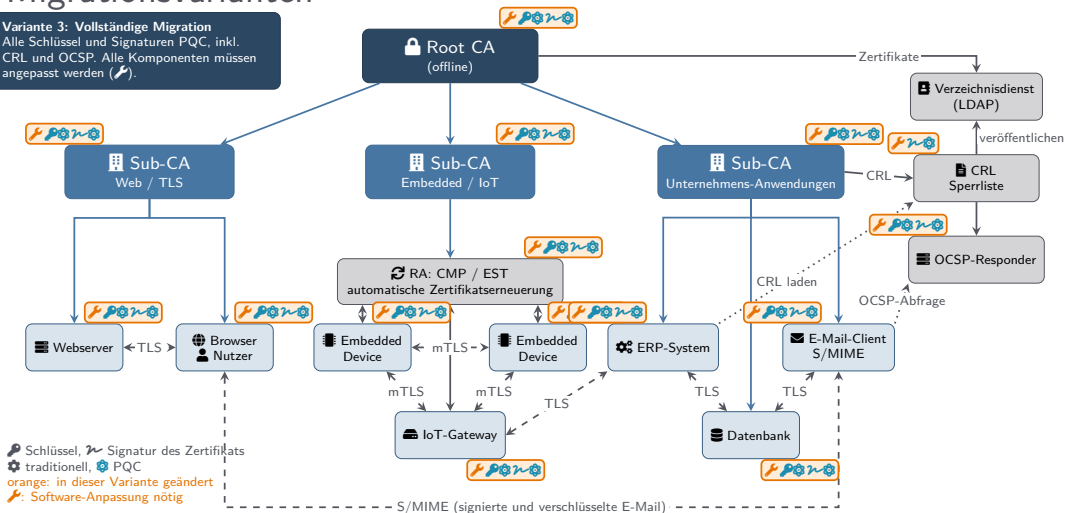
PKI Migrationsvarianten

Variante 2: Nur Root-Zertifikat
PQ-Root, Sub-CA-Zertifikate PQ-signiert, EE-Zertifikate unverändert. Nur die Pfadvalidierung muss PQC verstehen (🔧).

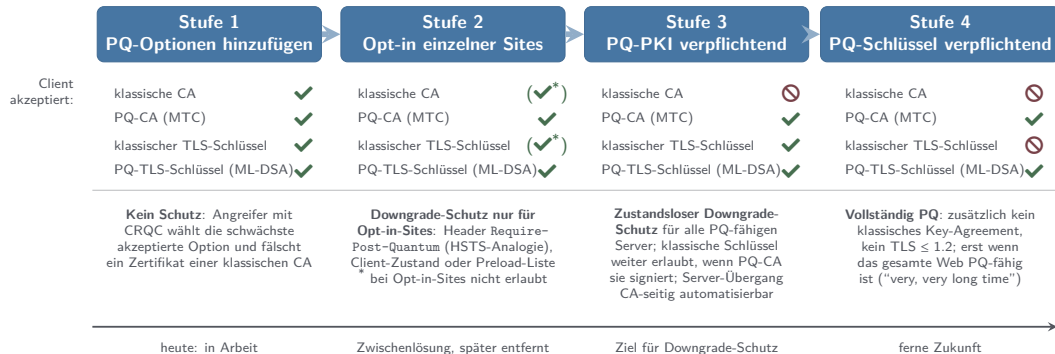


PKI Migrationsvarianten

Variante 3: Vollständige Migration
Alle Schlüssel und Signaturen PQC, inkl. CRL und OCSP. Alle Komponenten müssen angepasst werden (🔧).

