

A Scalable Cloud-Orchestrated and Service-Oriented Multi-Domain QKD Network with PQC Integration

Konstantinos Krilakis^{1,2†}, Antonia Tsili^{2†},
Katerina Mandilara^{1,2†}, Dimitris Syvridis^{1,2*†}

¹Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Ilisia, Athens, 16122, Greece.

²Eulambia Advanced Technologies Ltd., Agiou Ioannou, Athens, 15342, Greece.

*Corresponding author(s). E-mail(s): dsyvridi@di.uoa.gr;
Contributing authors: konstantinos.krilakis@eulambia.com;
tonia.tsili@eulambia.com; mandkat@di.uoa.gr;

[†]These authors contributed equally to this work.

Abstract

Quantum key distribution (QKD) offers unconditional security but existing QKD networks remain difficult to scale across heterogeneous infrastructures and administrative domains due to vendor-specific interfaces, trusted-node constraints, and limited interoperability. This work presents a flexible multi-domain and multi-site quantum-secure network architecture integrating vendor-agnostic QKD, SDN orchestration, and cloud-managed trust services. Communication is based on Zero Trust Network Access protocols featuring multi-level authentication mechanisms building upon post-quantum cryptography (PQC) signature and key encapsulation algorithms. The system is deployed on a real-world testbed with domains incorporating QKD nodes from 3 vendors, as well as domains without QKD infrastructure elements. Experimental results show that PQC and SDN overhead remain relatively low even on constrained devices, with the main bottleneck being QKD key retrieval and vendor-specific key streaming limitations. The proposed framework extends quantum-safe key transport beyond native QKD boundaries while preserving flexibility, interoperability, and compatibility with existing infrastructures.

Keywords: QKD, PQC, multi-domain network, cloud management, service provision, scalable, trust, multi-factor authentication

1 Introduction

Research in applied cybersecurity is essentially converging towards two emerging frontiers [1] that offer the properties required to safeguard the global network against the dangers of quantum computer cyber attacks; quantum key distribution (QKD) and post-quantum cryptography (PQC). Each approach entails distinct trade-offs. QKD requires specialized and often costly hardware, but can offer information-theoretic security (ITS) [2] under well-defined assumptions. PQC can be easily integrated into existing infrastructures, but misses an equilibrium that balances proven security guarantees and performance efficiency.

Quantum Key distribution is a method for distributing cryptographic keys with theoretically unconditional security. It operates at the physical layer of the network and the basic scheme includes two terminals (Alice and Bob) connected with an ideally dark fibre of limited length. Clients on either Alice or Bob end ultimately receive symmetric cryptography keys through authorised and secure access to the corresponding QKD terminal. This pairwise scheme substantially restricts application range and scalability, especially under the constraint of the devices' increased financial cost. Under certain assumptions and configurations, symmetric keys can be shared between clients on the outer nodes of pairs that are connected through chain-like QKD links. The links are formed using trusted nodes (TNs) [3], and extend the reach of key agreement beyond the pair's distance limitations. Nevertheless, installation of multiple QKD nodes increases the cost even more, and the symmetric keys can only be shared between clients attached to nodes that are parts of this sequential chain at a certain rate. Optical switches could overcome the issue of strict sequential interactions [4], as they offer flexible path reconfiguration, but they introduce additional insertion loss at every switching stage, reducing the secret key rate and maximum achievable distance. Furthermore, optical switches only provide passive routing and do not regenerate or store keys, meaning losses and errors propagate across the entire path. In contrast, trusted nodes terminate the quantum link at each hop, generate fresh keys, and forward them using one-time-pad encryption, enabling longer distances and more stable performance. As a result, trusted-node architectures remain more practical for large-scale QKD deployments.

Several works over the past years have focused on overcoming QKD limitations ranging from theoretical key management and new hybrid protocols that improve the trust over relay nodes [5] to practical QKD network (QKDN) extensions [6–8]. Dervisevic et al. [9] conducted a survey providing a comprehensive system-level overview of key management in QKD networks, focusing on architectures based on trusted relay nodes. The authors highlight the differences between QKD and classical key management, emphasizing the need to decouple key generation and consumption due to the limited rate and point-to-point nature of QKD links. The survey identifies open challenges such as scalability, synchronization, and multi-vendor interoperability. Works

inspired by the decoupling of key generation and consumption propose the integration of key pools, for example Wang et al. [10] introduce an algorithm based on balancing key resources and routing hops by managing resource consumption and QKD key accumulation in pools, in the context of a multi-domain network. Zhu et al. [11] discuss QKD key provisioning to partially-QKD-deployed optical networks using multi-level key pool slicing. The goal is to efficiently accumulate and distribute key material to E2E applications either they have access to key streams of QKD nodes or not, a useful and detailed key management approach which requires extensive configurations, thus a laborious transition of existing QKDNs, which still need to undergo significant scaling [7] challenges. Another set of works have concentrated on the fact that authentication in QKD interactions is not standardised, attempting to efficiently and securely inject PQC methods to this aspect. Kozłovičs et al. [12] propose two protocols supporting the key agreement between two clients that obtain QKD-produced keys, while upgrading TLS 1.3 with PQC algorithms. The idea is that the clients receive parts of the keys from different pathways, further securing key distribution, with servers acting as the interface of the QKD terminals. Atutxa et al. [13] employ PQC digital signatures and study their effect on Quantum Bit Error Rate (QBER), which measures the ratio of incorrect bits to total bits received, acting as an indicator of eavesdropping or channel noise, while Tsili et al [14] create an automated PQC PKI framework that can be deployed on QKDNs. Finally, an applied research direction of interest introduces real QKDNs deployments with PQC integration consisting multiple nodes and TNs [15–17]. Such solutions succeed in demonstrating real-world application and viable adaptations with tangible results – which is the aim behind the current work. Recent efforts have also explored federation mechanisms for extending QKD infrastructures beyond isolated metropolitan deployments. In particular, Barral et al. [18] proposed an ETSI-compliant hybrid PQC/QKD federation architecture for interconnecting geographically separated QKD “islands” through distributed key management overlays and trusted relay federation. Their approach combines QKD-generated secrets with PQC mechanisms to enable secure inter-domain key transport across heterogeneous infrastructures while maintaining interoperability with ETSI GS QKD standards. Similar to MadQCI, the work emphasizes multi-vendor interoperability, distributed orchestration, and practical deployment considerations in carrier-grade environments. However, the federation model remains primarily centered around KMS-level relay coordination and ETSI transport overlays. A detailed breakdown of the corresponding specifications can be found in the Supplementary Information. Here, we present a multi-site enterprise-level network infrastructure and dedicated cloud management designed to enable quantum-secure communication among all network entities (NEs). Contrary to Mendez et al. [16], where PQC is used in some segments of a QKDN for emulating long-distance QKD connections, post-quantum methods are employed to securely conduct the provision of secret key material that could be provided via QKD or other means, and other cryptographic services, over classical channels. As a result, the need for extensive QKDN installations is avoided, while exploiting existing QKD pairs and essentially extending quantum-safe key relay in a way compatible to legacy networks. The architecture is organized into separate, autonomous domains with a central supervisor which provides the capability of

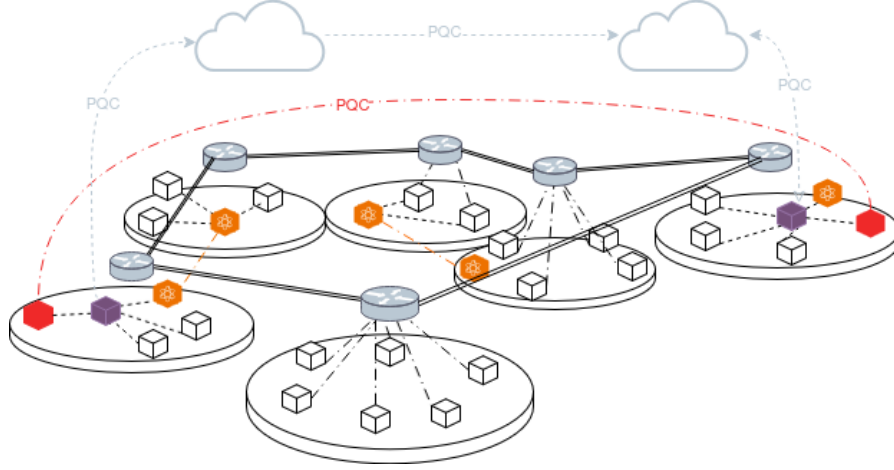


Fig. 1 High-level schematic illustration of the proposed network model. Each disk represents a domain; The white cubes represent client NEs; the purple cubes represent Domain Secure Services Providers (DSSPs); the client application communication with PQC is shown in red; the secure services PQC connections are shown in gray.

the system is the delivery of cryptographic keys generated via QKD to NEs across domains, as well as the creation of secure tunnel links for encrypted data exchange. More specifically, we distinguish the following properties and novelties:

- A native, cross-platform, multi-domain network architecture with support for heterogeneous topologies.
- An agnostic design supported by a zero-touch provisioning (ZTP) driver framework.
- Integration of multi-vendor QKD systems within some domains, combined with PQC KEMs to enable unlimited distance quantum-secure key relay.
- A unified security layer incorporating multi-factor authentication and cloud-based key management for trust, identity, and interoperability.
- SDN-driven automation with detailed telemetry for monitoring and reporting.

