Mixed Certificate Chains for Post-Quantum Authentication*

Sebastian Paul (Bosch Research)

Post-Quantum Cryptography Conference

March 3, 2023 – Ottawa, Canada

Full paper: S. Paul, Y. Kuzovkova, N. Lahr, and R. Niederhagen. 2022. *Mixed Certificate Chains for the Transition to Post-Quantum Authentication in TLS 1.3*. ASIA CCS '22. DOI: 10.1145/3488932.3497755.

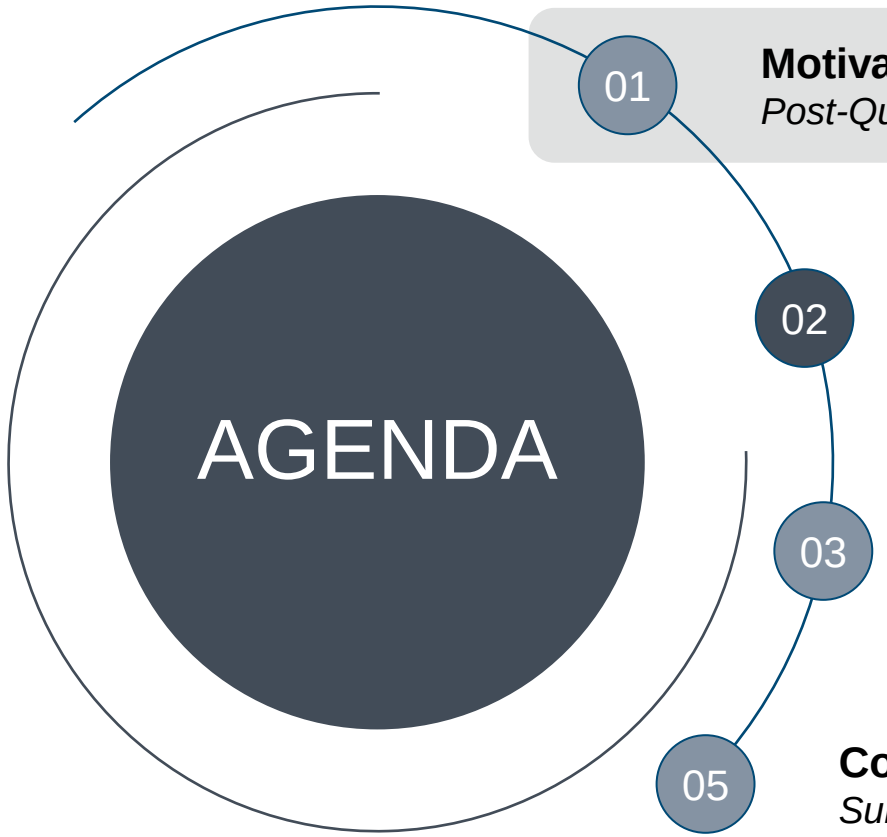
BOSCH

Mixed Certificate Chains for Post-Quantum Authentication
About Me

Sebastian Paul

- Security Research Engineer at Bosch with focus on Industrial IoT and PQC
- Bosch project lead for BMBF-funded project **Full Lifecycle Post-Quantum PKI - FLOQI**
- PhD from Technical University Darmstadt in Applied Post-Quantum Cryptography
  Thesis: "*On the Transition to Post-Quantum Cryptography in the Industrial Internet of Things*"
- MSc in Electrical Engineering from Karlsruhe Institute of Technology (KIT)

Mixed Certificate Chains for Post-Quantum Authentication



01 **Motivation**
Post-Quantum Authentication

02 **Mixed Certificate Chains**
Two-step Migration Strategy and Post-Quantum TLS 1.3

03 **Measurements & Evaluation**
Experimental Setup and Results

05 **Conclusion**
Summary and Future Work

Mixed Certificate Chains for Post-Quantum Authentication

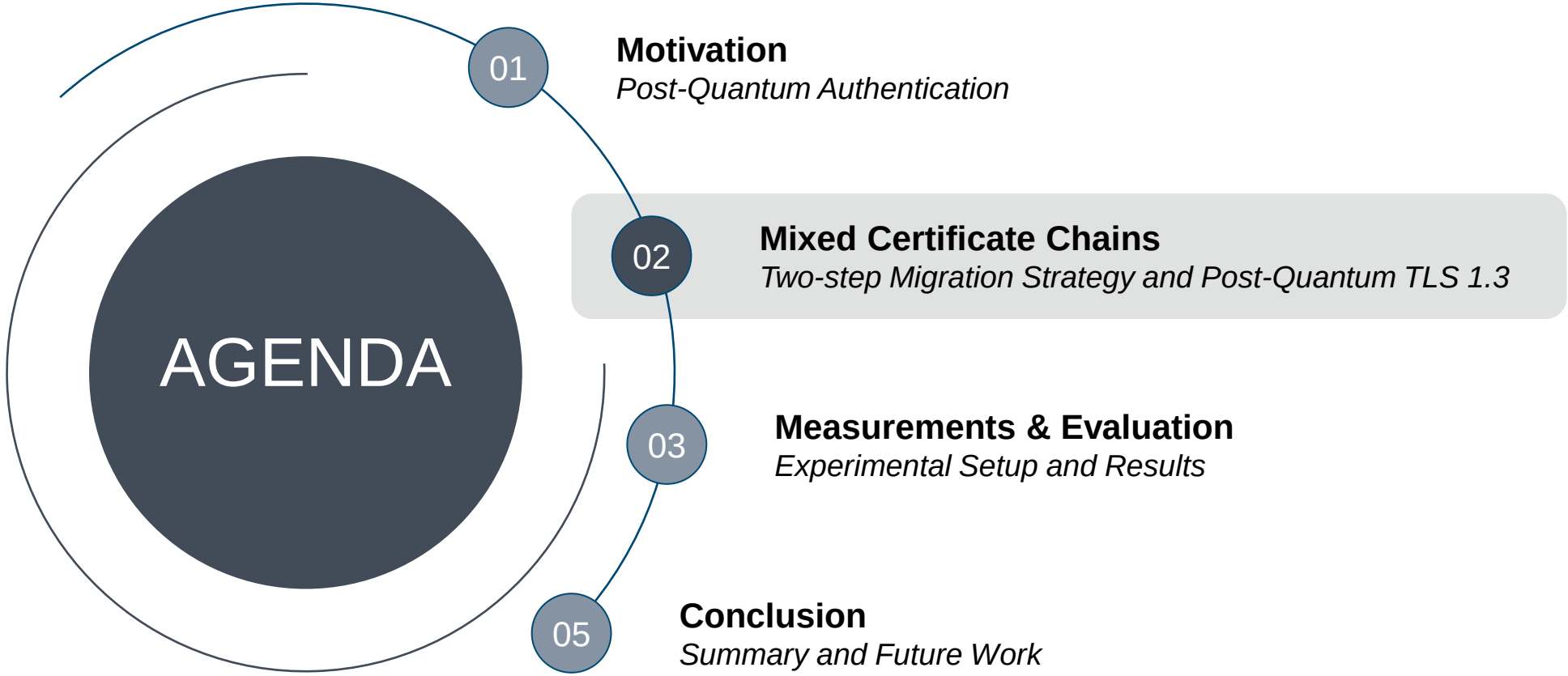
Why Worry About Post-Quantum Authentication Now?