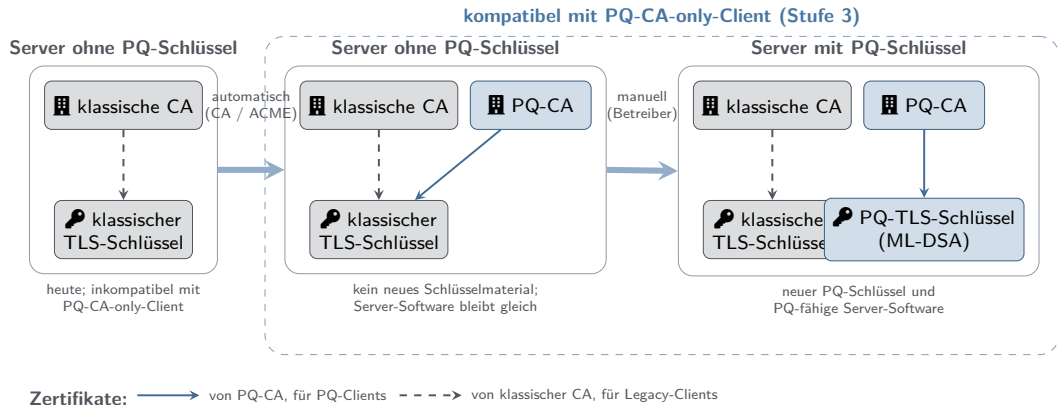


Chrome-Roadmap: PQ-Authentisierung in HTTPS^g



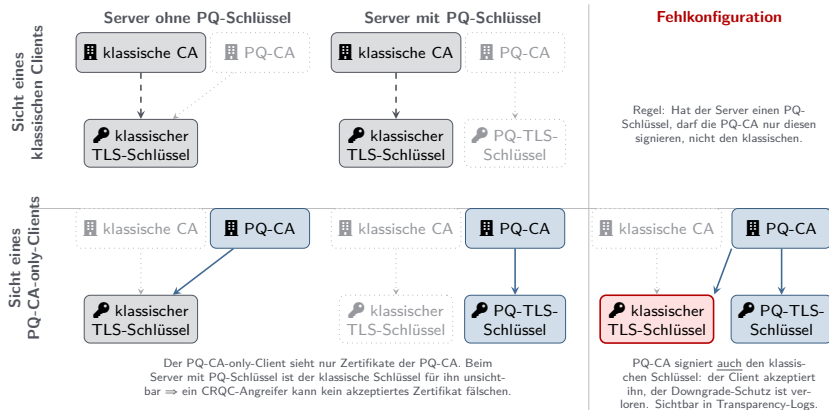
^g [Chromium: Post-Quantum HTTPS Authentication Roadmap](#), D. Benjamin, J. DeBlasio, 27.02.2026. Grafiken in Anlehnung an die Diagramme der Seite.

Stufe 3 ("root=PQC"): Übergang der Server^h



^hChromium: [Post-Quantum HTTPS Authentication Roadmap](#), D. Benjamin, J. DeBlasio, 27.02.2026. Grafiken in Anlehnung an die Diagramme der Seite.

Stufe 3 ("root=PQC"): Client-Sicht und Downgrade-Schutzⁱ



gepunktet/grau: für diesen Client unsichtbar; blau durchgezogen: Zertifikat der PQ-CA; grau gestrichelt: Zertifikat der klassischen CA

ⁱ Chromium: [Post-Quantum HTTPS Authentication Roadmap](#), D. Benjamin, J. DeBlasio, 27.02.2026. Grafiken in Anlehnung an die Diagramme der Seite.

OpenSSL: PQC-Unterstützung – aktueller Stand^j

Algorithmus	Bibliothek (EVP, CLI)	X.509	IANA (TLS)	RFC / Draft (TLS)	TLS 1.3
ML-KEM 512/768/1024	✓ seit 3.5	✓ RFC 9935 (CMS) RFC 9936, seit 3.6)	✓ Groups 0x0200–0x0202; Hybride 0x11EB– 0x11EE	Hybride: RFC 10024 ; rein: RFC-Editor-Queue; SM2-Hybrid: individual, expired	✓ MLKEM512/768/1024; X25519MLKEM768 (Default), SecP256r1-, SecP384r1-, curveSM2-Hybride (4.0)
ML-DSA 44/65/87	✓ seit 3.5	✓ RFC 9881; Zertifikate, CSRs, CRLs	✓ Sigalgs 0x0904–0x0906	draft-ietf-tls-mldsa: RFC-Editor-Queue	✓ Sigalgs mldsa65/87/44, in Default-Liste vorn
SLH-DSA	✓ seit 3.5	✓ RFC 9909; Zertifikate, CSRs, CRLs	✓ Sigalgs 0x0911–0x091C	draft-reddy-tls-slhdsa: individual, expired	✗ keine TLS-Sigalgs
LMS / HSS	(✓) nur Verifikation (3.6), Build-Option enable-lms	✗ RFC 9802 nicht umgesetzt (nur SPKI, 4.0)	✗	kein TLS-Draft	✗
XMSS / XMSS^{MT}	✗ nicht implementiert (Issue #21360)	✗ RFC 9802 nicht umgesetzt	✗	kein TLS-Draft	✗
Composite ML-DSA	✗ Issue #26121 offen seit Dez. 2024	✗ LAMPS-Draft in RFC-Editor-Queue	✗ Codepoints “TBD”	draft-reddy-tls- composite-mldsa: individual	✗
FN-DSA (Falcon)	✗ kein Issue, kein Branch	✗ LAMPS-WG-Draft (Mai 2026)	✗	kein TLS-Draft	✗
HQC	(✓) Feature-Branch; wartet auf NIST	✗	✗	kein TLS-Draft	✗