The above network was experimentally evaluated on a 12-domain testbed with heterogeneous hardware, quantifying end-to-end setup latency and packet processing overhead. A high-level illustration of the infrastructure is depicted in Figure 1, where each disk represents a domain comprising NEs, a gateway router, a Domain Secure Services Provider (DSSP) and some domains integrate a QKD node as the key generator. The figure shows two kinds of PQC-protected interactions; the direct communication of two NEs located in different domains (red), established through the service provision of the DSSP of each domain, and the management of the DSSPs from the cloud (gray), as cloud-based routers facilitate inter-domain connectivity and aggregation of the domains orchestrated by the visualised-SDN data plane components.

The remainder of this work is organized as follows. In Section 2, we present the resulting multi-domain quantum-secure network architecture and describe its underlying classical and quantum infrastructure. We detail the system operation and the

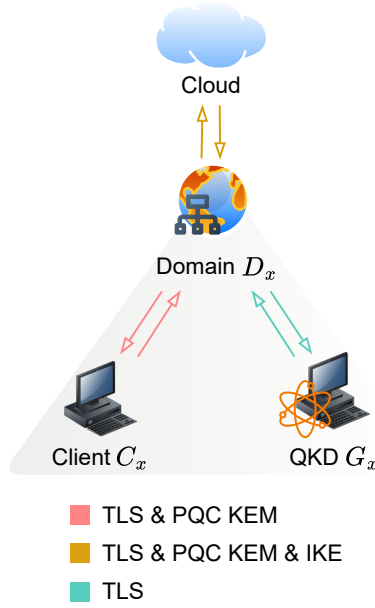


Fig. 2 A domain with a QKD terminal as key generator and the authentication mechanisms per connection type according to the participating peers.

workflows enabling inter-domain key relay across hybrid quantum-classical environments, but also list experimental results derived from our heterogeneous 12-domain deployment. In Section 3, we describe the logical organisation of our system and the main components developed for the construction and seamless operation and orchestration of the offered cybersecurity services. Finally, we analyse the practical implications, limitations, and scalability challenges of the proposed approach, concluding with directions for future work towards large-scale quantum-secure deployments.

2 Results and Discussion

2.1 Network Topology

The proposed work introduces a multi-site enterprise-grade network infrastructure that allows *quantum secure* communication between any network entities (NEs) involved. Our deployment consists of twelve (12) independent network domains, designed after real-world network equivalents, to which NEs and key generators are attached, see for example Figure 2. Formally, an infrastructure domain is defined as a physical or virtual network segment –implemented through hardware NAT boundaries, VLAN isolation, or equivalent mechanisms– that exposes a functional grouping of network resources governed by a specific set of security policies, authentication protocols, and distributed administration rules. The domains’ inner topology is abstracted to a version that can be matched to a wide range of fundamental network units with little to no requirements. Each domain contains a router that functions as a gateway, a low-power device functioning as the Domain Secure Services Provider (DSSP), and

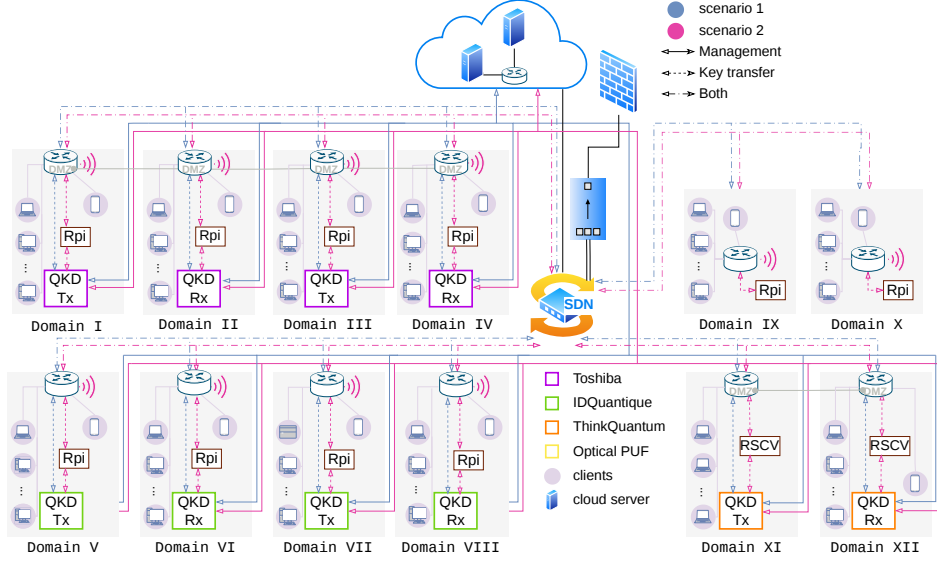


Fig. 3 Detailed overview of the testbed’s architecture. This figure shows explicit connections within the domains and the cloud. Dotted arrow links show key transfer within each domain, solid arrows show management control originating from the cloud, and the remaining interdomain arrows are governed by both connection types. There are three (3) possible scenarios concerning intradomain key transfer, depending on transaction type: the **blue** scenario (1) describes usual QKD function; in the **magenta** scenario (2), QKD keys are processed from the gateway before reaching the client; the third scenario uses a wifi link between the gateway and the client.

at least one key generator and one client. Clients are organised in client pools, where a client pool is a virtualized SDN network pool that combines flow rules and port policies policies to create a programmable network under each hardware domain. We focus on the application wherein the key generators are QKD terminals, thus we distinguish two kinds, the classical or *non-quantum*, and the QKD or *quantum* domains. Classical domains adhere to classical channel communication principles. Quantum domains contain a QKD node each and are vendor-agnostic.

Classical Network

Each administrative domain is defined by minimal design assumptions enabling a technology-agnostic, service-oriented deployment that simplifies management and enhances monitoring. Domains are delimited by hardware gateway routers providing local connectivity, including wireless access where required, while domain environments are realized using heterogeneous computing platforms ranging from constrained Raspberry Pi and Android-powered devices to x64 and RISC-V-based systems. These devices support gateway operations and services deployed within demilitarized zones (DMZs) in conjunction with the router. Domain components interconnect through a combination of VLAN-based traffic isolation, proprietary VPN tunnelling, and identity-based segmentation orchestrated by the cloud control plane, whereas

clients connect to the domain router either directly through Wi-Fi links or virtually through software-defined networking (SDN). The SDN infrastructure employs OpenFlow 1.3 [19] through an ONOS controller [20] hosted within the cloud environment and coordinated with a dual-mode managed switch running in virtualized SDN mode and supporting both aggregated (non-SDN) and SDN-based architectures on the same device. Network Address Translation (NAT) is used to obscure internal topology and retain traffic processing within the SDN data plane, while the SDN switch dynamically redirects traffic according to controller-defined flow rules and port policies, effectively placing clients into different logical positions across segmented network domains. Consequently, each NE can be programmatically assigned to dedicated configuration and operational positions, enabling dynamic establishment and orchestration of domain layouts, as depicted in Figure 3. For simplicity, the topology assumes one Network Interface Card (NIC) per device, although multiple devices support concurrent multi-NIC parallel connections. The network is protected using an open-source Network Access Control and Identity Manager behind a 1G hardware firewall, with authentication signalling between the cloud, domains, and external connections secured via the RADSEC protocol. This consolidates control and monitoring while enforcing security boundaries across external testbeds, using MACsec for hop-by-hop encryption and IPsec for end-to-end protection across multiple hops, creating a federated enterprise network. Our implementation remains fully independent of the OQS project and relies exclusively on portable standardized algorithm releases, ensuring portability and avoiding tight coupling with specific hardware platforms while providing reproducible measurements. Working in parallel with existing directory-service models, the system is deployed as an additional non-discoverable hidden directory-service layer coordinated through compatible upper management levels. Furthermore, cryptographic functionality is implemented through a lightweight custom application-layer library with minimal kernel dependencies (e.g., KTLS), rendering the software transferable, transparent, and readily integrable with existing applications.

QKD Network

At the lowest level and closest to the physical characteristics of the system lies the quantum layer, which is largely driven by the configuration of the QKD nodes. The quantum layer comprises pairs of QKD nodes running a vendor-specific version of the BB84 protocol [21], among which key agreement takes place in the quantum channel, and pairs that form TNs. TNs operate as repeaters in QKD networks, extending the physical reach of the key agreement process and creating a single end-to-end logical chained link between QKD nodes. This functionality requires that the QKD devices forming the TN are purposely configured for this reason, while remaining in a protected environment. From an operational perspective, each QKD node is associated with a specific administrative domain managed by a DSSP, which acts as a trusted proxy for inter-domain key relaying. When forwarding a key K to a neighbouring domain, the DSSP retrieves a locally generated QKD key K' from the quantum layer and computes $K \oplus K'$. The resulting value is then transmitted to the next domain, where the corresponding DSSP can recover K using the shared key

material. In this way, secure end-to-end key transport across multiple administrative domains is achieved without directly exposing the original key during transit. Here, we assume the existence of inter-domain links that lack the interoperability and protection mechanisms required for seamless trusted-node operation, particularly in multi-vendor deployments. We refer to such incomplete relay links as dangling chain edges (DCEs). The chain segments between DCEs are reinforced with PQC KEMs, thus filling the chain’s non quantum-resistant gaps and yielding a chained link of arbitrary length. We exhibit this paradigm on a testbed of five (5) QKD pairs, two (2) Toshiba MU [22] pairs, two (2) ID Quantique Clavis XG [23] pairs and a ThinkQuantum QUKY [24] pair. Photographs of the physical devices can be found in the Supplementary Information.

