

Implementation of a Post-Quantum Hybrid Group Key Exchange Protocol

Workshop on Secure Protocol Implementations in the Quantum Era (SPIQE 2026)

Tomáš Fabšič, Samuel Klement, Zoltán Raffay, and **Pavol Zajac**

Slovak University of Technology in Bratislava
`{name.surname}@stuba.sk`

Friday, July 10, 2026

Post-conference workshop at Euro S&P 2026
Lisbon, Portugal

Acknowledgment: Supported in part by the NATO SPS project G5985 and VEGA grant 1/0105/23.

Presentation Agenda

- **Motivation**
- **Protocol Architecture and Core Concepts**
- **Implementation and Network Topology**
- **Challenges, Discussion and Lessons Learned**

The Quantum Threat and Modern Mitigations

- **Infrastructure at Risk:** Classic primitives like RSA and Diffie-Hellman based on (EC)-DLP are vulnerable to sufficiently large quantum computers.
- **Post-Quantum Cryptography (PQC):**
 - Software-based algorithms (such as ML-KEM and ML-DSA) secure against quantum adversaries.
 - Requires significant engineering and infrastructure migration effort.
- **Quantum Key Distribution (QKD):**
 - Physical link protection relying on quantum mechanics (non-clonability).
 - Restrained by specialized hardware requirements and point-to-point topologies.

Proposed Solution

A scalable group authenticated key exchange protocol that can combine the infrastructure flexibility of PQC and the hard physical security of quantum links.

NATO SPS Project G5985

Protocol implementation was developed under the NATO Science for Peace and Security (SPS) Project G5985: *Secure Communication via Classical and Quantum Technologies*.

Project participants

- **Lead:** University of Alabama in Huntsville (USA).
- **PQC:** Universidad Carlos III de Madrid (Spain), Slovak University of Technology in Bratislava (Slovakia).
- **QKD:** VTT Technical Research Centre of Finland, Slovak Academy of Sciences.

NATO SPS Project G5985

Protocol implementation was developed under the NATO Science for Peace and Security (SPS) Project G5985: *Secure Communication via Classical and Quantum Technologies*.

Project mission

Post-quantum cryptography (PQC) and quantum cryptography, including quantum key distribution (QKD), offer different capabilities for securing sensitive communication in the decades to come. This project designs, analyzes, and implements a cryptographic protocol for secure group communication on distributed, hybrid networks, integrating PQC and QKD as available on the distributed network infrastructure. We explicitly address the critical task of key management and train young scientists across the PQC and QKD research communities.

<https://www.quantum-safe-cryptography.science/home>

Protocol implementation at FEI STU

Our focus within the project: Training of young scientists while preparing a secure implementation of the hybrid protocol (developed by Steinwandt and Gonzalez Vasco).

Implementation history

- Started as a team project (5 MSc students, 2024/25), initial prototype.
- Prototype presented at project meeting (Sep 29, 2025), network problems required some rework.
- Dedicated update and polishing (2 MSc students, 2025).

Open source implementation available: <https://github.com/FEI-PQGCH>

Protocol Architecture (Steinwandt and Gonzalez Vasco)

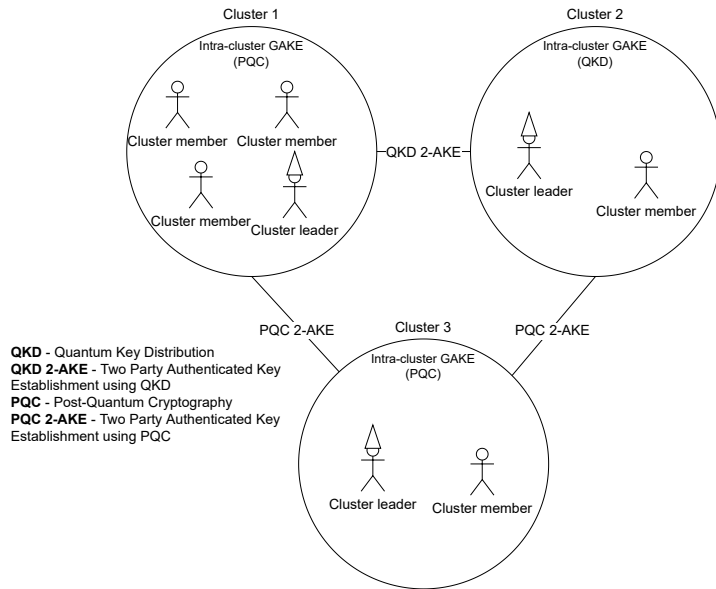
The system organizes a heterogeneous communicating group into a partitioned network:

- **Cluster Partitioning:** The group is split into $n \geq 1$ non-empty clusters:

$$P = P^{(0)} \uplus P^{(1)} \uplus \dots \uplus P^{(n-1)}$$

- **Cluster Leaders:** Each cluster $P^{(i)}$ fixes a designated leader $\widehat{P}^{(i)}$, forming a logical ring topology.
- **Cryptographic Primitives:**
 - ① **Intra-cluster:** Group Authenticated Key Exchange (GAKE⁽ⁱ⁾) creating a shared cluster key $csk^{(i)}$.
 - ② **Inter-leader:** 2-party Authenticated Key Establishment (2-AKE⁽ⁱ⁾) between neighboring ring leaders.

Hybrid Protocol Architecture Overview



The Three Protocol Phases

The hybrid protocol executes across three sequential phases:

Step A: Local Key Establishment

Each cluster establishes its internal $csk^{(i)}$, while ring leaders execute pairwise 2-AKE sessions with their left and right neighbors.

Step B: Leader Commitments

Leaders commit via a public random oracle H to an XOR-combination of their neighboring keys, broadcast, and later open it.