^j Stand Sept. 2026: OpenSSL 3.5 LTS (Apr. 2025), 3.6, 4.0 (Apr. 2026), 4.1.0-alpha1 (Sept. 2026). Stufen: individual → WG-Draft → RFC-Editor-Queue → RFC. Weitere Verfahren: [oqs-provider](#). Quellen: Folie “Quellen”.

✓ RFC 9980
BSI, MTG, Proton

algorithms	requ.	security
ML-KEM-768+X25519	MUST	192 / 128
ML-KEM-1024+X448	SHOULD	256 / 224
ML-DSA-65+Ed25519	MUST	192 / 128
ML-DSA-87+Ed448	SHOULD	256 / 224
SLH-DSA-SHAKE-128s	MAY	128
SLH-DSA-SHAKE-128f	MAY	128
SLH-DSA-SHAKE-256s	MAY	256

⚙️ draft-ietf-openpgp-nist-bp-comp
in WGLC
BSI, MTG, NIST

algorithms	security
ML-KEM-512+ECDH- NIST-P-256	128 / 128
ML-KEM-768+ECDH- NIST-P-384	192 / 192
ML-KEM-1024+ECDH- NIST-P-384	256 / 192
ML-KEM-768+ECDH- brpP256r1	192 / 128
ML-KEM-1024+ECDH- brpP384r1	256 / 192
ML-DSA-44+ECDSA- NIST-P-256	128 / 128
ML-DSA-65+ECDSA- NIST-P-384	192 / 192
ML-DSA-87+ECDSA- NIST-P-384	256 / 192
ML-DSA-65+ECDSA- brpP256r1	192 / 128
ML-DSA-87+ECDSA- brpP384r1	256 / 192

all "MAY"

OpenPGP PQC Rollout

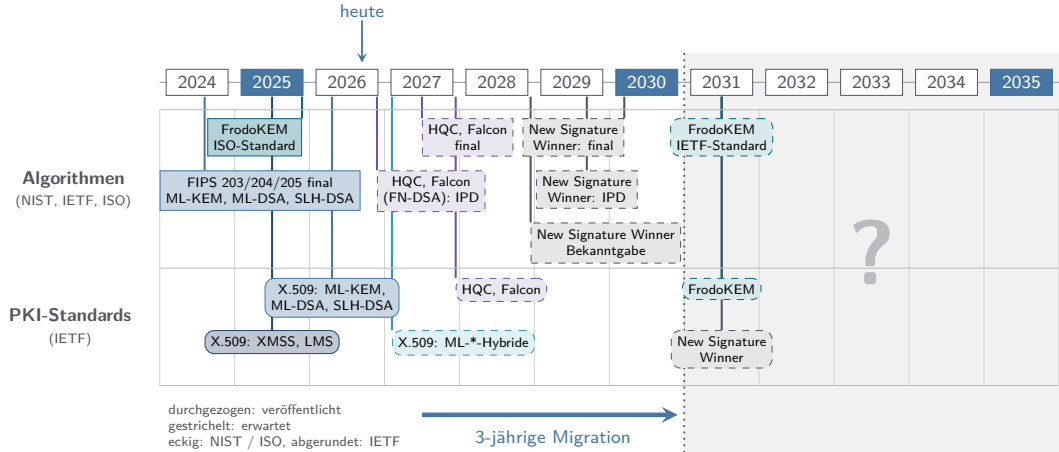
- ▶ OpenPGP Draft mit ML-KEM und ML-DSA hybride, SLH-DSA im Publikationsprozess
- ▶ Proton bereitet PQC Rollout vor
- ▶ Erste Feldtests:
 - ▶ Thunderbird “verschluckt” sich an PQC Artefakten
- ▶ Wichtiger Schritt: Graceful failure
 - ▶ Legacy Implementierungen mit PQC Artefakten konfrontieren

