Building FIPS-compliant Quantum-Safe TLS Key Exchange

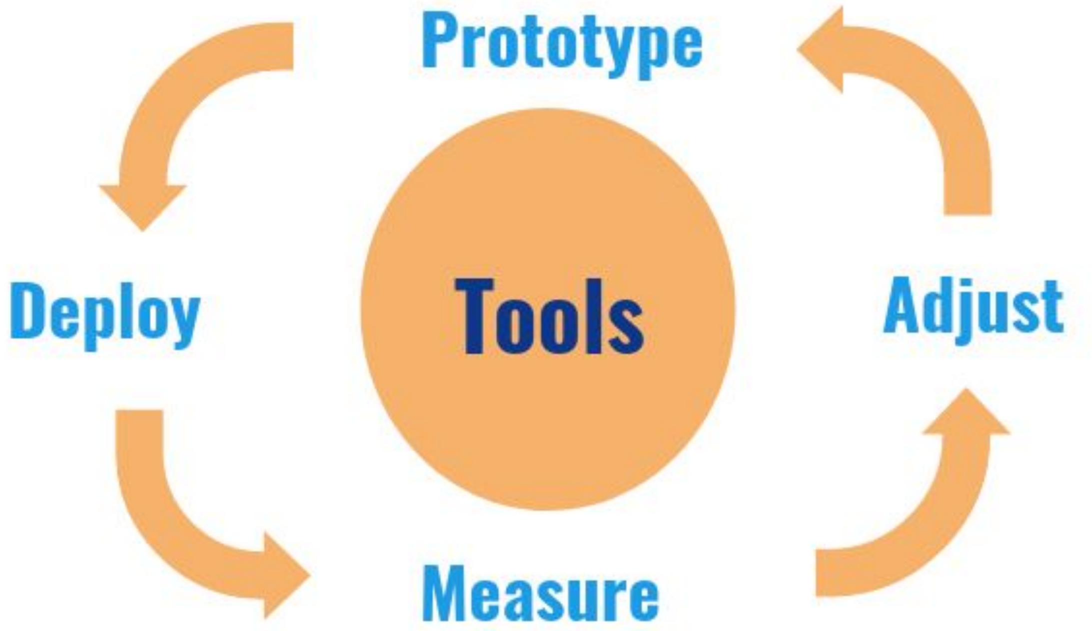
Think openly, build securely

May, 23rd 2022

Kris Kwiatkowski
Sr. Cryptography Engineer, PQShield LTD

Motivation

- **NIST PQC Standardization is at the final stage**
  - Secure implementations take time
  - Backward compatibility must be provided during migration to support not updated parties
- **Availability of progressive migrate**
  - Long adoption of cryptosystems
  - Operational aspect needs to be well studied by before deployment happens
- **Customer driven**
  - Mitigation of risks to vulnerable cryptosystems