- Existing migration strategies focus on confidentiality:
  - Confidentiality can be broken retroactively → *“Store now, decrypt later” attacks*
  - Hybrid key exchange → Combine conventional key agreement with a post-quantum key encapsulation mechanism

- Migration to **post-quantum authentication** needs to be completed before large-scale quantum computers exist:
  - Authentication typically based on certificates and public-key infrastructures
  - Complex and time-consuming migration process

→ Our goal: **Propose and investigate a migration strategy towards post-quantum authentication**

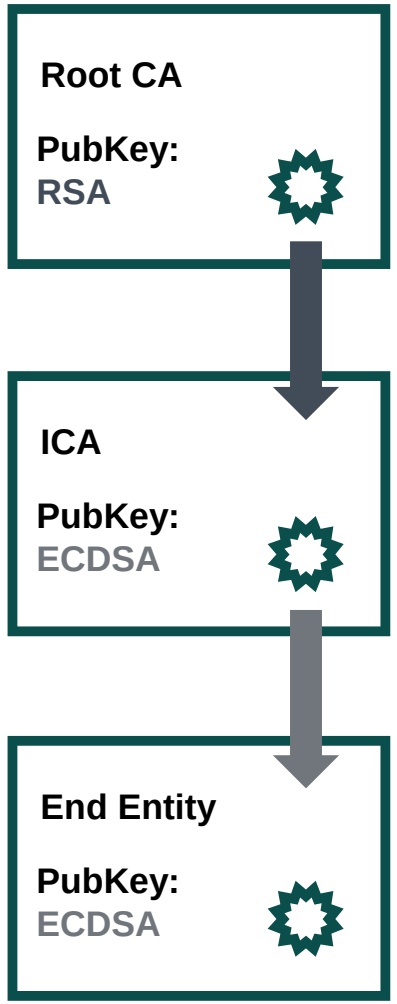
Mixed Certificate Chains for Post-Quantum Authentication



BOSCH

Mixed Certificate Chains for Post-Quantum Authentication
Two-Step Migration Strategy

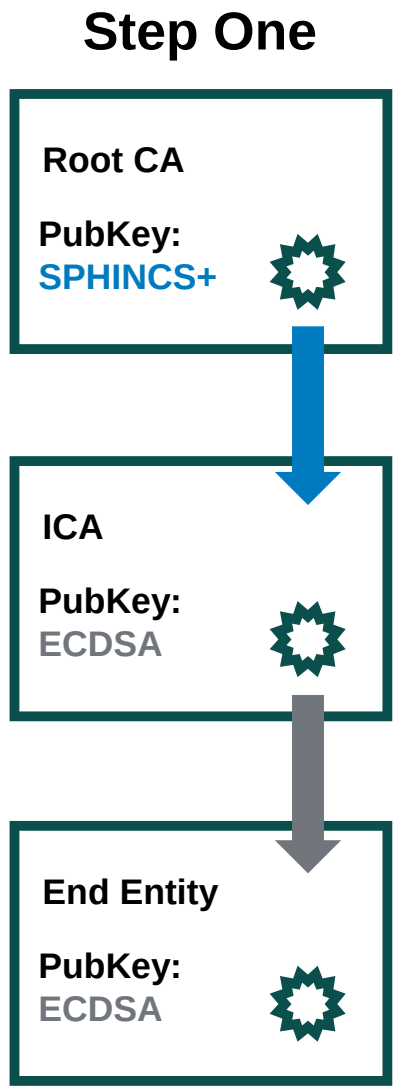
- Combine different signature algorithms within the same certificate chain
  → **Mixed certificate chains**



Mixed Certificate Chains for Post-Quantum Authentication

Two-Step Migration Strategy

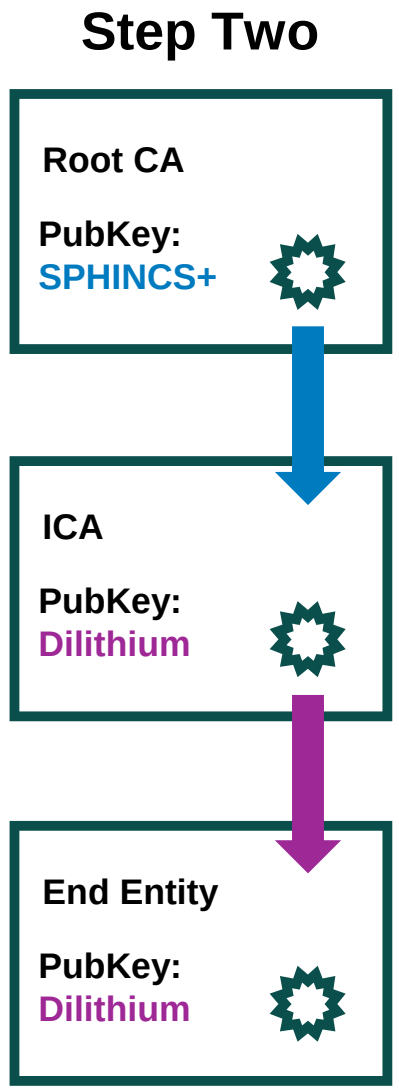
- Combine different signature algorithms within the same certificate chain
  → **Mixed certificate chains**
- Combine well-studied and trusted *hash-based signature schemes (SPHINCS+ or XMSS)* at **Root CA**, with:
  (1) Conventional ECC (ECDSA) at **ICA** and **End-Entity**



Mixed Certificate Chains for Post-Quantum Authentication

Two-Step Migration Strategy

- Combine different signature algorithms within the same certificate chain
  → **Mixed certificate chains**
- Combine well-studied and trusted *hash-based signature schemes (SPHINCS+ or XMSS)* at **Root CA**, with:
  (1) Conventional ECC (ECDSA) at **ICA** and **End-Entity**
  (2) Fast but newer *lattice-based schemes (Dilithium or Falcon)* at **ICA** and **End-Entity**
- **Goals:**
  (1) Seamless protection against quantum adversaries
  (2) Small certificates at End Entity Level
  (3) Feasible connection establishment time with little overhead because of additional signature schemes



Mixed Certificate Chains for Post-Quantum Authentication
Evaluated Scheme Combinations

Control: $DSA_{Root\&ICA\&EE}$ - KEX

Mixed Certificate Chain: XMS + $DSA_{ICA\&EE}$

Mixed Certificate Chain: SPf + $DSA_{ICA\&EE}$

SPf: speed-optimized SPHINCS$^+$

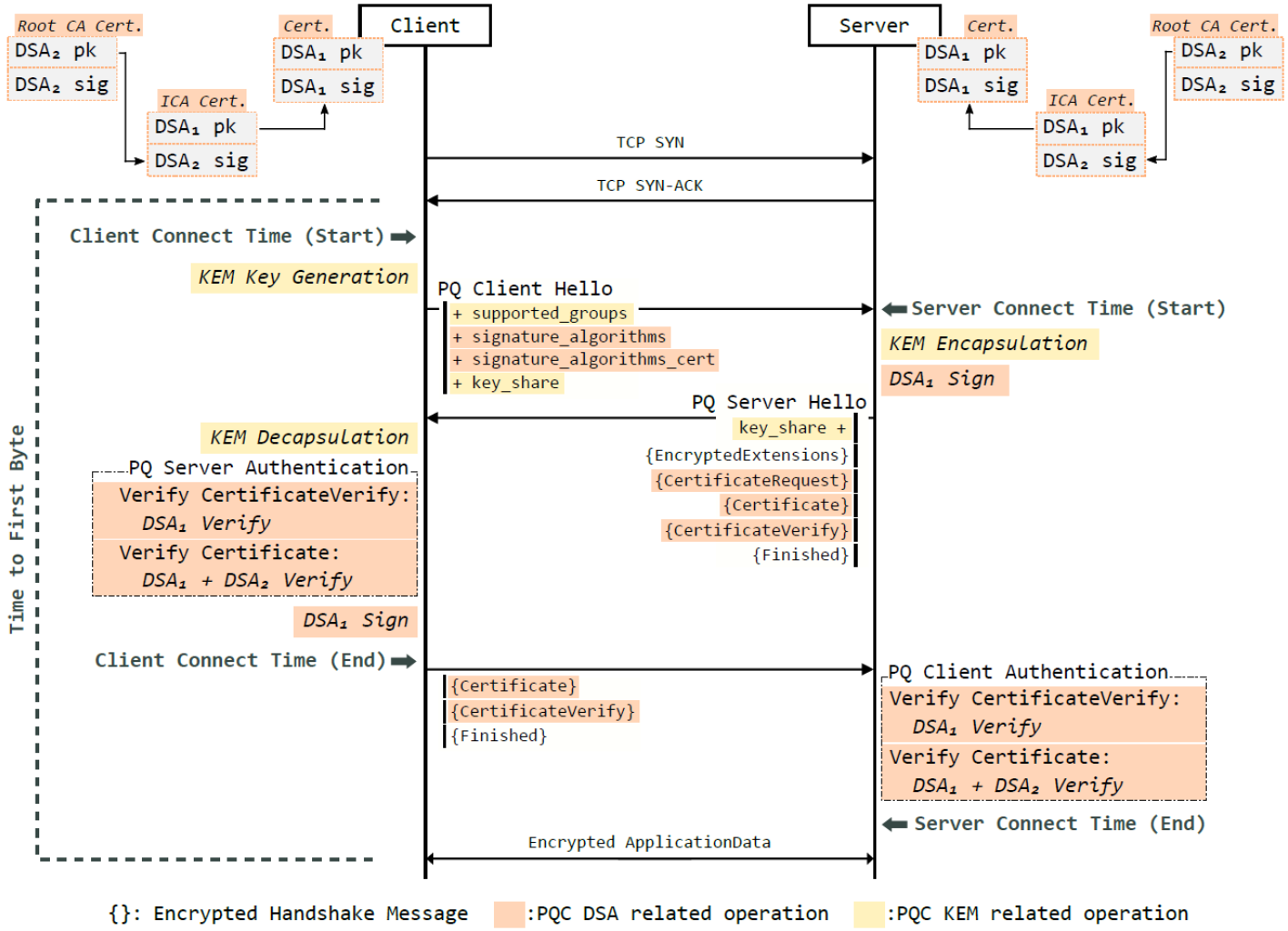
Mixed Certificate Chain: SPs + $DSA_{ICA\&EE}$