Strategische Aspekte der Migration

- ▶ Technische Abhängigkeiten
 - ▶ organisationsintern
 - ▶ von Vendors
 - ▶ Algorithmenauswahl?
 - ▶ Performanz?
 - ▶ Release-Zeitpunkte?
 - ▶ von Open Source
 - ▶ Gerade große Organisationen sollten aktiv werden bei den Vendors
- ▶ Priorisierung vs. Gesamtplanung
 - ▶ Asset-Modellierung ist Grundlage
 - ▶ Priorisierung nach Wichtigkeit ist sinnvoll, führt aber ggf. zu einem verzerrten Bild
 - ▶ Hands-On Erfahrung sollte in allen Klassen von Applikationen gewonnen werden
- ▶ Kostenmodell als mittelfristiges Ziel
 - ▶ basierend auf Hands-On Erfahrung

Kosten – Beispiel Signaturen

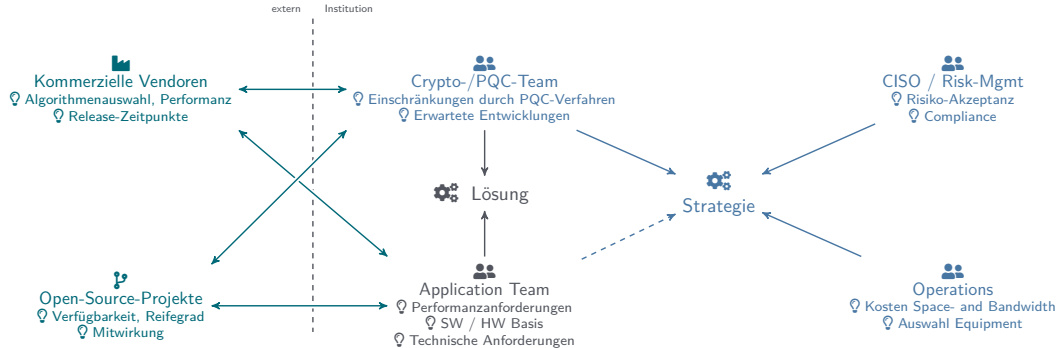
	LMS  Stateful-hash	ML-DSA  Gitter	Einmalige Kosten	Betriebs- und Folgekosten
Key mgmt complexity			HSM-Beschaffung und Integration	HSM-Administration
Crypto-agility requirement			Entwicklungskosten	Erneuter Verfahrenswechsel, neues Rollout
Signature size				Speicherplatz, Bandbreite
Speicherbedarf (große Nachrichten)			Abhängig von Use-case, Hardware-Support, etc.	



Kosten und Strategie

- ▶ Weitere Entscheidungsmöglichkeiten
 - ▶ Verzögerte Umsetzung wegen Warten auf zukünftige Verfahren oder Technologien
 - ▶ Einsatz von nicht standardisierten Verfahren
 - ▶ Wechsel von Sicherheitsprotokollen
 - ▶ Re-Engineering von proprietären Applikationsprotokollen
- ▶ Strategische Gesichtspunkte
 - ▶ Migration ist getrieben von Compliance-Anforderungen oder durch ermitteltes Risiko?
 - ▶ Knowhow-Management

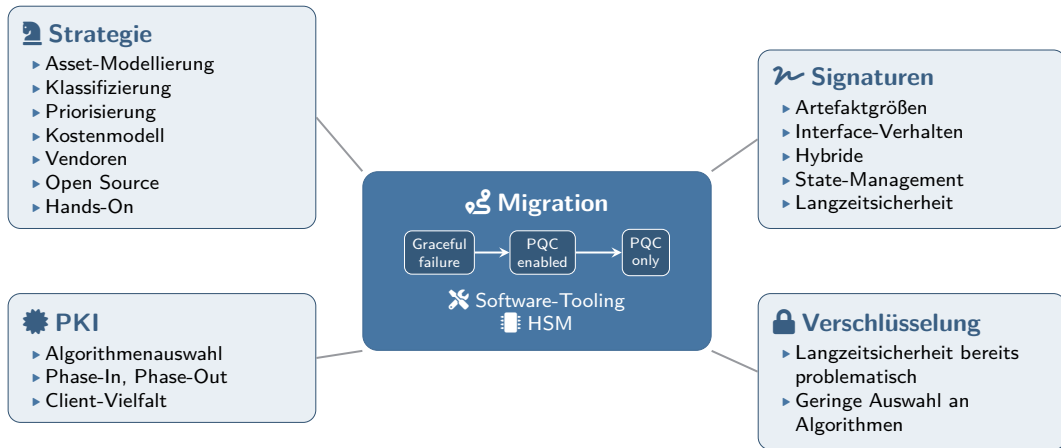
Migration: Aufgabenverteilung und Knowhow-Management