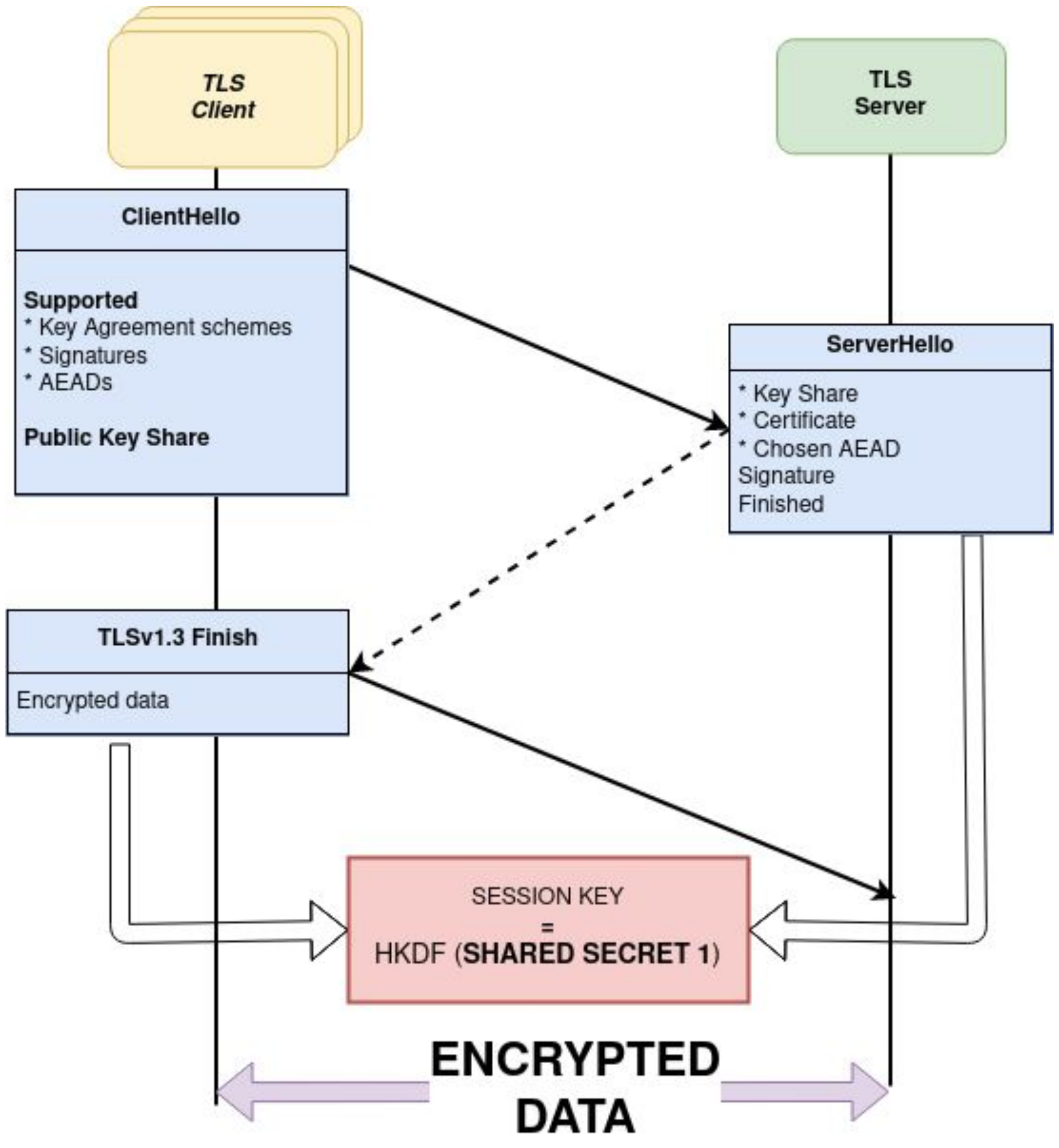


TLS

- **Among the most important driving forces for the migration to PQ cryptosystems**
  - Solution should be limited to the newest version of the protocol
  - **Goal**: Clear migration path
- **Key Agreement in the TLS protocol**
  - Possibility to attack retrospectively
  - Industry led experimental deployments, provide meaningful data
  - **Goal**: Backward compatibility for supporting not updated parties
- **Currently out-of-scope**
  - No support for any of PQ cryptosystems in TLS, as specified by IETF
  - Authentication seems more complicated. Future work

Protocol shape

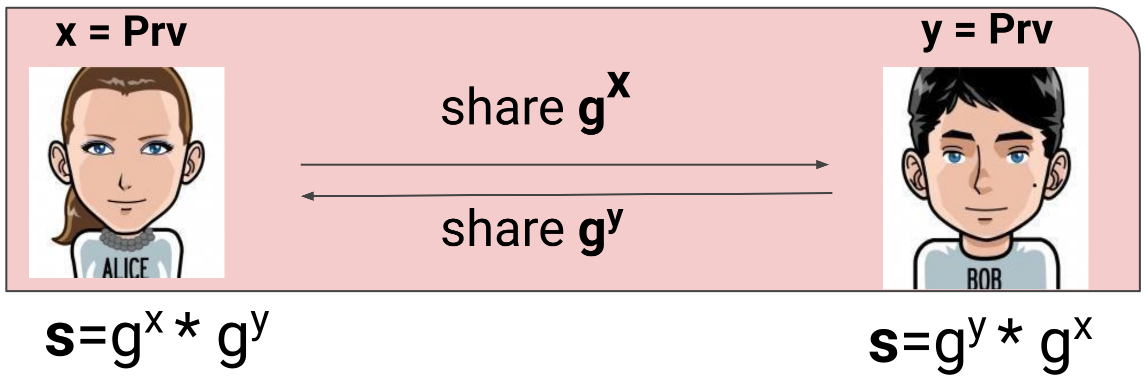
- One Round Trip protocol
- **The client** initiates with its key share and list of supported algorithms
- **The server** responds with its key share, certificate and signature
- Both sides use two-step, extract-then-expand KDF (HKDF) used for session key derivation
- Possible to fit **Post-Quantum KEM**



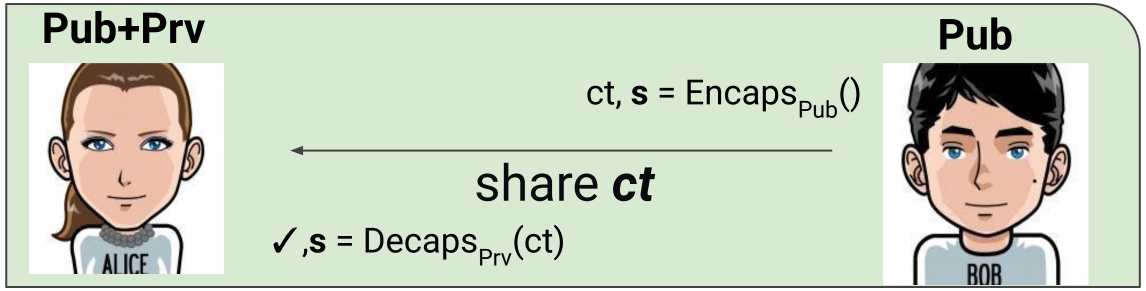
Diffie-Hellman and PQ KEM
Differences

- **KEM Interface**
  - Asymmetric: Both sides perform different operation
  - Doesn't fit into DH interfaces
- **Operations**
  - Randomized encapsulation
  - Deterministic decapsulation requires both public and private keys
- **IND-CCA2 security**
  - Shared secret **s** always indistinguishable from random (even if attacker has an ability to decapsulate arbitrary ciphertexts).
  - Security against active attacker