TN emulation

We implement an emulated equivalent of a conventional TNs via a point to point topology, which can further safeguard the communication of DCEs, provided that a MACsec tunnel has been previously setup. The tunnel can be deployed either as cloud-hosted Secure Access Service Edge (SASE) endpoints or as static on-premises endpoints. The endpoint service operates a PQC-KEM-secured post-quantum mutual TLS streaming encryptor/decryptor pair that accepts only authorized DSSPs as peers. Each pair is instantiated with dynamically defined protocol parameters, dedicated PQC KEM root keys, and post-quantum mTLS certificates, making every connection uniquely bound and non-interchangeable. A schematic of the proposed setup can be found in the Supplementary Information.

Encryptors

The DSSP of domains 1, 2, 3, 4, 9, and 10 have integrated software encryptors that are cross-platform, crypto-agile packet encryptors designed to deliver high-assurance tunneling services between network domains. These software services support multi-domain interoperability and are engineered to operate within heterogeneous cryptographic environments, providing multi-layer encryption, identification through PQC key pairs and multi-vendor QKD support. Configuration is performed through a PQC-KEM-secured network management interface, which communicates directly with the cloud key management system (KMS). This mechanism ensures post-quantum-resilient provisioning of configuration parameters and cryptographic material. The encryptors support different protocols for key-generating sources, including:

- QKD ETSI 014 [25]
- SKIP
- Direct integration with the cloud KMS as a key-provisioning entity

This flexibility enables seamless operation in hybrid classical-quantum key distribution environments.

2.2 System Operation

Consider the setting where a set of m NEs designated as clients $C^1, C^2, \dots, C^m$ is distributed among domains $D_1, D_2, \dots, D_n$. Domains D_x $x \in [n]$ integrating a QKD node G_x are hereafter called quantum domains, and are denoted D_x^* if they fall into the DCE category (see Section 2). The assignment yields

$$\begin{aligned} &C_1^{y_1}, C_1^{y_2}, \dots, C_1^{y_{p_1}} \text{ of } D_1, \\ &C_2^{y_{p_1+1}}, C_2^{y_{p_1+2}}, \dots, C_2^{y_{p_1+p_2}} \text{ of } D_2 \\ &\vdots \end{aligned}$$

where $y_i \in [m]$, $i \in \{1, 2, \dots, p_1, p_1 + 1, \dots, p_1 + p_2, \dots\}$. Let us denote $C_1^A := C_1^{y_1}$ after ‘Alice’, and $C_2^B := C_2^{y_{p_1+1}}$ after ‘Bob’. We assume that client device C_1^A of domain D_1 , which comprises key generator G_1 , requests cryptographic material that will allow the encrypted communication with client C_2^B of domain D_2 , as in Figures 4-8. It follows that, the ultimate goal for C_1^A and C_2^B is to share the symmetric key $K_{1,2}^{A,B}$ for encryption and decryption. During the domain and client pool creation, each NE is provided with digital certificates and an initial key pair (pk^{KEM}, sk^{KEM}) produced with a PQC KEM, bound to its identity (see Section 3.4). The public key pk^{KEM} is stored on the cloud for peer-to-peer identification and trust establishment.

Intradomain modules and services are consistently accessible within each domain, maintaining their integrity while forming functionally complete units that can operate independently and collaborate seamlessly. Inside the domain D_1 , a client C_1^A is authenticated through multi-factor means. Namely, successful authentication is dependent on TLS and an additional security layer based on the ability of C_1^A to decrypt with their secret key sk_s^{KEM} , confirming their identity. Once the client device C_1^A has been authenticated, an encapsulated key request is sent to the domain gateway. The domain services of D_1 will attempt to gain authenticated access to the cloud services, and retrieve the updated policy restrictions that apply to client C_1^A . Depending on the nature of the request, the client will proceed to perform a key agreement with client C_i^B , $i \in [n]$, or the domain services will prompt the local key generator G_1 to provide key $K_{1,2}^{A,B}$.

In more detail, suppose that client with identity profile C_1^A wants to request a QKD-produced key $K_{1,2}^{A,B}$ from their local QKD terminal. The client logs onto the domain services using their signature $\sigma_{C_1^A}$ and the respective user-driven policy $\Pi_{C_1^A}$ that must be followed is updated. The update is performed by interchange of the domain-facing and cloud services, wherein the client destination is identified by their PQC KEM public key $pk_{C_1^A}^{KEM}$. If authorised, the domain-facing services allow role-based access (RBAC) and produce a token t_{LOG} . The client subsequently uses their profile C_1^A and the received token t_{LOG} to request and receive the QKD-produced key $K_{1,2}^{A,B}$, along with its identifier ID_K , through the domain services. The process is shown in full in Figure 5. Next, consider the case wherein C_1^B has initiated key agreement with C_1^A ; cloud key, domain and identity management will perform multi-layer

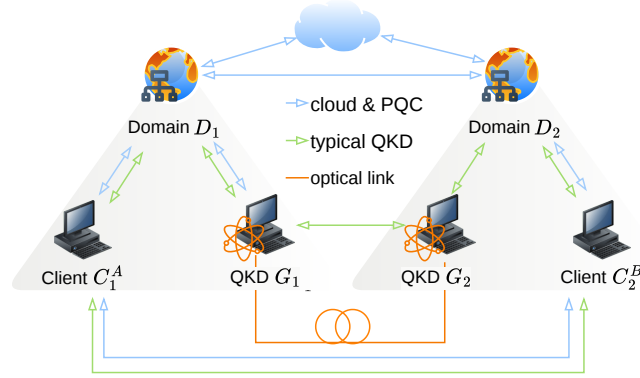


Fig. 4 A scenario wherein the users belong to different quantum domains and the QKD nodes form an Alice-Bob pair. The green arrows represent conventional QKD usage. The blue arrows represent QKD-derived key sharing through forwarding PQC encapsulated messages.

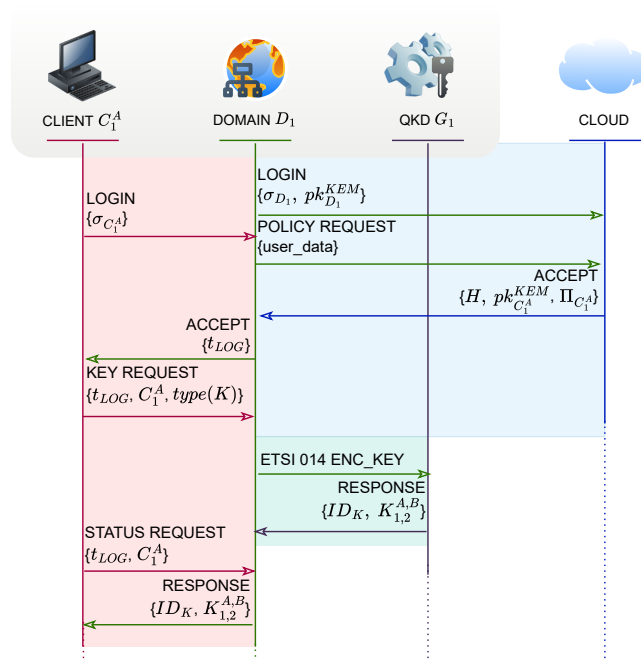


Fig. 5 The intradomain transactions leading to key generation. The client logs onto the domain services, requesting a key from the QKD terminal. Coloured areas denote the authentication protocols: pink for TLS and KEM public key identification; blue for TLS, KEM public key identification and Internet Key Exchange (IKE); green for single-direction or mutual TLS.

authentication between domain and cloud, similar to the previous-stage authentication mechanism, and will supply C_1^B with the means for retrieving the correct key from the local key generator, a.k.a the QKD node G_1 .

2.3 Connecting DCEs for Multi-domain Key Distribution

Our vendor-agnostic approach to key relay is based on cloud-managed post-quantum key encapsulation and authentication, as well as SDN functionality, all conveniently complementing and reinforcing QKD across domains. PQC mechanisms are efficiently exploited, enhancing the resilience and forward secrecy of communications, while cloud management ensures that only legitimate and continuously validated users access the provided security services. Ultimately, one key can be shared between two NEs without distance limitations following a two-phase process described below. The handshake phase ensures transfer only to strictly authorized parties, following the Zero Trust Network Access (ZTNA) premises, while the transfer phase defies vendor-specific integration, e.g. as required by interfaces conforming to ETSI GS QKD 015 [26].