SPs: size-optimized SPHINCS$^+$

	Signature Scheme				
	Root CA	*Interm. CA*	*End Entity*	*KEX*	*Notation*[4]
Control	ECDSA	ECDSA	ECDSA	ECDHE	*EDS-EDH*
	Dilithium	Dilithium	Dilithium	Kyber	*Dil-Kyb*
	Falcon	Falcon	Falcon	Kyber	*Fal-Kyb*
	XMSS	XMSS	XMSS	Kyber	*XMS-Kyb*
	SPHINCS$^+$-f	SPHINCS$^+$-f	SPHINCS$^+$-f	Kyber	*SPf-Kyb*
	SPHINCS$^+$-s	SPHINCS$^+$-s	SPHINCS$^+$-s	Kyber	*SPs-Kyb*
Mixed Certificate Chain	XMSS	ECDSA	ECDSA	Kyber	*XMS+EDS-Kyb*
	XMSS	Dilithium	Dilithium	Kyber	*XMS+Dil-Kyb*
	XMSS	Falcon	Falcon	Kyber	*XMS+Fal-Kyb*
	SPHINCS$^+$-f	ECDSA	ECDSA	Kyber	*SPf+EDS-Kyb*
	SPHINCS$^+$-f	Dilithium	Dilithium	Kyber	*SPf+Dil-Kyb*
	SPHINCS$^+$-f	Falcon	Falcon	Kyber	*SPf+Fal-Kyb*
	SPHINCS$^+$-s	ECDSA	ECDSA	Kyber	*SPs+EDS-Kyb*
	SPHINCS$^+$-s	Dilithium	Dilithium	Kyber	*SPs+Dil-Kyb*
	SPHINCS$^+$-s	Falcon	Falcon	Kyber	*SPs+Fal-Kyb*

Mixed Certificate Chains for Post-Quantum Authentication

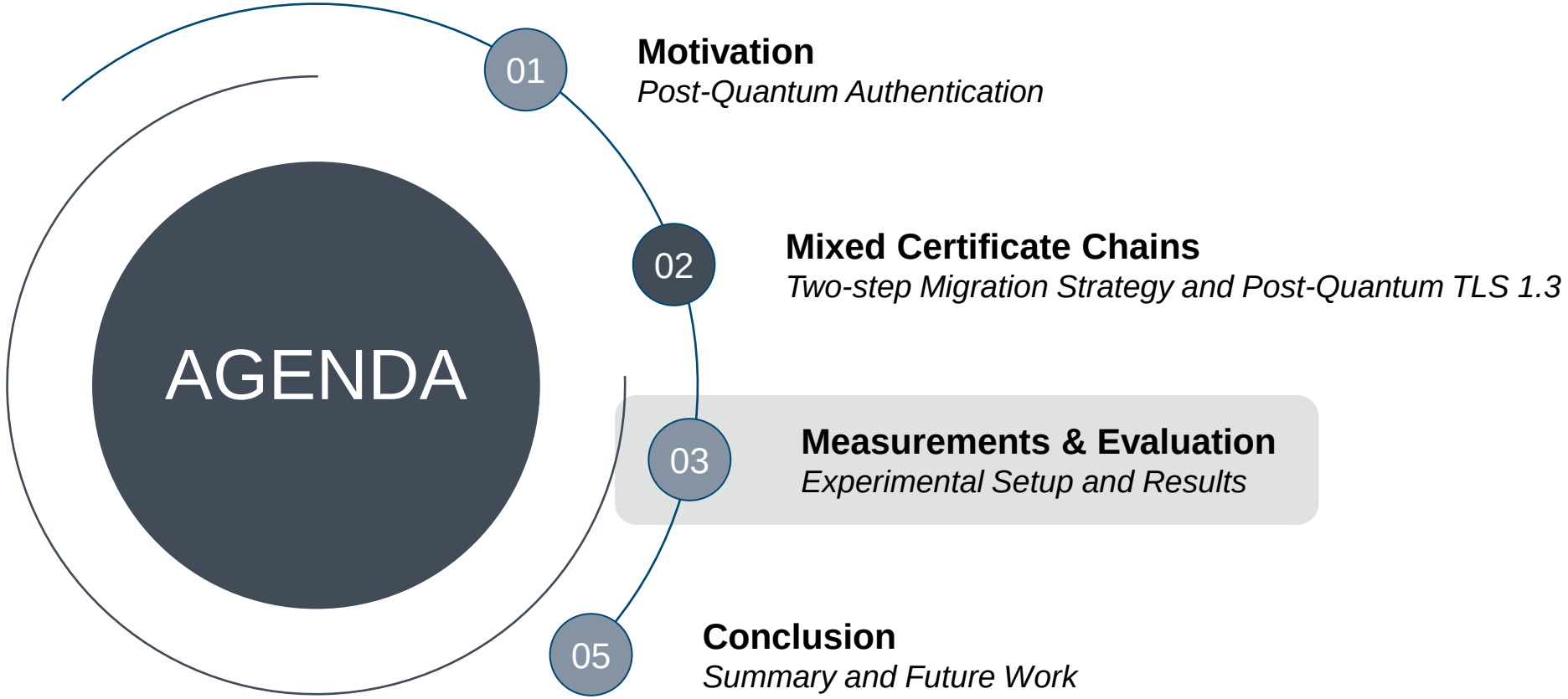
Post-Quantum TLS 1.3



Setup:

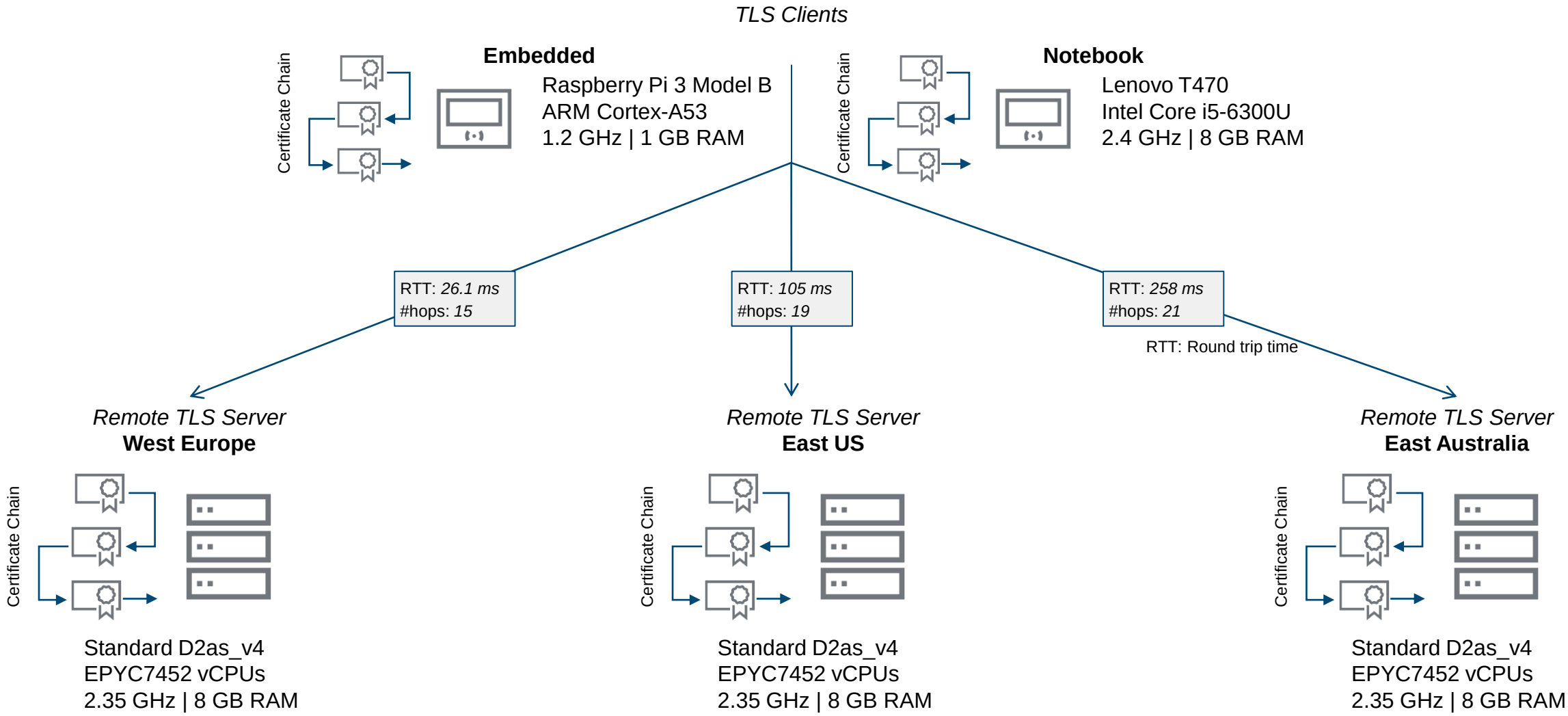
- Integration of PQC reference implementations into wolfSSL publicly available: https://github.com/boschresearch/pq-wolfSSL
- **Mutually authenticated TLS 1.3 handshake** using the full 1-RTT mode without pre-shared key-resumption.
- Selected cipher suite is TLS_AES_256_GCM_SHA384
- Certificate chain length of 3: Root – ICA – End Entity
- Select Kyber as efficient post-quantum KEM
- **KEM-operations:**
  - Client: 1x Key Generation + 1x Decapsulation
  - Server: 1x Encapsulation
- **DSA-operations:**
  - Client: 3x Verify + 1x Sign
  - Server: 3x Verify + 1x Sign
- Measurements:
  - Time to First Byte (TTFB)
  - Client Connect Time
  - Server Connect Time

Mixed Certificate Chains for Post-Quantum Authentication



BOSCH

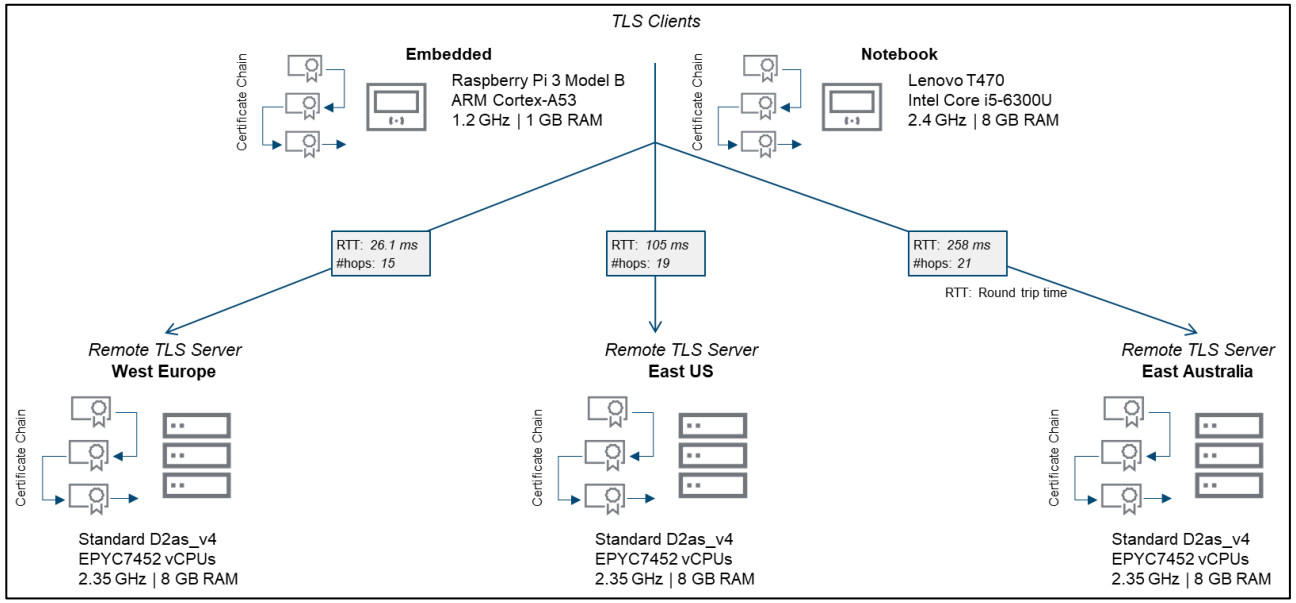
Mixed Certificate Chains for Post-Quantum Authentication
Experimental Setup



Mixed Certificate Chains for Post-Quantum Authentication
Performance Study

Measurements*

1) **Performance benchmark** of cryptographic primitives on all device types (Raspberry Pi3, Notebook, Azure VM)
2) **TLS connection establishment times**
3) **Certificate and communication size**
4) **Peak memory** usage (stack & heap)



* **Check out full paper for details:** 10.1145/3488932.3497755

Mixed Certificate Chains for Post-Quantum Authentication
Benchmark of Evaluated Signature Schemes