DH



KEM



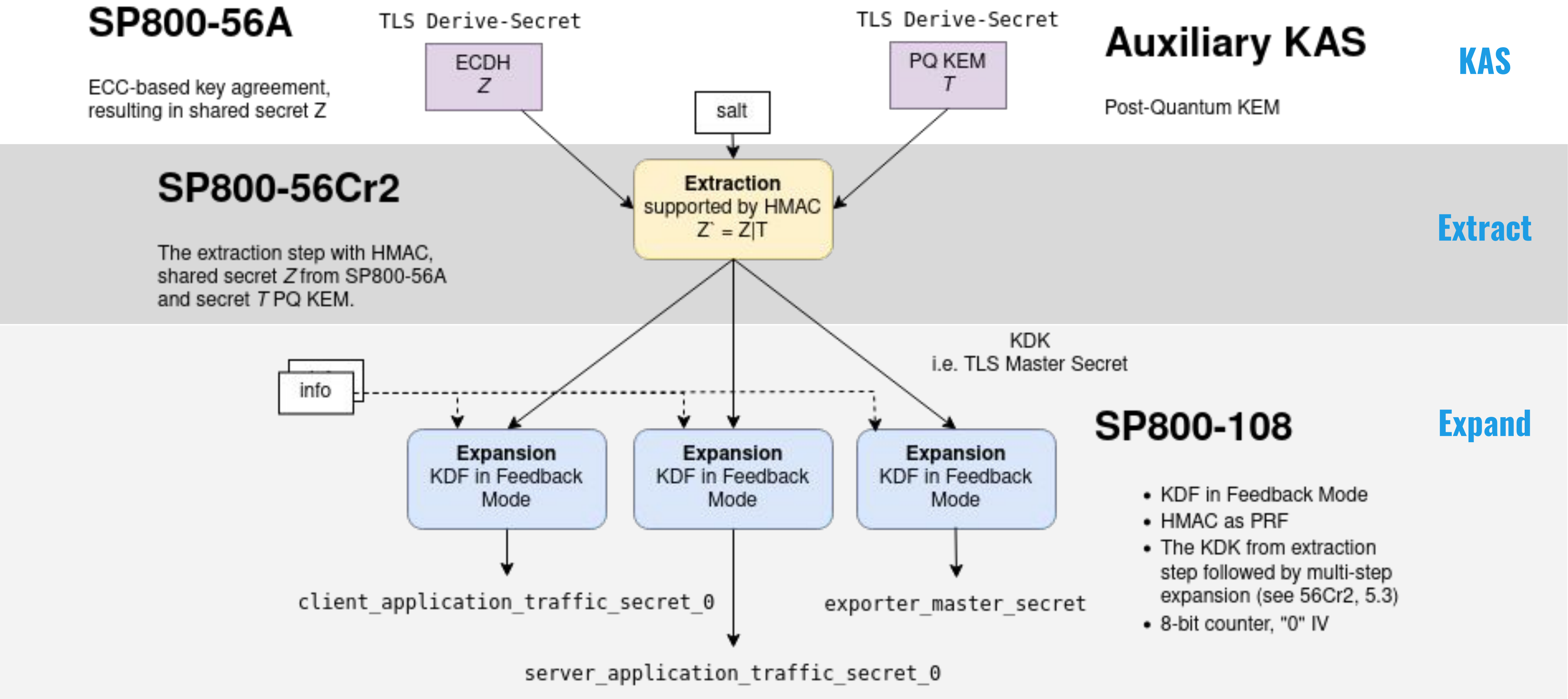
Approach to FIPS-certifiable Key Exchange

- **The "Hybrid" mode**
  - Concatenate output of two key agreements
  - Combine with some Extract-then-Expand KDF (HKDF)
- **NIST Special Publication 800-56Cr2**
  - Allows mixing outputs of FIPS-approved and auxiliary key agreement scheme (KAS)
- **No security can be claimed on auxiliary KAS**

SP800-56C rev2

In addition to the currently **approved** techniques for the generation of the shared secret Z as specified in SP 800-56A and SP 800-56B, this Recommendation permits the use of a "hybrid" shared secret of the form $Z' = Z \| T$, a concatenation consisting of a "standard" shared secret Z that was generated during the execution of a key-establishment scheme (as currently specified in [SP 800-56A] or [SP 800-56B]) followed by an auxiliary shared secret T that has been generated using some other method. The content, format, length, and method used to generate T must be known