Step C: Derivation and Transport

Leaders verify openings, reconstruct ring keys, derive the global group session key, and distribute it inside clusters via one-time-pad masking.

Implementation Details

A practical demonstration application was built to validate the architecture:

- **Language Stack:** Developed primarily in **Go** with supporting routing infrastructure written in **Rust**.
- **PQC Layer:** Instantiated via an existing Kyber-GAKE software framework¹.
- **Quantum Layer:** Managed via an ETSI API interface connector or local key-file abstractions.
- **Data Serialization:** Fully encoded using custom, easily parsing JSON formats.

¹J. I. Escribano Pablos and M. I. González Vasco, "Secure post-quantum group key exchange: Implementing a solution based on Kyber," IET Communications, vol. 17, no. 6, pp. 758–773, 2023.       

Data Serialization and Configuration Formats

```
{
  "sendId": 1,
  "clusterId": 0,
  "recvId": 2,
  "type": 1,
  "sender": "Alice",
  "content": "SGVsbG8sIGhvdvBhcmUgeW91Pwo="
}
```

Figure: Custom JSON protocol message format.

```
{
  "server": "1.1.1.1:9000",
  "name": "Alice",
  "clusterID": 0,
  "cluster": {
    "memberID": 0,
    "nMembers": 4,
    "publicKeys": "cluster_public_keys.json",
    "secretKey": "secret.json"
  }
}
```

Figure: Cluster member configuration.

```
{
  "server": "2.2.2.2:9000",
  "name": "Leader1",
  "clusterID": 0,
  "cluster": {
    "memberID": 3,
    "nMembers": 4,
    "publicKeys": "cluster_public_keys.json",
    "secretKey": "secret.json"
  },
  "leaders": {
    "nClusters": 3,
    "leftCrypto": "pk_left.json",
    "rightCrypto": "pk_right.json",
    "secretKey": "secret_leader.json"
  }
}
```

Figure: Cluster leader configuration layout.

Integrating Quantum (QKD) Links

QKD pipelines abstract the key source so the upper layers remain unchanged:

ETSI QKD 014

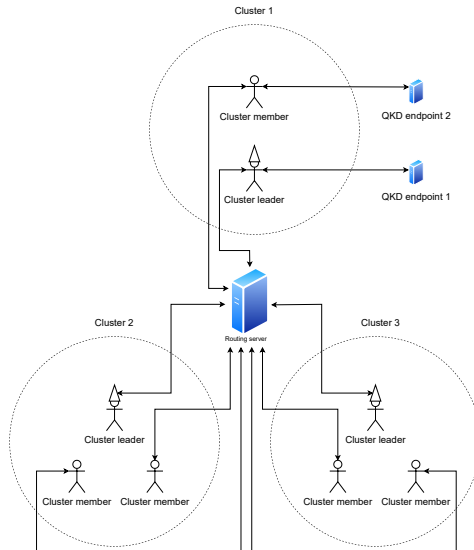
- **Initiator:** Requests random Key and Key ID using `enc_keys`.
- **Receiver:** Uses matching Key ID to collect the identical key via `dec_keys`.
- Cluster leader initiates intra-cluster; left leader initiates inter-leader sessions.

File-Simulated Approach

- Used to bypass operational hardware barriers during tests.
- Requires symmetric access to a local pre-shared key file.
- Connection handling is managed by the local administrator.

Network Configuration: Peer-to-Peer vs. Central Server

- **Peer-to-Peer Initial Design:** Leaders routed traffic directly to neighbors using assigned IP and port endpoints.
- **Real-World Pitfalls:** Complex topologies suffered from firewall drops, strict NAT boundaries, and operational desynchronization.
- **Centralized Routing Solution:**
 - Dedicated routing application written in **Rust** to handle participant traffic.
 - Relays incoming JSON packages using explicit client `memberID` and `clusterID` parameters.
 - **Security Isolation:** The routing server handles transport only; it remains entirely blind to cryptographic operations and key material.



Notes for the PQC Migration

The transition to quantum-safe security is not a simple algorithm swap. Real-world implementation observations:

- Replacing RSA with PQC primitives (like ML-KEM) in isolated libraries is (relatively) straightforward.
- The bottleneck is integrating these primitives into complex, multi-party, and dynamic communication environments:
 - **Topology Restrictions:** Combining software PQC with hardware-bound QKD constraints, and network security restrictions forces specific workarounds.
 - **The Post-Quantum PKI Gap:** Standard certificate infrastructure is not yet optimized for large PQC public keys and signatures.

The Engineering Takeaway

Our project demonstrates that the primary obstacles to a quantum-safe future are not the mathematical primitives themselves, but the system integration, state synchronization, and structural framework constraints.

Deployment and Usability Challenges

- **Topology Trade-offs:** Migrating to a central routing layer fixed firewall issues but introduced a clear Denial of Service vulnerability.
- **Configuration Overhead:** Cluster spaces are static; adding or dropping participants mid-session requires recreating config maps out-of-band.
- **Key Distribution:** Public keys require out-of-band loading; future architectures must combine post-quantum signatures with certified key servers.

Critical QKD Security Note

Querying an ETSI API server via HTTPS over public networks subjects information-theoretic QKD security to the mathematical constraints of standard TLS.

Mitigation: Establish dedicated, physically secure lines for API calls.

Lessons Learned

- **Feasibility Validated:** Successfully prototyped a hybrid group key agreement mechanism running over live messaging channels, protected against active quantum adversary.
- **PQC Migration Challenges:**
 - Using PQC in protocols is easy (ready-made libraries and APIs).
 - Protocol design is difficult, but solutions are already available.
 - Real PQC migration problems lie in key management and network infrastructure constraints.

Thank You!

Questions and Discussion