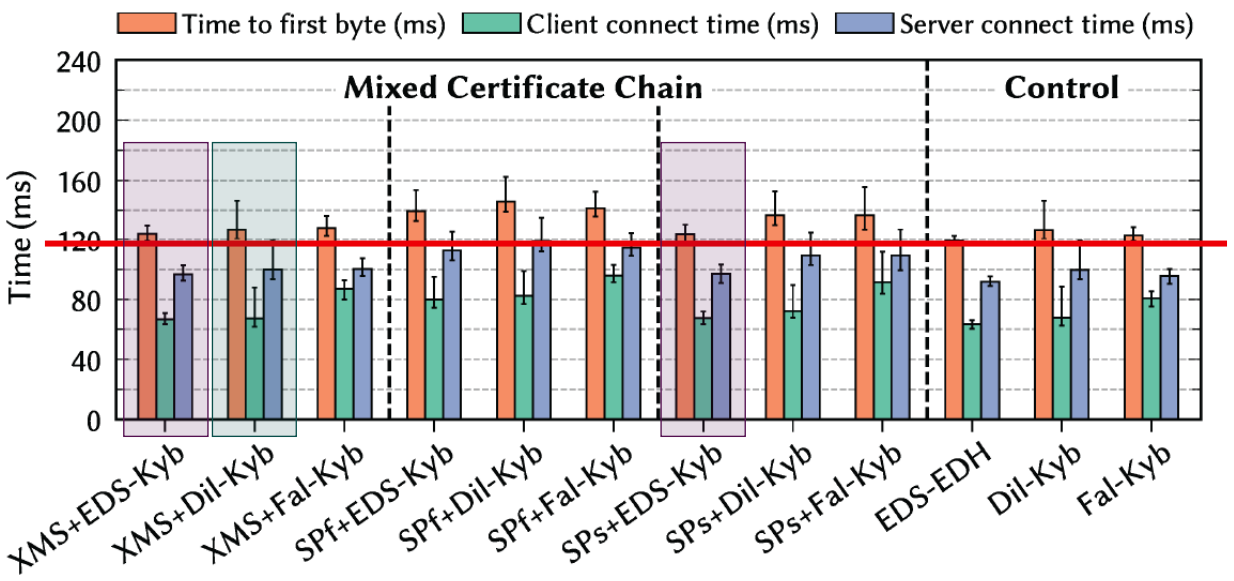
	Algorithm (Parameter)	NIST Level	Sizes (byte)		Performance (ms) Embed.	Notebook	Server
			Signature Schemes				
Conventional Cryptography	ECDSA (SECP256R1)	×	**sk:** 32 **pk:** 65 **sig:** 73	**gen:** **sign:** **vfy:**	1.52 1.94 4.85	0.0920 0.116 0.285	0.0910 0.119 0.301
Lattice-Based PQC	Dilithium [7] (Dilithium-2)	2	**sk:** 2544 **pk:** 1312 **sig:** 2420	**gen:** **sign:** **vfy:**	2.04 11.9 2.21	0.107 0.414 0.121	0.0880 0.389 0.0990
	Falcon [26] (Falcon-512)	1	**sk:** 1281 **pk:** 897 **sig:** 666	**gen:** **sign:** **vfy:**	158 35.7 0.435	20.1 5.90 0.0420	16.9 4.91 0.0310
Hash-Based PQC	SPHINCS$^+$ [5] (SHA-256-128s -simple)	1	**sk:** 64 **pk:** 32 **sig:** 7856	**gen:** **sign:** **vfy:**	473 3540 3.53	114 866 0.876	93.6 710 0.678
	SPHINCS$^+$ [5] (SHA-256-128f -simple)	1	**sk:** 64 **pk:** 32 **sig:** 17,088	**gen:** **sign:** **vfy:**	7.33 183 10.2	1.75 43.3 2.46	1.47 36.4 2.05
	XMSS [30] (XMSS-SHA2 -10-256)	–[9]	**sk:** 36 **pk:** 64 **sig:** 2500	**gen:** **sign:** **vfy:**	11,300 50.1 6.49	2190 9.70 1.20	1870 8.26 1.03

Results

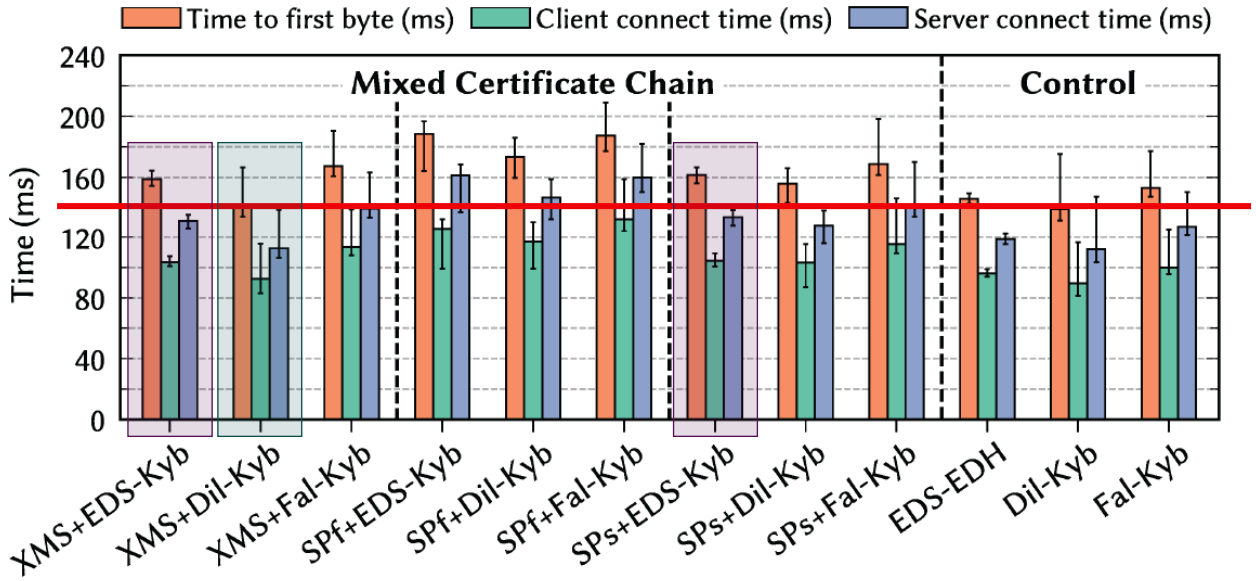
- *Key sizes:*
  - Lattice-based schemes: Very balanced profile, nevertheless larger keys/sigs than ECDSA
  - Hash-based schemes: Very large signatures, but small public and private keys
- *Performance:*
  - Signing operation very expensive in hash-based signature schemes → up to 3.5 s
  - Verification operation feasible in all PQC schemes → Dilithium and Falcon even faster than ECDSA

Notation:
secret key (sk), public key (pk), signature (sig)
key generation (gen), sign (sign), verify (vfy)

Mixed Certificate Chains for Post-Quantum Authentication
Connection Establishment Times: West Europe



Notebook → West Europe



Embedded → West Europe

Results

- Feasible increase in median TTFB across all nine evaluated mixed certificate chains compared to ECC-based control handshake (EDS-EDH).
- For connections to the server in West Europe, this increase is +12.4% (Notebook) and +14.4% (Embedded)
- As expected, the increase becomes less significant when connecting to server located at greater distances
- *Intermediate migration step:* The combinations **SPs+EDS-Kyb** and **XMS+EDS-Kyb** seem promising transitional candidates
- *Final migration step:* the combination **XMS+Dil-Kyb** shows the fastest TTFB

Mixed Certificate Chains for Post-Quantum Authentication
Certificate Sizes

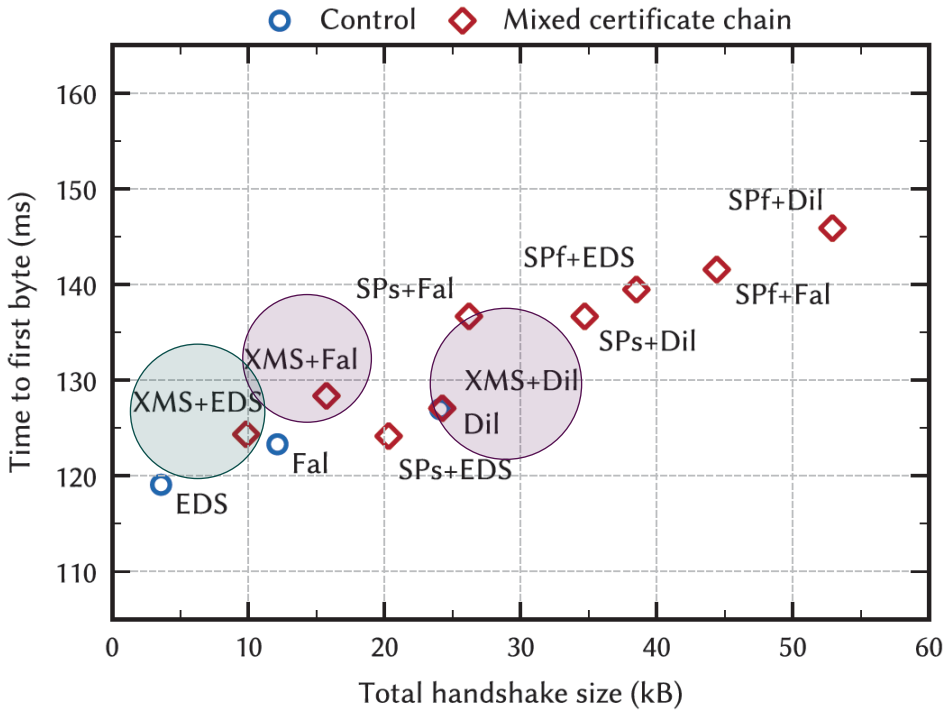
Mixed Certificate Chain	Group	Certificate Size (kB)			Chain Size (excl. root; kB)	Δ (%)
		Root CA	ICA	End-Entity		
EDS-EDH	control	0.775	0.803	0.764	1.57	—
Dil-Kyb		5.59	5.62	5.58	11.2	+615
Fal-Kyb		2.71	2.74	2.69	5.43	+246
XMS-Kyb		4.04	4.07	4.03	8.10	+417
SPf-Kyb		23.3	23.3	23.3	46.6	+2870
SPs-Kyb		11.1	11.1	11.1	22.2	+1320
XMS+EDS-Kyb	mixed	4.04	4.09	0.760	4.85	+209
XMS+Dil-Kyb		4.04	5.72	5.58	11.3	+621
XMS+Fal-Kyb		4.04	5.17	2.68	7.87	+402
SPf+EDS-Kyb	mixed	23.3	23.4	0.764	24.1	+1440
SPf+Dil-Kyb		23.3	25.0	5.58	30.6	+1850
SPf+Fal-Kyb		23.3	24.5	2.70	27.2	+1630
SPs+EDS-Kyb	mixed	11.1	11.2	0.760	11.9	+662
SPs+Dil-Kyb		11.1	12.8	5.58	18.4	+1070
SPs+Fal-Kyb		11.1	12.3	2.70	15.0	+855