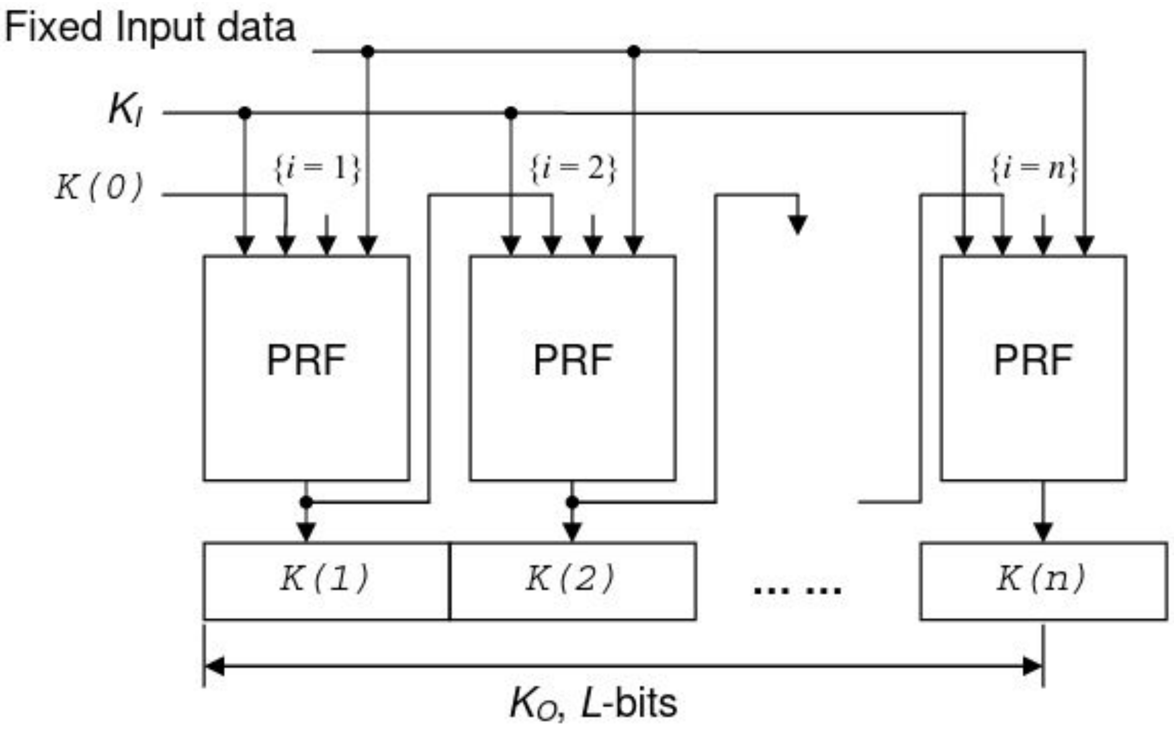
FIPS compliance rationale



TLS compliance rationale
SP800-108 Key expansion

TLS Key Derivation

- HKDF based
- Randomness extraction step: HMAC-based
- Multi-expansion step:
  - HMAC is used as PRF
  - KDF in feedback mode
  - 8-bit counter
  - zero-length IV, used as an initial value of K(0)



Example of integration into TLS

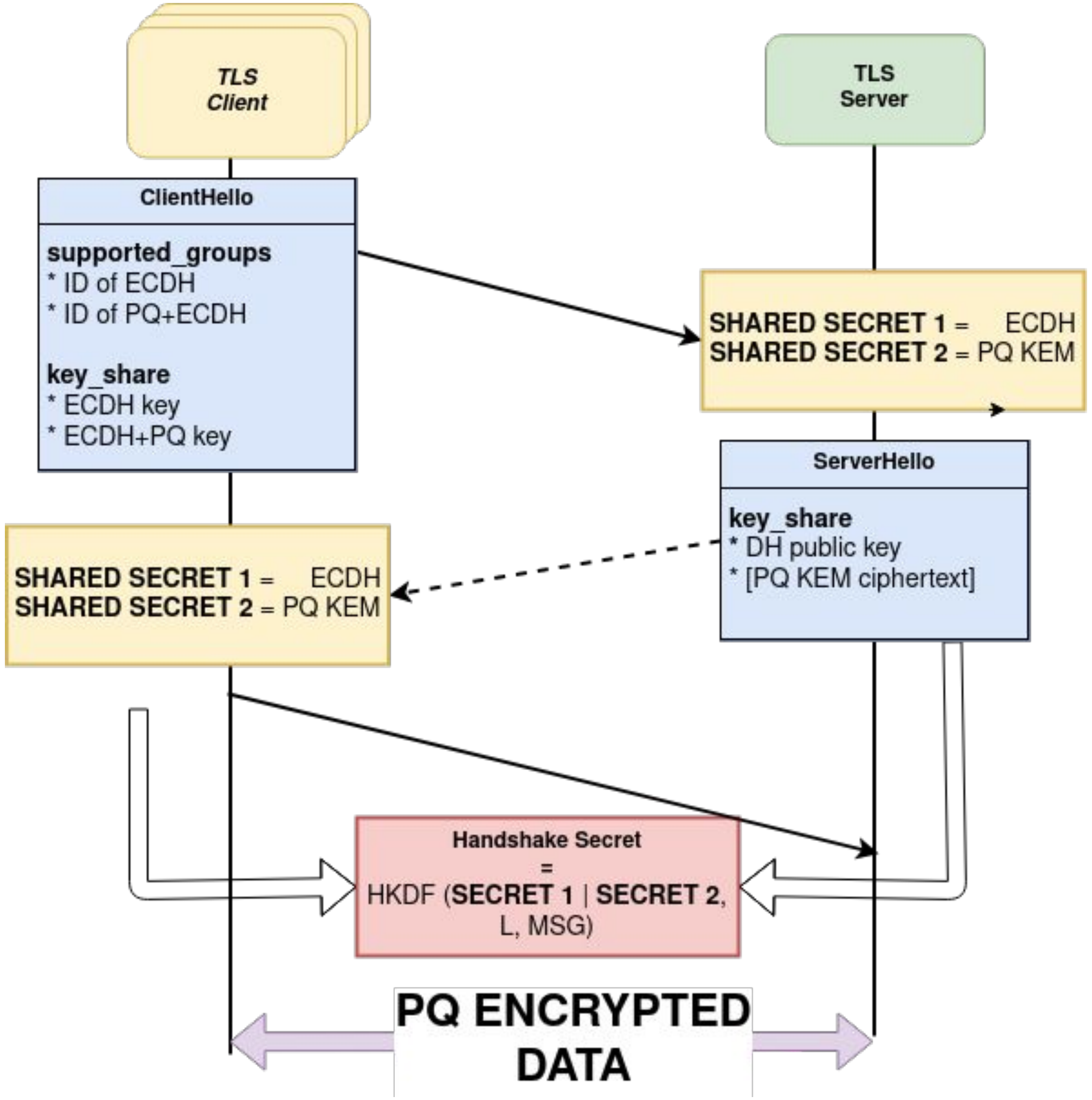
- One ID per each combination of PQ and classical scheme