Handshake Phase

The functionality described in Section 2.2 is applied consistently across all domains. Initially, domains D_1 and D_2 log onto the cloud services, updating the policies and permissions applicable to their corresponding clients C_1^A and C_2^B , and the clients, in turn, log onto the respective domain services. These login steps of this process are the same as in Figure 5 for all endpoints of the respective domains D_1 and D_2 . All clients expose a service API specifically purposed to serve communication intentions. Hence, once clients C_1^A and C_2^B have received a respectively unique login token t_{LOG} for service access, C_2^B enrolls to the services exposed by C_1^A . Client C_1^A will send its profile data, the login token and a token $t_{1,2}^{A,B}$ to the cloud through the domain services. Client C_1^A stores token $t_{1,2}^{A,B}$ on the cloud, in order to be shared with the communication peer, and expects to receive it from C_2^B as proof that it belongs to a known, authenticated, and accessible domain, in this case D_2 . Afterwards, C_1^A will communicate the identifier ID_t to C_2^B . The latter, once authorized, will resolve the token $t_{1,2}^{A,B}$ with identifier ID_t by accessing the cloud through the domain services, then encapsulate and present $t_{1,2}^{A,B}$ to C_1^A , thereby establishing trust.

Transfer Phase

Assuming that clients C_1^A of domain D_1 and C_2^B of domain D_2 need to share the private symmetric key $K_{1,2}^{A,B}$ and that key generators G_i of domain D_i , $i \in [n]$ and G_j , $j \in [n]$ of domain D_j are QKD terminals, we distinguish the following cases depending on their relation:

- Key generators $G_1 \equiv G_i$ and $G_2 \equiv G_j$ are a QKD pair, as in Figure 4. The DSSP of the source domain D_1 acts as a trusted proxy by masking the relay key $K_{1,2}^{A,B}$ with a second QKD-generated key $K'_{1,2}$ using the operation $K_{1,2}^{A,B} \oplus K'_{1,2}$. The result is transmitted to the adjacent domain, which then reconstructs $K_{1,2}^{A,B}$ using the corresponding shared QKD key material.

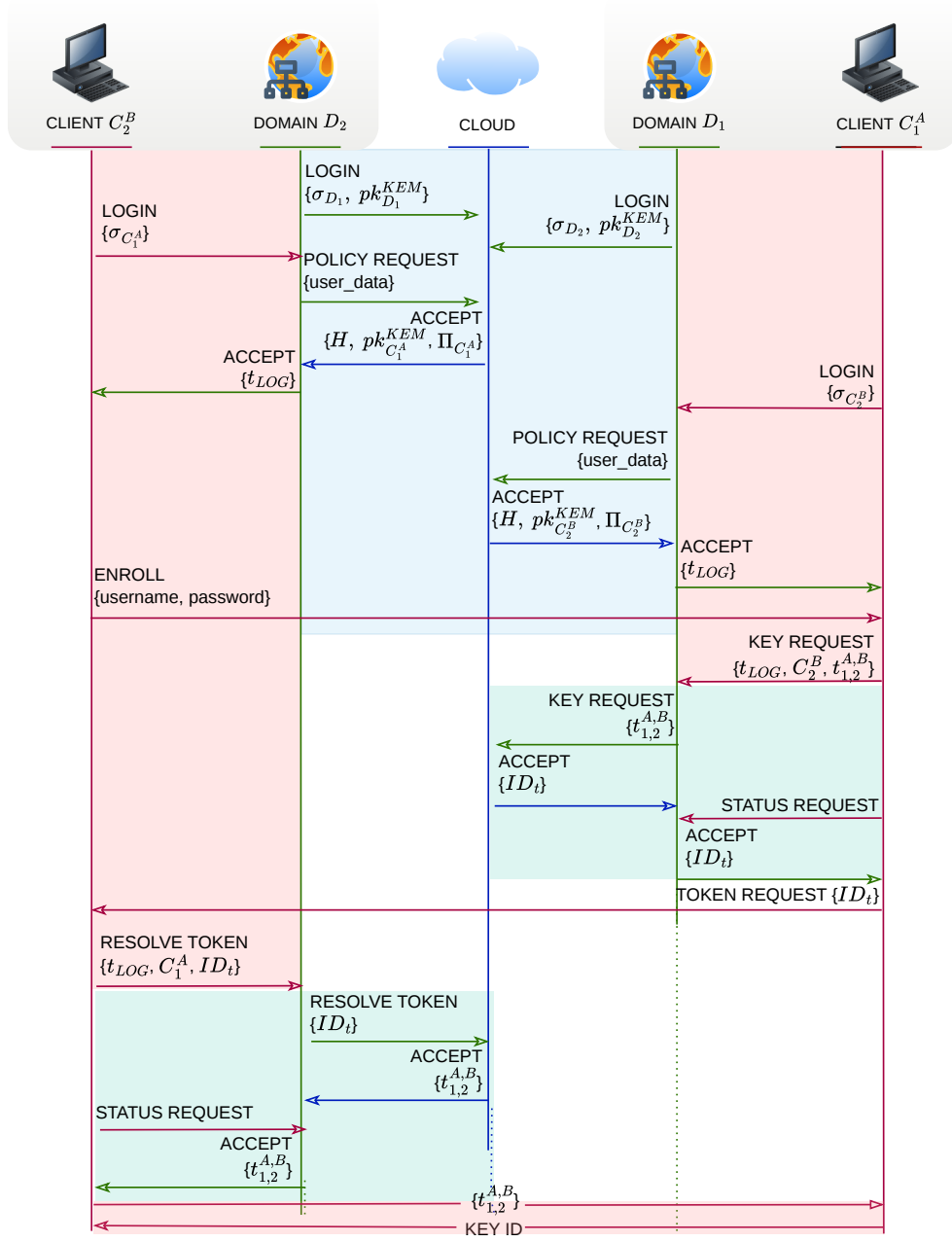


Fig. 6 The interdomain transactions for key agreement between clients C_1^A and C_2^B . The first login steps are the same as in Figure 5. Then, C_2^B enrolls to the services exposed by C_1^A and C_1^A stores a token on the cloud. The peer receives and resolves the token, which is used to obtain the ID of the symmetric key to be used for encryption. Coloured areas denote the authentication protocols (see Figure 5).

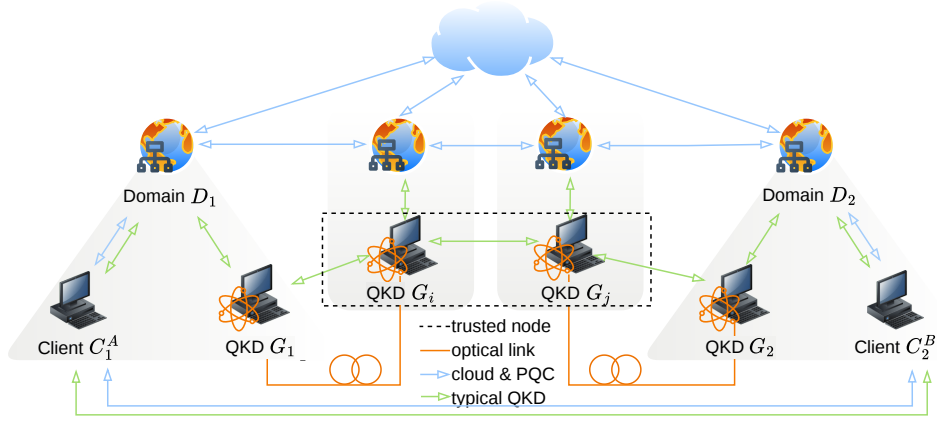


Fig. 7 The clients are placed in two remote quantum domains with QKD terminals connected through a TN. The green arrows represent the conventional QKD usage. The blue arrows represent QKD-derived key sharing through forwarding PQC encapsulated messages.

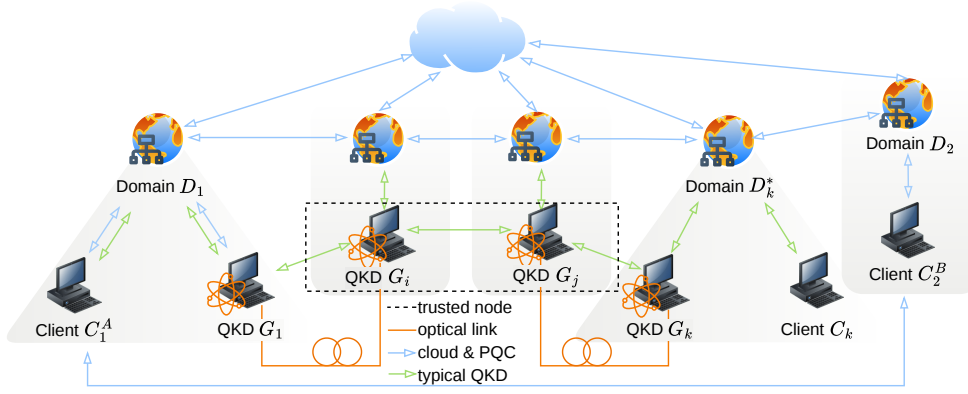


Fig. 8 The clients are placed in two remote domains, a quantum D_1 and a non-quantum D_2 . The green arrows represent the conventional QKD possible interactions, which are limited among domains D_1, D_i, D_j, D_k^* . The blue arrows represent the extended QKD-derived and quantum-safe key sharing offered by our implementation through forwarding PQC encapsulated messages.