Certificate sizes of evaluated scheme combinations (reported in kB; rounded to three significant figures)

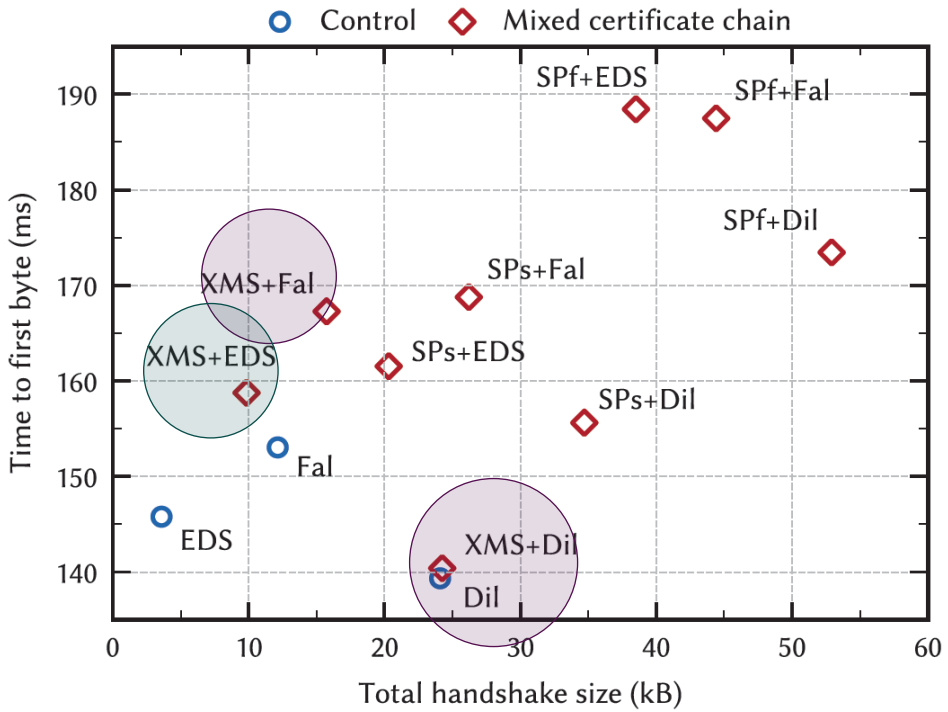
Results

- Size of certificates and cert. chains increase significantly
- *Intermediate migration step:* Combination of XMSS and ECDSA shows smallest increase (+ 3.01 kB)
- *Final migration step:* Combination of XMSS and Falcon leads to the smallest increase (+ 6.3 kB)
- Speed-optimized variant of SPHINCS$^+$ (SPf) leads to largest certificate chain sizes due to its large signatures (16,7 kB)

Mixed Certificate Chains for Post-Quantum Authentication
Communication Size



Notebook → West Europe*



Embedded → West Europe*

Results

- Increase in communication size across all evaluated scheme combinations as a result of larger certificate chains, PQ signatures, and PQ ciphertexts
- *Intermediate migration step:* **XMS+EDS-Kyb** leads to smallest total handshake size (10.1 kB)
- *Final migration step*: **XMS+Fal-Kyb** has lowest total handshake size (16.1 kB), but slower median TTFB compared to **XMS+Dil-Kyb** (24.8 kB).

Mixed Certificate Chains for Post-Quantum Authentication

Client Program – Peak Memory Usage

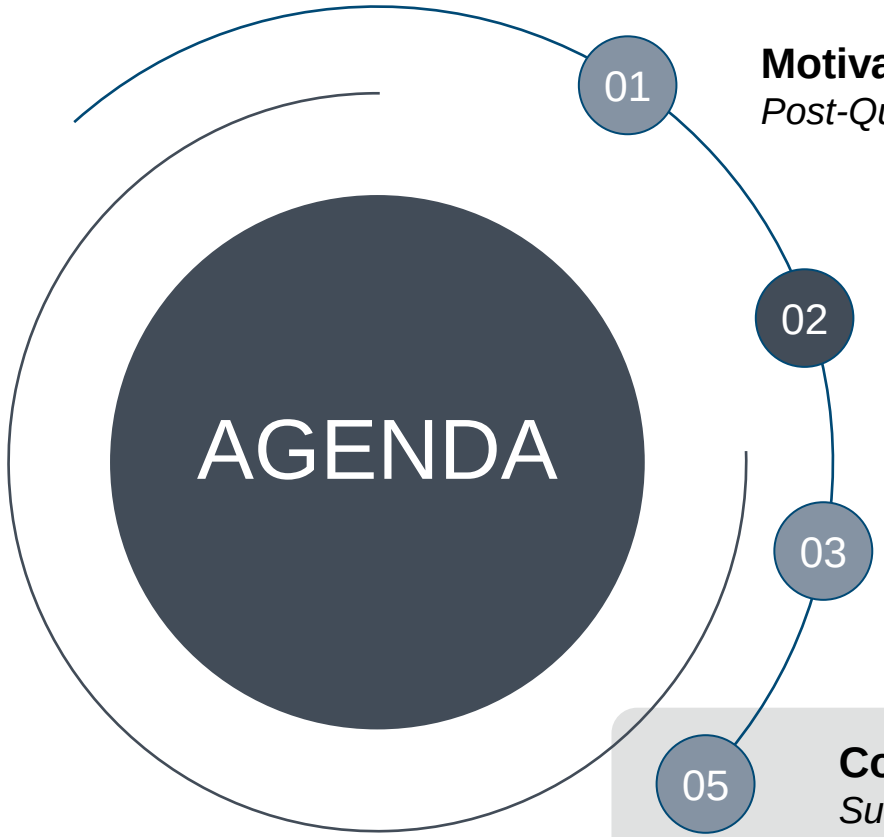
	Algorithm Combination	Client: Embedded			
		heap	*stack*	*total*	Δ (%)
Control	EDS-EDH	107	62.8	170	—
	Dil-Kyb	128	111	240	+41.2
	Fal-Kyb	119	102	221	+30.2
	XMS-Kyb	122	63.5	186	+9.49
	SPf-Kyb	167	61.3	228	+34.4
	SPs-Kyb	129	61.9	191	+12.4
Mixed Certificate Chain	XMS+EDS-Kyb	115	63.3	178	+5.19
	XMS+Dil-Kyb	129	111	240	+41.4
	XMS+Fal-Kyb	119	102	221	+30.2
	SPf+EDS-Kyb	131	68.3	199	+17.2
	SPf+Dil-Kyb	142	111	253	+49.0
	SPf+Fal-Kyb	135	102	236	+39.3
	SPs+EDS-Kyb	119	63.3	182	+7.43
	SPs+Dil-Kyb	133	111	244	+43.7
	SPs+Fal-Kyb	126	102	227	+34.0

Peak memory usage of client program on embedded platform

Results

- Heap usage is mostly affected by dynamic memory allocations related to buffers for sending messages
  → Increase of heap usage across all evaluated combinations due to larger certificates and cryptographic material.
- *Intermediate migration step:* Combination of XMSS and ECDSA shows smallest increase (+5.2 %)
- *Final migration step:* Combination of XMSS and Falcon leads to the smallest increase (+30.2 %)
- Dilithium leads to high increase in stack usage, which depends on implementation of underlying hard mathematical problems
  → Optimizations may be required for more resource-constrained embedded devices.