- Concatenation of public keys and shared secrets (no structure)

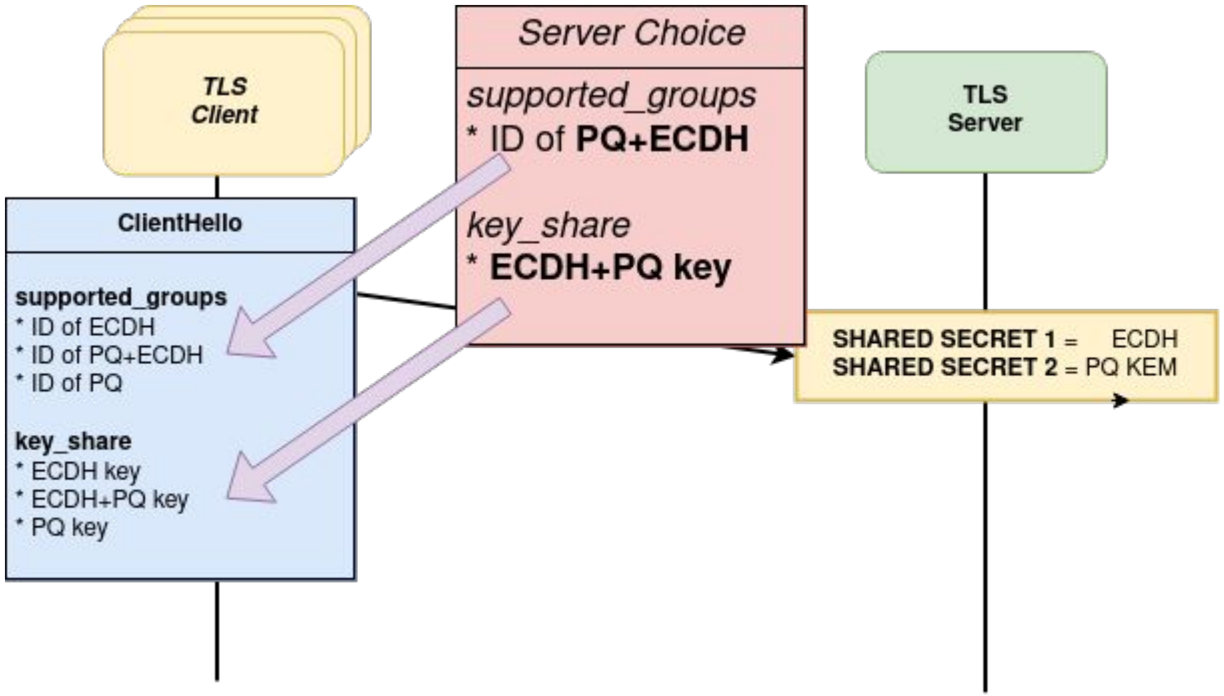
- Server performs KEM encapsulation, Client performs KEM decapsulation.

- Backward compatibility
  - Client sends multiple key shares
  - **Pros**: simplicity, **Cons**: duplication of data