- Key generators G_1 and G_2 are parts of QKD pairs connected through a TN comprising terminals G_i, G_j , as in Figure 7. The DSSP of the source domain D_1 masks the relay key $K_{1,2}^{A,B}$ with a locally shared QKD key K_1 using $C_1 = K_{1,2}^{A,B} \oplus K_{1,i}$ and forwards the result through the intermediate domains; the trusted node recovers the

- original key as $K_{1,2}^{A,B} = C_1 \oplus K_{1,i}$. The trusted node then establishes a new QKD-derived key $K_{j,2}$ with the next domain, computes $C_2 = K_{1,2}^{A,B} \oplus K_{j,2}$, and forwards C_2 . The destination domain D_2 will recover the relayed key with $K_{1,2}^{A,B} = C_2 \oplus K_{j,2}$.
- Key generators $G_1 \equiv G_i$ and G_j are connected in one of the ways described in the previous cases and client C_2 belongs to a non-quantum domain, as in Figure 8. The DSSP in the source domain first computes $C_1 = K_{1,2}^{A,B} \oplus K_{1,i}$ and forwards C_1 to the first trusted node, which recovers the original key as $K_{1,2}^{A,B} = C_1 \oplus K_{1,i}$. The trusted node then establishes a new QKD-derived key $K_{j,2}$ with the next domain, computes $C_2 = K_{1,2}^{A,B} \oplus K_{j,2}$, and forwards C_2 . The DCE recovers $K_{1,2}^{A,B} = C_2 \oplus K_{j,2}$ and, instead of applying another XOR-based masking operation, encapsulates $K_{1,2}^{A,B}$ using a KEM public key associated with the destination domain, producing a ciphertext C_{KEM} that is transmitted to the destination, where $K_{1,2}^{A,B}$ is recovered through KEM decapsulation.
 - Key generators G_i and G_j are connected in one of the ways described in the previous cases and clients C_1, C_2 belong to domains without a QKD terminal. The source domain encapsulates the session key $K_{1,2}^{A,B}$ using the destination domain's KEM public key, producing a ciphertext $C_{\text{KEM}} = \text{Encaps}(pk_{\text{dst}}, K_{1,2}^{A,B})$, which is transmitted across the network. Upon reception, the destination domain recovers the key by performing decapsulation, $K_{1,2}^{A,B} = \text{Decaps}(sk_{\text{dst}}, C_{\text{KEM}})$, enabling secure end-to-end key delivery without relying on QKD-generated key material.

Using the proposed system, the QKD-derived key material is encapsulated with PQC KEMs and exported under cloud supervision through the described transaction-based workflow. While information-theoretic security is inherently confined to the QKD segment, this approach preserves quantum resistance across domain boundaries and enforces controlled access, traceability, and policy compliance.

2.4 Experimental results

We conducted experiments on a testbed of ten (10) domains containing QKD nodes from three vendors –Toshiba, ID Quantique, ThinkQuantum– alongside two (2) domains without QKD nodes (Figures 3). Each domain includes a gateway router, a low-power DSSP device, and client nodes (Table 1), while an aggregation router enables inter-domain and cloud connectivity via an SDN-enabled data plane. QKD communication relies on ETSI 014 [25].

Connection latency is between 10 and 25 ms per link, with the multi-authentication combination mTLS and PQC KEM included in the measurement. SDN functionality increases latency by a 10 ms SDN overhead. Cloud management assumes a predictable 20-70 ms afflicted by database internal operations plus the 10-25 ms connection margin. Intradomain transactions complete in $t \in [10 + x, 25 + x]$ ms, where $x \in \{20, 250\}$ ms depending on the QKD vendor, while cloud access introduces an overhead of ~ 13 to 52 ms per request. The dominant latency factor is QKD key retrieval, which is approximately one order of magnitude slower than classical cryptographic and networking operations, measurements for which can be seen in Figures 9-11, whereas network jitter remains negligible. The measurements in question correspond to up to

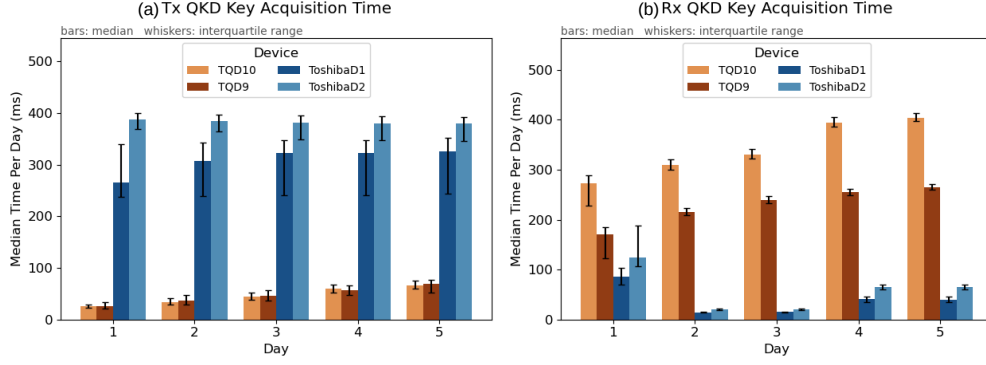


Fig. 9 Key acquisition time measurements over a course of a 5-day continuous usage. Bars show the daily median and whiskers the interquartile range. (a) Measurements of the latency for the DSSP to obtain a key from the corresponding QKD node initiating a key agreement, or Alice (Tx). (b) Measurements of the latency for the DSSP to obtain a key from the corresponding QKD node that responds to a key agreement, or Bob (Rx).

Platform	Role	Specifications
Poweredge R730	cloud servers	32 cores 32GB Ubuntu server 24.04 PCI exp 3.0
Laptop	client	I713700H 20 Cores 16GB Ubuntu 25.04 PCIE 4.0
Raspberry Pi 3b	DSSP	768 MB RAM heatsink no fan 10 MB/s SD card
Raspberry Pi 5	DSSP	2GB 4 core Aarch64 100 MB/s SD card
RISCV64	DSSP	Sifive P550 premier Arm RISCV64 32GB EMMC (no SSD)

Table 1 Devices used in our testbed with their specifications and role.

$\approx 273 \cdot 10^3$ samples per device. Samples above three times a device’s own 99.9th percentile are excluded as stalled sessions — between 0 and 33 samples per device. They show the different behaviour of the devices, causing delays either on the transmitting (Alice) – the node initiating the key agreement (Figure 9(a)) – or the receiving end (Bob) – the node helping to finalise the key agreement (Figure 9(b)). In fact, it becomes apparent that, apart from the ID Quantique nodes, buffer congestion during continuous usage decelerates the process to a point that the devices become inefficient without manual intervention. Here, it is important to mention that ID Quantique nodes provided the least number of measurements due to response failure in many cases. Table 2 reports the processing overhead introduced by encapsulation and decapsulation operations across heterogeneous DSSP platforms. It is worth noting that the RISCV64 device is burdened with a multiplicity (up to 4 times more than the Raspberry Pi 5) of underlying processes due to its default configuration, compared to the rest of the devices.

For PQC, NTRU761 KEM pair generation – measured using a fully portable version of the algorithm without processing optimisation instructions – requires 37.6 ms per device on a laptop and 71 ms in cloud containers, with per-domain generation

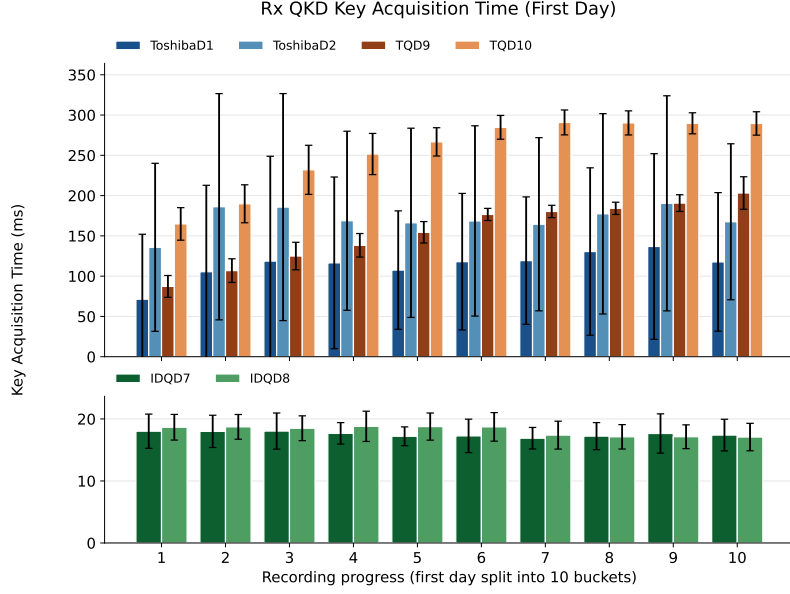


Fig. 10 Key acquisition time measurements over a course of 1 day. The duration of a day has been split in equal-sized 10 buckets. Bars show the bucket mean and whiskers one standard deviation.

Platform	Encryption	Decryption
Raspberry Pi 3B (full desktop)	39.01 ms	86.90 ms
Raspberry Pi 5 (no desktop)	4.24	6.34 ms
RISCV64 (full desktop)	10.35	24.3 ms

Table 2 Mean performance observed measurements for encryption and decryption of the encapsulated packet travelling between domains.