Mixed Certificate Chains for Post-Quantum Authentication



01 **Motivation**
Post-Quantum Authentication

02 **Mixed Certificate Chains**
Two-step Migration Strategy and Post-Quantum TLS 1.3

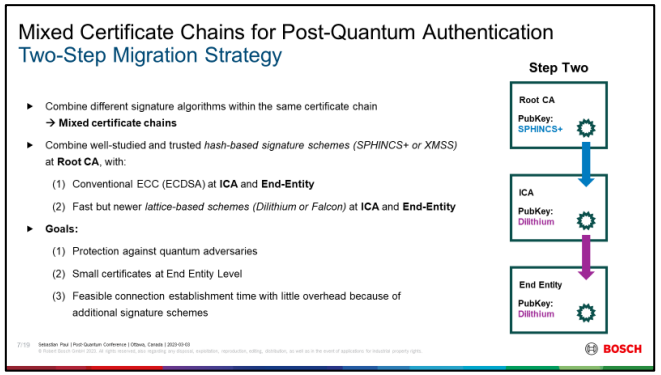
03 **Measurements & Evaluation**
Experimental Setup and Results

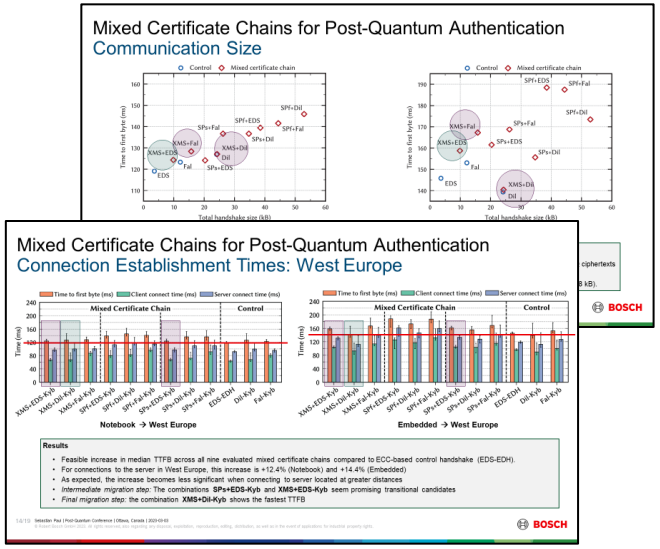
05 **Conclusion**
Summary and Future Work

BOSCH

Mixed Certificate Chains for Post-Quantum Authentication
Summary

- Proposed migration strategy based on **mixed certificate chains** is feasible
- *Intermediate migration step:* **XMSS+ECDSA-Kyber** shows fast connection establishment times, lowest overhead in communication and code size, as well as lowest memory usage:
  - Hash-based signatures at the root CA level offer conservative security
  - Alleviate drawbacks of hash-based signature schemes
- *Final migration step:* **XMSS+Dilithium-Kyber** is feasible for both client devices in terms of connection establishment times:
  - Impact on RAM significant → *high stack usage of Dilithium's implementation*





Thank you!

Questions?

sebastian.paul2@de.bosch.com

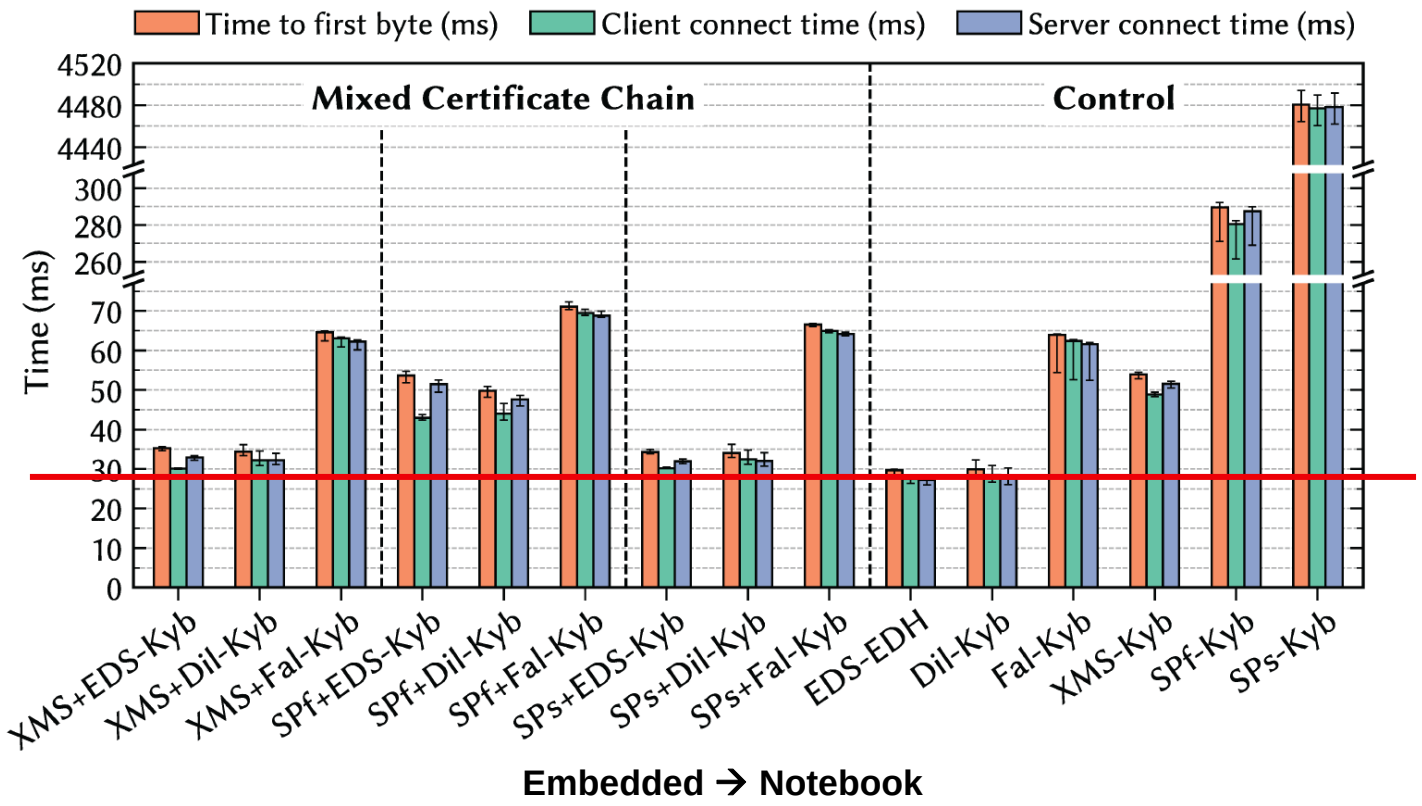
BOSCH

Mixed Certificate Chains for Post-Quantum Authentication
Benchmark of Evaluated KEMs

Overview of evaluated key establishment schemes including performance benchmark on target platforms.