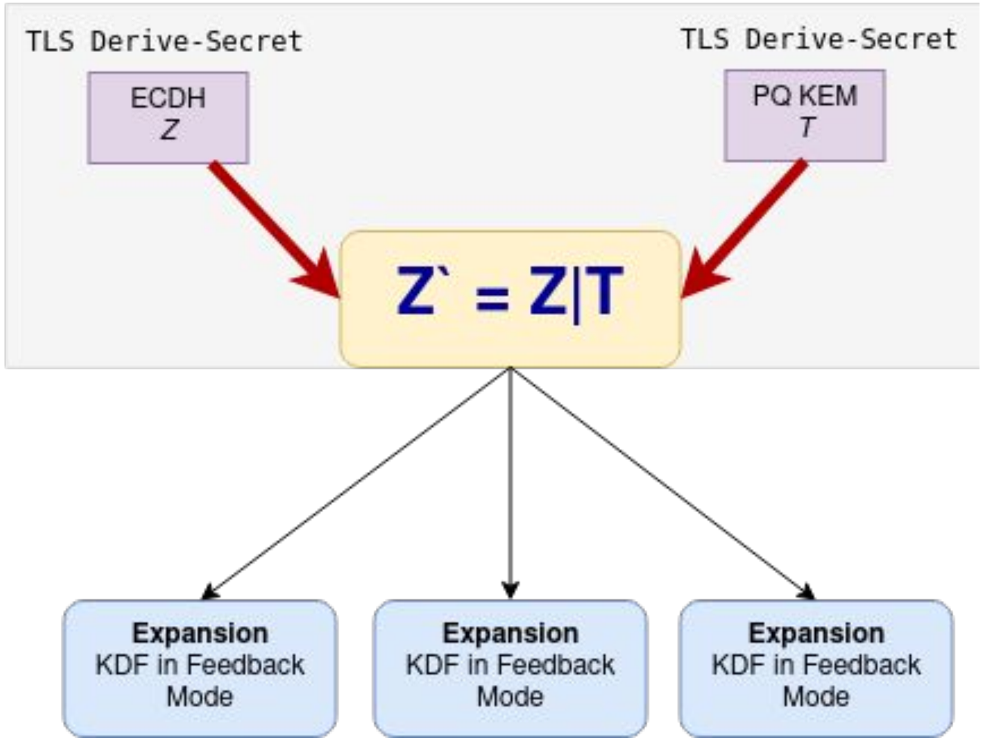
- Forward compatibility
  - **TLS HelloRetryRequest** used in case of different PQ scheme supported by the server (useful during migration)