(four pairs) reaching 125 ms and 280 ms respectively. Increasing CPU cores (2 vs. 10) does not improve latency, indicating dependence on single-core performance. PQC operations remain stable across heterogeneous environments. Measurements of the time to prepare the encapsulation packet, together with the overhead of transacting with TLS and transmitting it to the other domain, averaged per platform, are shown in Figure 11.

Certificate generation is dominated by RSA-4096 costs. On the laptop, CA-ICA tree construction (18 certificates) requires 412 ms, while QKD infrastructure initialization takes ~ 1227 ms, with only ~ 27 ms attributed to ECDSA P-384 and ~ 1200 ms to RSA. In cloud environments, initialization increases to ~ 13.4 – 16.9 s, again largely due to RSA generation, with minimal gains from additional CPU cores. Table 3 summarizes the measured certificate and KEM generation overhead across platforms.

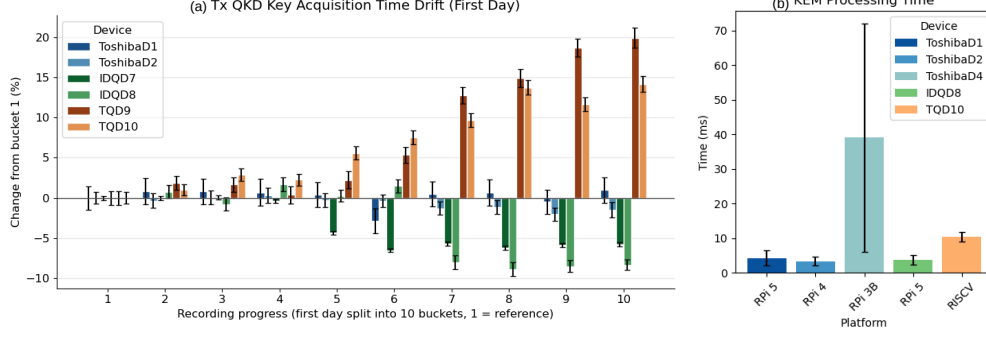


Fig. 11 Drift in key acquisition time over a day and average KEM processing time per DSSP platform. (a) The drift is expressed as the percentage change from each device’s own first-bucket mean. Bucket 1 is the reference and is 0% by construction; whiskers show the standard error of the bucket mean, and the baseline’s own uncertainty is not propagated. (b) Key preparation plus transaction by host platform. Bars show the mean over the full run and whiskers one standard deviation;

Platform	1 KEM pair	4 KEM pairs	CA - ICA (18 crts)	QKD (16 crts)
Laptop	37.58	125.26	411.82	1227.35
Cloud server (2 cores)	71	280.31	4377.59	16888.82
Cloud server (10 cores)	71	280.31	4377.59	13443.88

Table 3 Measurements regarding the creation of key pairs and certificates on each device of Table 1. The KEM used is NTRU761, while a part of the listed certificates (crts) were produced with ECDSA P-384 and the rest with RSA 4096. the results are reported in milliseconds.

Overall, system overhead from SDN control and service orchestration remains low and manageable even on constrained devices. In contrast, QKD key retrieval and infrastructure initialization constitute the primary bottlenecks, with vendor-specific implementations introducing higher latency in exchange for robustness and advanced authentication support. Classical certificate generation is resource-sensitive, whereas PQC operations exhibit consistent performance across platforms.

3 Methods

We present a system with multi-level considerations that offers aggregation and network flexibility through its underlying complex design. Our proposed system comprises a set of individual components whose seamless integration is crucial for ensuring its smooth functionality and wide flexibility, while maintaining high service rates. These modules collectively define the system operation and are discussed in detail in the following section.

3.1 Architectural Planes

The presented system follows a layered architectural model consisting of an administrative plane, an SDN-orchestrated control plane, and a data plane, spanning the control, application, network, and quantum layers shown in Figure 12, the latter being treated separately due to its distinct management and operational requirements. The administrative plane constitutes the management of these layers, defining policies, supervising configuration, and controlling security-critical operations. The SDN-orchestrated control plane maintains a global view of the network and enables dynamic configuration and reconfiguration of network paths in response to topology changes or policy decisions. The data plane is responsible for the execution of communication and processing functions, including the transport of user data, the operation of communication channels, and the application of cryptographic mechanisms. Together, these planes provide a clear separation between governance, decision-making, and execution across the system, with each one representing a different risk priority in case of failure or compromise; the corresponding points of security interest are identified (see Table 4) and appropriately hardened, as explained in the following.

3.2 Administration and Control

The administrative plane supervises the configuration and coordination of system components, without directly participating in their operation. It defines and enforces system-wide policies, manages client permissions and possible actions, and regulates the sources of authorization material, including certificates, public key pairs, and tokens, according to user profiles. These services are domain-specific and supervised by the cloud control plane. Cloud services reside alongside the administration control and support system-wide operations, while the administrative plane manages actions with significant security concern, such as client addition and profile creation. The control plane leverages SDN control and automation – as SDN features that allow dynamic

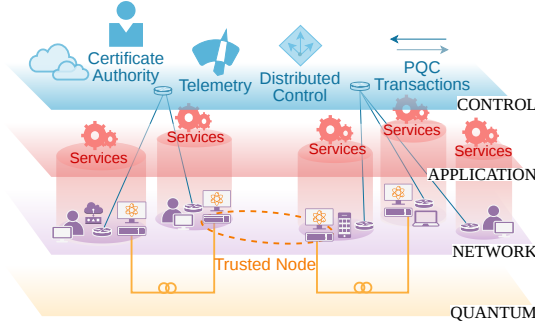


Fig. 12 Overview of the proposed framework's overall organisation. The main implementation components are distributed according to the physical layout in the network. We distinguish four (4) planes – quantum, network, application and control. *Quantum*: defines the quantum channels; *network*: models domains of clients and key generators while specifying TN connections; *application*: outlines service area of action; *control*: distributes crucial management operations among the corresponding cloud services.

configurations and regulated key retrieval become critical [17] – to dynamically configure network paths, isolate traffic, and react to topology or policy changes. In a small network, the cloud controller, serving as the principal channel for dispatching management directives, constitutes the system’s single point of failure; to mitigate this, the distributed nature of the system safeguards lower-layer operations, with DSSPs continuing to operate according to delegated policy profiles, namely providing keys to known clients through intradomain and relay workflows. In scaling scenarios, however, clustering and isolating controller instances allows severing problematic segments.

3.3 Certification, Authentication, Identification and Trust

Authentication is achieved with a strict scheme that involves certificates, PQC KEM key pairs (asymmetric) and policy alignment (e.g domain specific user access, RBAC, etc.). Client permissions and possible actions are explicitly defined and authorization material (certificates, public key pairs, tokens) sources are restrained according to user profile. The proposed authentication framework is based on a multi-level public key infrastructure (PKI) managed from the cloud, that employs separate roots for QKD nodes of different vendors and another root for the rest of the testbed. In more detail, the overall system employs two (2) root Certificate Authorities (CAs), and four (4) Intermediate Certificate Authority (ICA) categories. Separate roots not only ensure compatibility with the QKD nodes, which require distinct chain-of-trust formats, but also preserve integrity among chains and ensure interoperability. Certificates are categorized according to their operational role:

- infrastructure identity certificates for device-level identification,
- service device-bound identification and authorization certificates for client-service interactions, and
- key management trust certificates for secure inter-KMS communication and synchronization.

Certificates are pinned to the respective entity’s Internet Protocol (IP) address, username and password, hence their compromise is either prevented or easily discovered and mitigated according to Table 4. Certificate issuance, renewal, revocation, and policy enforcement are managed by dedicated services operating under cloud supervision. Certificate maintenance guarantees dependency dismantling on revocation or compromise events; in a partial certificate renewal case wherein a certificate becomes invalidated, the old certificate is returned to re-establish the client or domain, reconstructing the part of the network that could potentially accommodate an adversary; otherwise maintenance triggers system overhauling.

Trust is established through explicit cross-validation of service certificates rather than reliance on a single centralized issuer. During domain build, gateways are provisioned with certificates used for mutual TLS (mTLS) authentication and clients are defined by the administrator and securely initialised with digital certificates signed with post-quantum algorithms. Any NEs participating in authenticated interactions are also provided a key pair produced with PQC KEMs, bound to their certified identities, and each activated public key pk^{KEM} is stored on the cloud for peer-to-peer identification and trust establishment. Following initial authentication, the system

Component	Risk	Failsafe/Countermeasure
Cloud controller	failure compromise	Distributed failsafe operation via known neighbours. Architecture hardening via clustering or isolation.
Database	compromise	Multiple users on a need-to-know basis; users assigned per process.
Key management	compromise	Block device and alert human administrator; re-issue certificate (see Section 3.3).
Key generator	failure compromise	Non-quantum domain operation. Device-specific protection.
DSSP	failure compromise	Network redundancy. Protection lockdown/Trusted Firmware Application (TF-A).
Client	compromise	Client block.