Zieldefinition

- ▶ PQC-Migration einer Anwendung
 - ▶ Vollständige Umstellung aller direkt und indirekt beteiligten Komponenten
 - ▶ Beinhaltet:
 - ▶ Nach außen hin sichtbare Kryptografie 🌐
 - ▶ Administrationswerkzeuge 🛠️
 - ▶ Unternehmenskommunikation ✉️
 - ▶ Authentisierungsmittel (Smartcard)
 - ▶ Denkansatz: “traditionelle Kryptografie bietet Schutz = 0”

Zusammenfassung



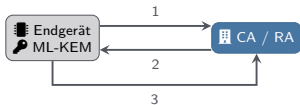
Vielen Dank für Ihre Aufmerksamkeit

Dr. Falko Strenzke
falko.strenzke@mtg.de
+49 6151 8000-24

MTG AG
www.mtg.de

Backup: KEM-Zertifikate – Proof of Possession und Zertifikatspaare

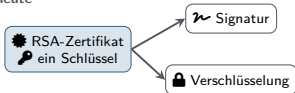
Proof of Possession (PoP)



- 1: CSR mit ML-KEM-Public-Key
- 2: Challenge oder verschlüsseltes Zertifikat, Geheimnis mit dem Public Key gekapselt
- 3: decapsulation, Hash bestätigen (CMP, rfc4210bis) Alternative RFC 9883: signierte Besitz-Erklärung mit bereits vorhandenem Signaturzertifikat

Zwei Zertifikate statt einem

heute



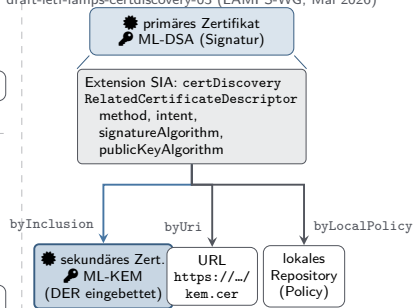
PQC: kein Verfahren für beides



→ zwei Schlüsselpaare, zwei CSRs, zwei PoP-Verfahren, zwei Zertifikate pro Identität (auch bei Composite)

Certificate Discovery

draft-ietf-lamps-certdiscovery-03 (LAMPs-WG, Mai 2026)



Ziel: ein Zertifikat ausliefern, das zweite finden oder gleich mitliefern

Backup: KEM-Zertifikate – Details

- ▶ **Proof of Possession für KEM-Schlüssel** (CMP, [rfc4210bis](#))
 - ▶ Indirekt: CA stellt das Zertifikat verschlüsselt aus (Schlüssel per KEM gekapselt), Endgerät dekapselt und bestätigt den Hash im `certConf`
 - ▶ Direkt: Challenge-Response mit gekapseltem Zufallswert
 - ▶ Schlüsselarchivierung: privater Schlüssel wird der CA/RA übergeben
- ▶ **RFC 9883** (Okt. 2025): PKCS#10-Attribut “Statement of Possession” – signierte Erklärung mit bereits ausgestellttem Signaturzertifikat; kein technischer Beweis, Subject muss übereinstimmen
- ▶ **Zwei Zertifikate**: ML-DSA für Signatur, ML-KEM für Verschlüsselung; RSA-“Dual-Use” entfällt (auch bei Composite)
- ▶ **draft-ietf-lamps-certdiscovery-03** (LAMPS-WG, Mai 2026)
 - ▶ SIA-Extension `certDiscovery` mit `RelatedCertificateDescriptor`: `method`, `intent`, `signatureAlgorithm`, `publicKeyAlgorithm`
 - ▶ Methoden: `byInclusion` (DER des zweiten Zertifikats direkt eingebettet), `byUri`, `byLocalPolicy`
 - ▶ Use-Cases: Algorithmen-Agilität, Backup-Zertifikate, “dual usage” (Signatur + Verschlüsselung); keine Identitätsbindung zwischen den Zertifikaten