Migration use case
Server supports classical and hybrid-PQ scheme

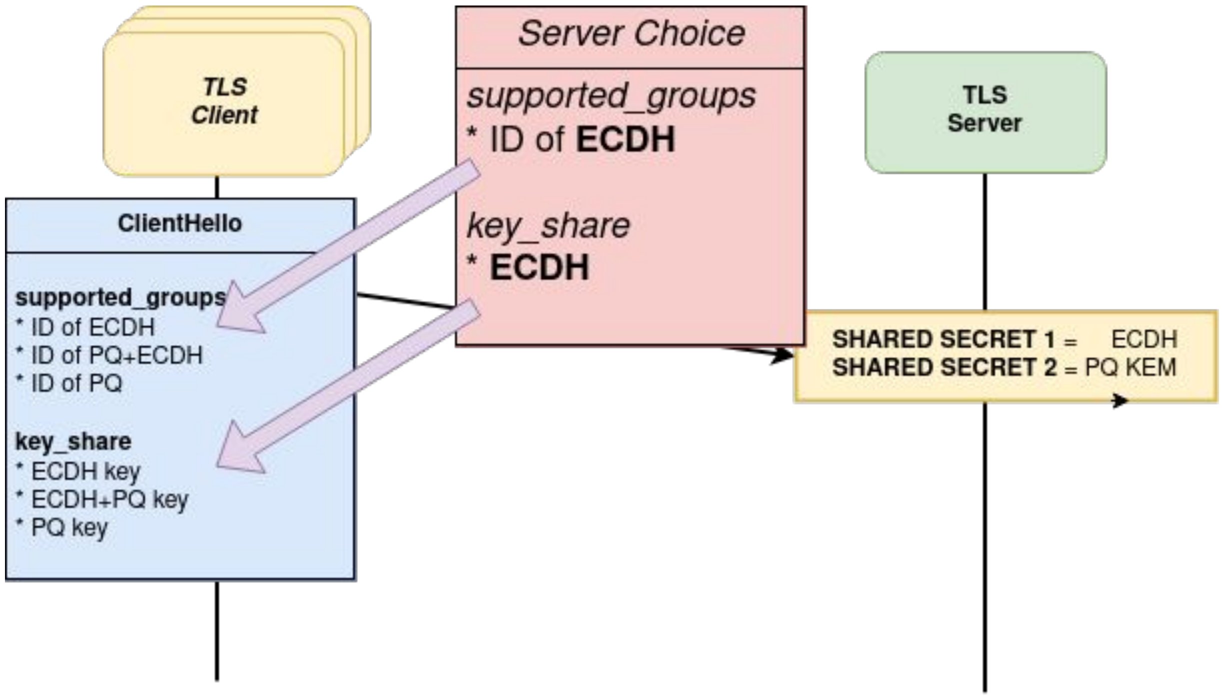


TLS Integration

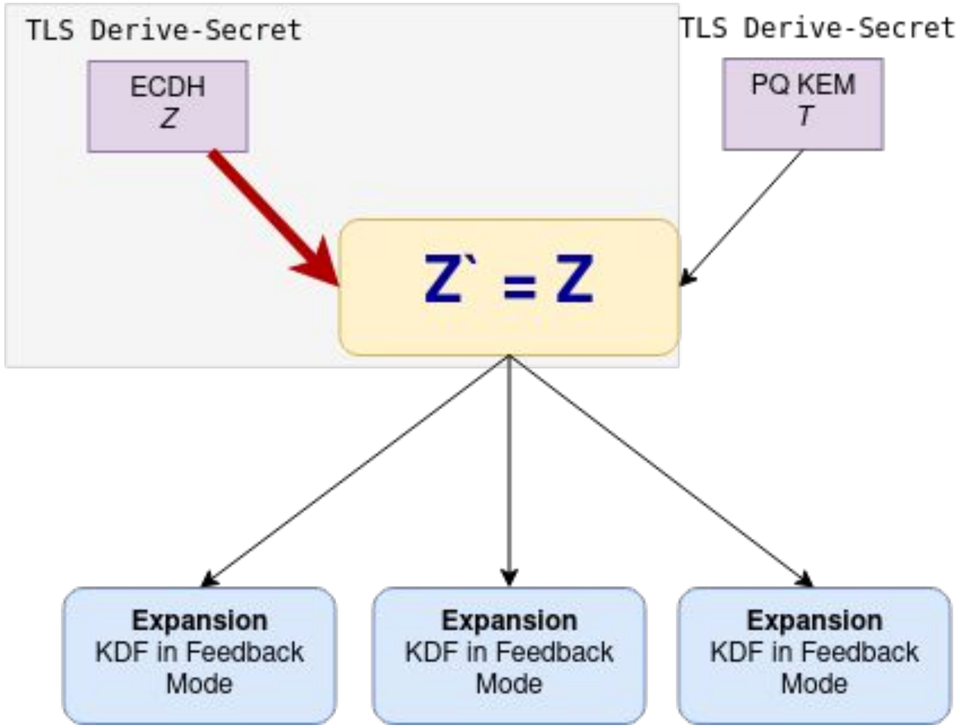


SP800 56C rev2

Migration use case
non-PQ aware server

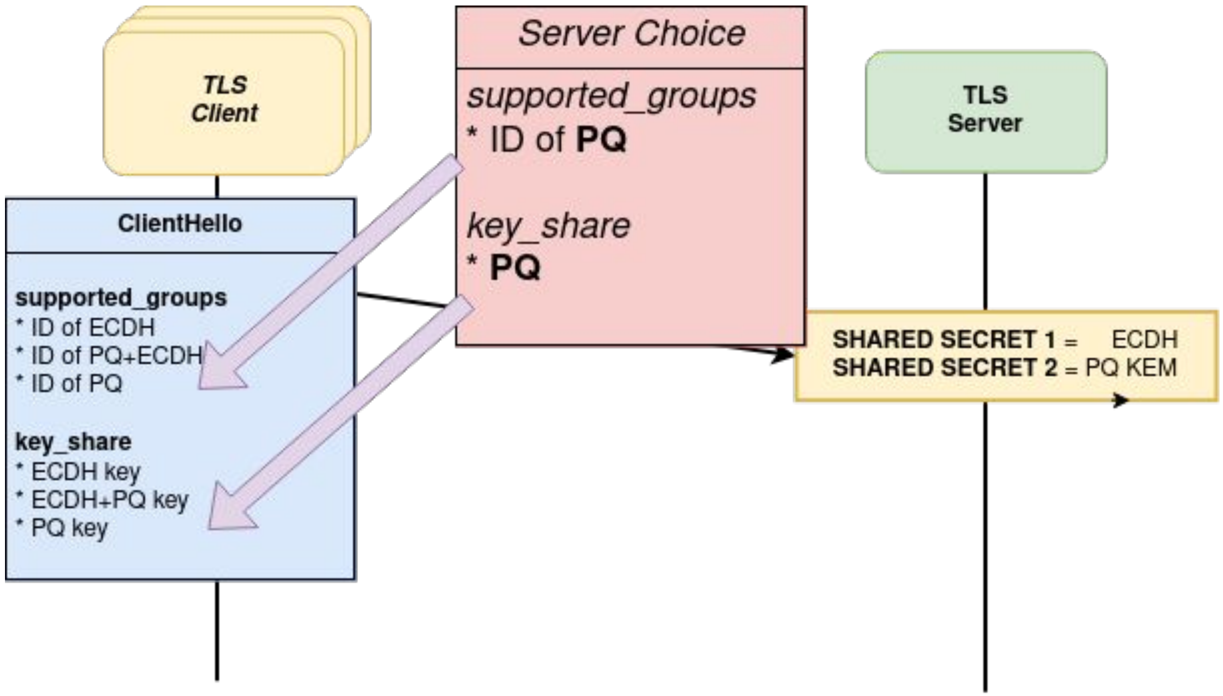


TLS Integration

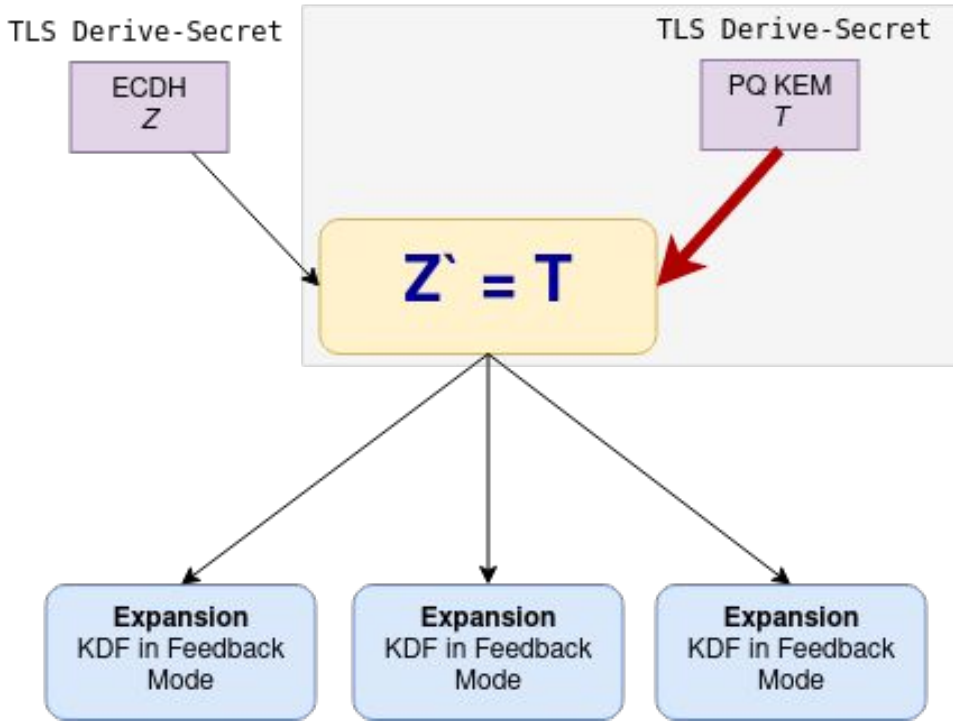


SP800 56C rev2

Migration use case
Server supports PQ scheme



TLS Integration



SP800 56C rev2

Security
Analytical Standpoint

- **Concatenation of two keys is modelled as dual-PRF**
  - dual-PRF is a PRF when either of both keys, guarantees pseudo-randomeness of the output, even if one of the keys is maliciously chosen (or broken) (Bellare and Lysyanskaya “*Symmetric and Dual PRFs from Standard Assumptions: A Generic Validation of an HMAC Assumption*”, 2015)
  - HKDF-extract (based on SHA2) can be modelled as dual-PRF combiner (Bindel et al. “*Hybrid key encapsulation mechanisms and authenticated key exchange.*”, 2019)

- **TLS v1.3 permits to use SHA2-256 and SHA2-384 in the key schedule.**