Algorithm (Parameter)	*NIST Level*	*Sizes (byte)*		*Performance (ms)*		
				Embed.	*Notebook*	*Server*
Key Encapsulation Schemes						
ECDHE (SECP256R1)	×	**sk:** 32	**gen:**	1.52	0.0920	0.0910
		pk: 65	**agmt:**	4.40	0.255	0.271
Kyber [6] (Kyber512)	1	**sk:** 1632	**gen:**	0.572	0.0380	0.0330
		pk: 800	**enc:**	0.772	0.0440	0.0370
		ct: 768	**dec:**	0.772	0.0490	0.0430

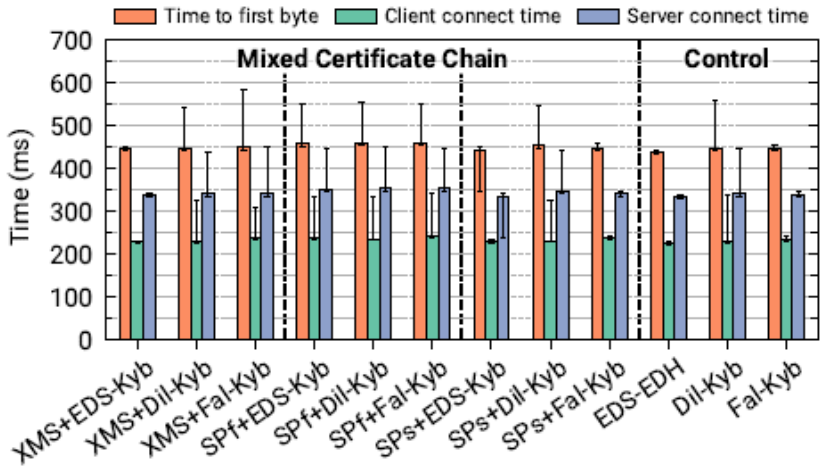
Mixed Certificate Chains for Post-Quantum Authentication
Connection Establishment Times: Local Server



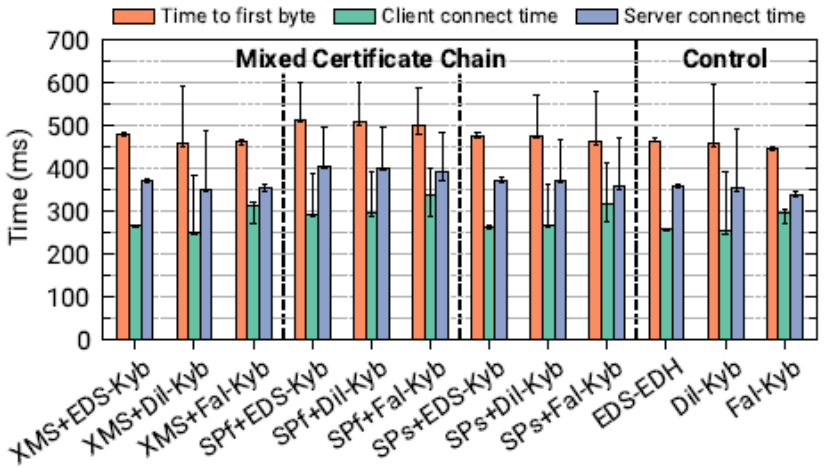
Results

- Two control combinations show best performance:
  - EDS-EDH: 29.7 ms
  - Dil-Kyb: 30.0 ms
  - SPHINCS$^+$ based control chain not feasible
- *Intermediate migration step:* Combinations of hash-based signature schemes with ECDSA are feasible
  - SPs+EDS-Kyb: +4.7 ms
  - XMS+EDS-Kyb: +5.6 ms
- *Final migration step:* Similar small increase in median time to first byte:
  - SPs+Dil-Kyb: +4.4 ms
  - XMS+Dil-Kyb: +4.7ms

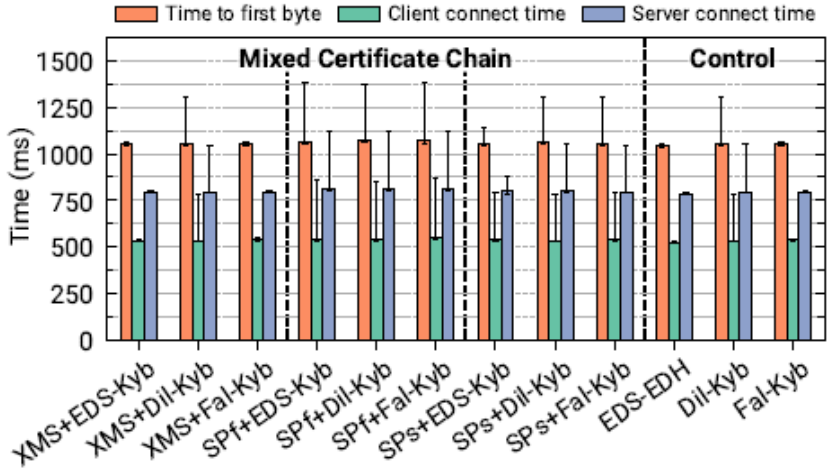
Mixed Certificate Chains for Post-Quantum Authentication
Connection Establishment Times: Remote Servers



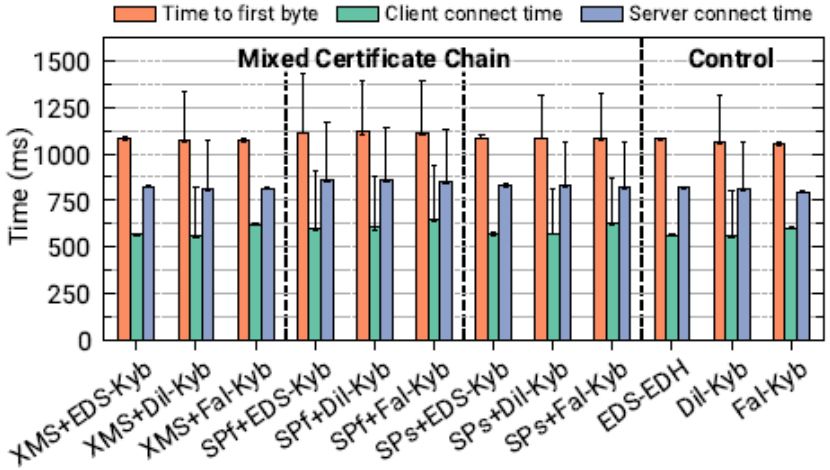
Notebook → Server (East US)



Embedded → Server (East US)



Notebook → Server (East Australia)



Embedded → Server (East Australia)

Mixed Certificate Chains for Post-Quantum Authentication
Library Code Size

Algorithm Combination	*Embedded*			*Notebook*		
	Code Size (kB)	*Overhead (kB)*	*Δ (%)*	*Code Size (kB)*	*Overhead (kB)*	*Δ (%)*
EDS-EDH	393	—	—	377	—	—
Dil-Kyb	633	240	+61.2	484	107	+28.5
Fal-Kyb	727	334	+85.1	569	192	+51.0
XMS(+EDS)-Kyb	602	209	+53.2	448	71.5	+19.0
XMS+Dil-Kyb	649	257	+65.3	503	126	+33.5
XMS+Fal-Kyb	743	350	+89.2	588	211	+56.0
SPf(+EDS)-Kyb	607	215	+54.6	457	79.7	+21.2
SPf+Dil-Kyb	655	262	+66.8	511	134	+35.7
SPf+Fal-Kyb	748	356	+90.6	596	219	+58.2
SPs(+EDS)-Kyb	607	214	+54.6	456	79.4	+21.1
SPs+Dil-Kyb	654	262	+66.7	511	134	+35.5
SPs+Fal-Kyb	748	355	+90.6	596	219	+58.1

Total code size of wolfSSL library for evaluated for evaluated scheme combinations

Results

- Integration of hash-based signature schemes and Kyber leads to smallest overhead in code size → *enables first migration step*
- As Kyber and Dilithium use wolfSSL's implementation of SHA3, combining hash-based signature schemes with Dilithium for a post-quantum secure TLS handshake leads to acceptable overhead
  → *enables final migration step*:
  - Embedded device: increases by another 12.1 %
  - Notebook: increase by another 14.5 %.
- Since most of required code size ends up in static flash memory, increase should be tolerable – even in resource constrained embedded systems.