Table 4 A list of potential risks associated to the critical modules in the network infrastructure together with failsafe options or protection countermeasures depending on the risk case – compromise or failure.

employs proof-of-possession mechanisms to avoid repeated transmission of long-term credentials. Instead, short-lived cryptographic tokens are issued and used as digital tickets for subsequent transactions. Possession of a valid token cryptographically demonstrates authorization, enabling stateless authentication at the service level while ensuring scalability, replay protection, and minimal exposure of sensitive credentials.

3.4 Key Management

The key management module orchestrates domain deployment and, as such, determines the behaviour of the final system with respect to trust and the cryptographic material’s lifecycle. In more detail, it coordinates the distribution of domains built beforehand and ensures that they are provisioned with certificates and key pairs sufficient to initiate trusted connections. It distributes and maintains certificates for each NE in a domain and connects each NE to a rotating key pair created with the PQC KEM NTRU [27]. Eventually, the KMS provides NEs with an initial transactional KEM pair (pk^{KEM}, sk^{KEM}) from which the public key is stored on the cloud for client discovery. Another pair is ephemeral and updated after each transaction with the help of the domain services, while the last is used for identification. Each sender holds a separate public key for each receiver, making sure that the message can be decapsulated and decrypted by the holder of the corresponding secret key, hence the KEM pair is used as an additional identity validation point.

With regards to key retrieval, our ZTP device driver architecture ensures automatic provisioning with full agnostic nature. On bootup, the DSSP can start in a hardened state: TF-A enables the network interface for the cloud connection and hands it to the DSSP application, but if TF-A’s startup checks on the application fail, lockdown is triggered and the interface is blocked. Key generators are provisioned to the DSSP over a specific secondary Network Interface Controller. This physical isolation provides protected access and identification, as each key generator can be registered to the system and identified via different types of identity claims (e.g. PUF, Printed Circuit Board specific features). The DSSP backend, then, uses a key

generator-specific driver that handles the key requests using the appropriate security claims. leading to interoperability and secure assimilation of any type of device for key generation. Despite low likelihood of the module’s compromise, implications from external sources, such as impersonation with a forged certificate or transaction with under-protected key generators, are possible; in which case immediate restriction of the identified device and administrator notification are triggered.

3.5 Vendor-Agnostic QKD Management

A significant portion of effort addressed the seamless incorporation of the QKD pairs. QKD management is performed from the cloud and is designed to accommodate the requirements of the devices in our testbed, while remaining readily extensible to support additional hardware. This module is meant to efficiently provide access to cryptographic material while maintaining smooth operation and communication with the domain, as commercial QKD devices frequently depend on rigid configuration requirements, limiting seamless integration in large-scale, multi-vendor environments. In our implementation, we found misalignment on four (4) levels, including administration or management, authentication mechanisms, SDN support and key retrieval interfaces.

3.6 Orchestration

The orchestration layer is responsible for the coordinated management and automation of network, control, and quantum resources across domains. It provides consistent configuration, deployment, and supervision of complex scenarios, but also integrates with the SDN controller to allow dynamic configuration, monitoring, and reconfiguration of network paths in response to topology changes or policy decisions. In the context of this work, orchestration plays a central role in the controlled bridging of high-level components with low-level system elements. A key function of the orchestration layer is the support for automated scenario building, whereby network topologies, domain relationships, and service dependencies can be instantiated programmatically. This includes the definition of participating network elements, inter-domain links, and associated control policies. According to the system operation described in Section 2, three main automated scenarios are prepared:

1. **QKD response time:** This test involves all the processes required for the key agreement between a QKD node pair located in separate domains, as well as the retrieval and transmission of its identifier as an encapsulated message across domains. The development of this test allowed to study the rate of key acquisition in a real-world deployment within the context of PQC-protected interdomain connections and revealed the issue of depletion, as some QKD nodes have limited key generation throughput.
2. **Intradomain key retrieval:** This test measures the end-to-end latency of service requests, decomposed into the communication and processing stages between user, device, domain, and key generation entities, including the return path of the generated key material. It captures both communication and processing delays involved in request propagation and key delivery.

3. **Cloud key retrieval:** This test measures the latency of client-domain-cloud exchanges during service enrollment and token handling. It includes client authentication and the transmission and retrieval of tokens and identifiers via domain services and cloud infrastructure.

3.7 Telemetry

A major concern since the early stages of the system’s development has been the design of an elaborate telemetry component that offers transparency, fault detection and localization, in addition to close performance monitoring. Telemetry has been tuned to be device-oriented and to provide a clear image of the network’s status, enabling observability and the rapid identification of problematic segments. To this end, telemetry data are collected and aggregated across domains, allowing both localized and system-wide conditions to be assessed in a consistent manner. The testbed logging has been tuned to avoid obstruction of normal operation, introducing only negligible performance impact while delivering meaningful and actionable updates. Given the high volume of telemetry and event messages generated by the system, device-oriented filtering and aggregation mechanisms have been employed to surface only significant state changes and relevant events, thereby preventing information overload. At the same time, clients are provided with control over telemetry and logging behavior, allowing them to enable, disable, or trigger logging operations manually according to their specific requirements. Logging messages produced cover key acquisition latency, transaction completion, client authentication events, and cryptographic operation timings, such as encryption and decryption. These metrics support both real-time monitoring and post-experiment analysis, contributing to validation, debugging, and performance characterization of the deployed system.

4 Conclusions

The presented work demonstrates a flexible multi-domain quantum-secure network architecture that combines QKD, PQC, SDN orchestration, and cloud-managed trust services into a unified framework. The proposed approach accommodates heterogeneous deployments, including fully PQC-based domains, hybrid quantum-classical infrastructures, and multiple types of key generators through an agnostic ZTP-driven integration model. This allows secure key transport and authenticated communication across domains without requiring homogeneous QKD deployments or tightly coupled vendor-specific infrastructures. The implementation further highlights the practical challenges associated with the integration of multi-vendor QKD systems, particularly with respect to incompatible management interfaces, authentication schemes, orchestration mechanisms, and key retrieval procedures. At the operational level, the limited key streaming and generation rates of current QKD nodes remain a major bottleneck compared to the relatively stable overhead introduced by PQC operations and SDN functionality. Nevertheless, the measurements indicate that the additional orchestration, authentication, and encapsulation layers can be integrated with limited performance impact even on constrained devices, supporting the practicality of the proposed architecture in realistic deployments. Another important aspect of

the proposed framework is its adaptability. The architecture can accommodate a wide range of deployment scenarios, including domains with native QKD support, partially connected trusted-node chains, and purely classical domains relying exclusively on PQC mechanisms for secure key transport. At the same time, the DSSP abstraction and cloud-supervised management model provide a consistent operational interface independently of the underlying cryptographic or networking technologies, simplifying interoperability and service orchestration across heterogeneous environments. Future work will focus on extending the infrastructure towards geographically remote domains and larger-scale deployments involving significantly higher numbers of clients and services. Such scaling introduces additional challenges related to orchestration complexity, latency accumulation, inter-domain trust propagation, certificate lifecycle management, QKD key depletion, telemetry aggregation, and the scalability of centralized cloud and SDN control services. Nevertheless, the modular and vendor-agnostic design of the proposed architecture provides the flexibility required to progressively address these challenges while enabling the evolution towards practical large-scale quantum-secure networks.

References

- [1] Chamola, V., Jolfaei, A., Chanana, V., Parashari, P., Hassija, V.: Information security in the post quantum era for 5G and beyond networks: Threats to existing cryptography, and post-quantum cryptography. *Computer Communications* **176**, 99–118 (2021) <https://doi.org/10.1016/j.comcom.2021.05.019> . Accessed 2025-05-14
- [2] Xu, F., Ma, X., Zhang, Q., Lo, H.-K., Pan, J.-W.: Secure quantum key distribution with realistic devices. *Rev. Mod. Phys.* **92**, 025002 (2020) <https://doi.org/10.1103/RevModPhys.92.025002>
- [3] Salvail, L., Peev, M., Diamanti, E., Alléaume, R., Lütkenhaus, N., Länger, T.: Security of trusted repeater quantum key distribution networks. *Journal of Computer Security* **18**(1), 61–87 (2010) <https://doi.org/10.3233/JCS-2010-0373> . Accessed 2026-01-29
- [4] Yang, Y.-H., Li, P.-Y., Ma, S.-Z., Qian, X.-C., Zhang, K.-Y., Wang, L.-J., Zhang, W.-L., Zhou, F., Tang, S.-B., Wang, J.-Y., Yu, Y., Zhang, Q., Pan, J.-W.: All optical metropolitan quantum key distribution network with post-quantum cryptography authentication. *Optics Express* **29**(16), 25859 (2021) <https://doi.org/10.1364/OE.432944> . Accessed 2026-02-09
- [5] Piétri, Y., Verdier, P.-E., Lacour, B., Gautier, M., Huang, H., Camus, T., Pegon, J.-S., Zuber, M., Faugère, J.-C., Schiavon, M., Rhouni, A., Jaouën, Y., Fabre, N., Alléaume, R., Rivera, T., Diamanti, E.: Quantum Key Distribution with Efficient Post-Quantum Cryptography-Secured Trusted Node on a Quantum Network. *arXiv*. arXiv:2504.01454 [quant-ph] (2025). <https://doi.org/10.48550/arXiv.2504.01454> . <http://arxiv.org/abs/2504.01454> Accessed 2026-04-23
- [6] Chen, T.-Y., Jiang, X., Tang, S.-B., Zhou, L., Yuan, X., Zhou, H., Wang, J., Liu, Y., Chen, L.-K., Liu, W.-Y., Zhang, H.-F., Cui, K., Liang, H., Li, X.-G., Mao, Y., Wang, L.-J., Feng, S.-B., Chen, Q., Zhang, Q., Li, L., Liu, N.-L., Peng, C.-Z., Ma, X., Zhao, Y., Pan, J.-W.: Implementation of a 46-node quantum metropolitan area network. *npj Quantum Information* **7**(1), 134 (2021) <https://doi.org/10.1038/s41534-021-00474-3> . Accessed 2026-04-27
- [7] Horoschenkoff, P., Henrich, J., Böhn, R., Khan, I., Rödiger, J., Gunkel, M., Bauch, M., Benda, J., Bläcker, P., Eichhammer, E., Eismann, U., Freck, G., Griesser, H., Jontofsohn, W., Kopshoff, N., Röhrich, S., Seidl, F., Schark, N., Sollner, E., Blanckenburg, D.v., Heinemann, A., Stiemerling, M., Gärtner, M.: DemoQuanDT: A Carrier-Grade QKD Network. *Journal of Optical Communications and Networking* **17**(9), 743 (2025) <https://doi.org/10.1364/JOCN.563470> . arXiv:2503.21186 [quant-ph]. Accessed 2025-08-20
- [8] Chen, H.-Z., Li, M.-H., Wang, Y.Z., Zhao, Z.-G., Ye, C., Li, F.L., Chen, Z., Han, S.-L., Tang, B., Miao, Y.J., Qi, W.: Implementation of carrier-grade quantum