  - Both are believed to be quantum-safe

- **Shared secrets used by HKDF-extract have fixed length**
  - Required by security proof of dual-PRF combiner (as described by Bindel).

ETSI

TS 103 744: “Quantum-safe Hybrid Key Exchanges”

CatKDF

Form *secret* = [psk] || k_1 || k_2 || …

Set *f_context* = f(context, msg_A, msg_B)

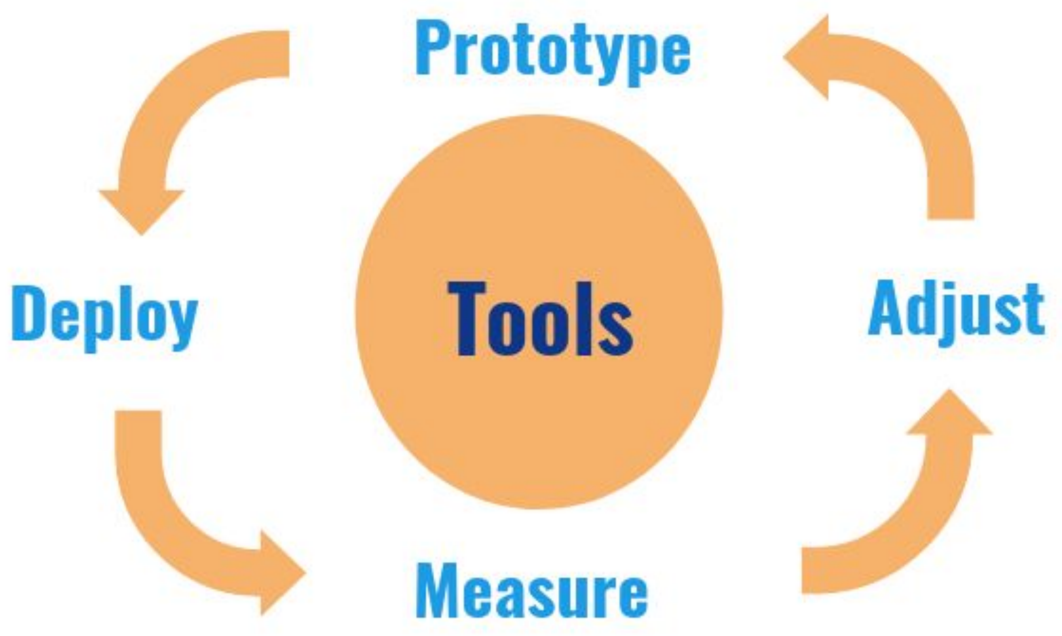
session_key = KDF(*secret*, *f_context*, …)

PQCryptoLib
PQShield's FIPS-certification

- Software Library of modern cryptographic primitives
  - C/C++ interfaces

- Supports TLSv1.3 key exchange
  - Allows to run workloads requiring FIPS 140-3 certification
  - FIPS CVL certificate

- Providing an option to use PQ schemes as "hybrid" key derivation
  - Production code
  - CM executes PQ in "Approved Mode" of operation
  - No security claimed on "SSPs" produced by PQ KEM
  - KAS-SSC and KDA certificates



Last Updated: 5/16/2022

Module Name	Vendor Name	Standard	IUT Date
PQCryptoLib	PQShield Ltd.	FIPS 140-3	5/2/2022

Questions