- communication networks over 10000 km. *npj Quantum Information* **11**(1), 137 (2025) <https://doi.org/10.1038/s41534-025-01089-8> . Accessed 2026-04-27
- [9] Dervisevic, E., Tankovic, A., Fazel, E., Kompella, R., Fazio, P., Voznak, M., Mehic, M.: Quantum Key Distribution Networks - Key Management: A Survey. *ACM Computing Surveys* **57**(10), 1–36 (2025) <https://doi.org/10.1145/3730575> . Accessed 2025-05-14
- [10] Wang, Q., Yu, X., Zhu, Q., Zhao, Y., Zhang, J.: Quantum key pool construction and key distribution scheme in multi-domain QKD optical networks (QKD-ON). In: Lu, C., Cai, Y., Chen, F., Li, Z. (eds.) 4th Optics Young Scientist Summit (OYSS 2020), p. 63. SPIE, Ningbo, China (2021). <https://doi.org/10.1117/12.2591316> . <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11781/2591316/Quantum-key-pool-construction-and-key-distribution-scheme-in-multi/10.1117/12.2591316.full> Accessed 2025-12-22
- [11] Zhu, Q., Yu, X., Zhao, Y., Nag, A., Zhang, J.: QKD Key Provisioning With Multi-Level Pool Slicing for End-to-End Security Services in Optical Networks. *IEEE Transactions on Network Science and Engineering* **11**(2), 2153–2169 (2024) <https://doi.org/10.1109/TNSE.2023.3338734> . Accessed 2025-12-22
- [12] Kozlovics, S., Petrucena, K., Larins, D., Viksna, J.: Quantum Key Distribution as a Service and Its Injection into TLS. In: Meng, W., Yan, Z., Piuri, V. (eds.) *Information Security Practice and Experience* vol. 14341, pp. 527–545. Springer, Singapore (2023). https://doi.org/10.1007/978-981-99-7032-2_31 . Series Title: *Lecture Notes in Computer Science*. https://link.springer.com/10.1007/978-981-99-7032-2_31 Accessed 2026-02-12
- [13] Atutxa, A., Sanz, A., Salegi, E., Huarte, M., Astorga, J., Jacob, E.: Authentication of the QKD classical channel through Post-Quantum Cryptography in a multi-site 5G/6G quantum-safe communication network. In: 2025 International Conference on Quantum Communications, Networking, and Computing (QCNC), pp. 648–654. IEEE, Nara, Japan (2025). <https://doi.org/10.1109/QCNC64685.2025.00108> . <https://ieeexplore.ieee.org/document/11000183/> Accessed 2025-06-18
- [14] Tsili, A., Kordolaimis, K., Krilakis, K., Syvridis, D.: A scalable framework for post-quantum authentication in public key infrastructures. In: 2025 International Conference on Quantum Communications, Networking, and Computing (QCNC), pp. 279–286 (2025). <https://doi.org/10.1109/QCNC64685.2025.00052>
- [15] Martin, V., Brito, J.P., Ortíz, L., Méndez, R.B., Buruaga, J.S., Vicente, R.J., Sebastián-Lombrana, A., Rincón, D., Pérez, F., Sánchez, C., Peev, M., Brunner, H.H., Fung, F., Poppe, A., Fröwis, F., Shields, A.J., Woodward, R.I., Griesser, H., Roehrich, S., De La Iglesia, F., Abellán, C., Hentschel, M., Rivas-Moscato, J.M., Pastor-Perales, A., Figueira, J., López, D.: MadQCI: a heterogeneous and

scalable SDN-QKD network deployed in production facilities. *npj Quantum Information* **10**(1), 80 (2024) <https://doi.org/10.1038/s41534-024-00873-2> . Accessed 2025-08-20

- [16] Mendez, R.B., Buruaga, J.S., Vicente, R.J., Mengual, L., Pastor, A., Muñiz, A., Morales, J., Canto, R., Folgueira, J., Lopez, D.R., Martin, V., Brito, J.P.: SDN-Based Hybrid Quantum-Safe Domain Intercommunication Within MadQCI. In: 2024 International Conference on Quantum Communications, Networking, and Computing (QCNC), pp. 168–175. IEEE, Kanazawa, Japan (2024). <https://doi.org/10.1109/QCNC62729.2024.00034> . <https://ieeexplore.ieee.org/document/10628168/> Accessed 2026-02-12
- [17] Méndez, R.B., Brunner, H.H., Brito, J.P., Taramit, H., Fung, C.-H.F., Pastor, A., Cantó, R., Folgueira, J., López, D.R., Peev, M., Martin, V.: Switching Coordinator: An SDN Application for Flexible QKD Networks. *Entropy* **28**(2), 219 (2026) <https://doi.org/10.3390/e28020219> . Accessed 2026-02-17
- [18] Barral, D., Brazaola-Vicario, A., Cifrián, D., Costas, N., Blázquez, G., Fernández-Vilas, A., Llovo, I.F., Otero-García, P., Rejo, P.P., Ruiz, A., Villasuso, J., Fernández-Veiga, M.: Interconnecting Regional QKD Networks: Hybrid Key Delivery Across Quantum Domains. *arXiv. arXiv:2604.20376 [cs.NI]* (2026). <https://doi.org/10.48550/arXiv.2604.20376> . <http://arxiv.org/abs/2604.20376> Accessed 2026-05-15
- [19] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* **38**(2), 69–74 (2008)
- [20] Berde, P.e.a.: Onos: towards an open, distributed sdn os. In: *HotSDN*, pp. 1–6 (2014)
- [21] Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, vol. 175 (1984). Bangalore, India
- [22] Toshiba Europe Limited: Multiplexed QKD System MU. <https://www.toshiba.eu/quantum/products/quantum-key-distribution/>. Official product page describing the MU QKD system and its BB84-based protocol implementation (2025)
- [23] ID Quantique: Clavis XG Quantum Key Distribution System. <https://www.idquantique.com/quantum-safe-security/products/clavis-xg-qkd-system/>. Official product page for the Clavis XG commercial QKD system (2022)
- [24] ThinkQuantum S.r.l.: QUKY Quantum Key Distribution Platform. <https://www.thinkquantum.com/quky/>. Official product page describing the QUKY

- [25] European Telecommunications Standards Institute: Quantum key distribution (qkd); protocol and data format of rest-based key delivery api. Technical report, European Telecommunications Standards Institute (ETSI), Sophia Antipolis, France (2019). Defines a protocol and data format for REST-based key delivery in QKD networks. https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pdf
- [26] ETSI Industry Specification Group for Quantum Key Distribution (ISG QKD): Etsi gs qkd 015 v2.1.1 (2022-04): Quantum key distribution (qkd); control interface for software defined networks. Group Specification GS QKD 015 V2.1.1, ETSI (April 2022). Specifies management/abstraction interfaces between SDN controllers and QKD nodes for network control plane operations. https://www.etsi.org/deliver/etsi_gs/QKD/001_099/015/02.01.01_60/gs_QKD015v020101p.pdf
- [27] Hoffstein, J., Pipher, J., Silverman, J.H.: Ntru: A ring-based public key cryptosystem. In: Buhler, J.P. (ed.) Algorithmic Number Theory, Third International Symposium, ANTS III. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer, Berlin, Heidelberg (1998). <https://doi.org/10.1007/BFb0054868>

Supplementary information. The article has an accompanying supplementary file.

Declarations

Data availability

All data generated or analysed during this study are included in this published article

Competing interests

Authors A. T., K. K., A. M. and D. S. were employed by the company Eulambia Advanced Technologies.

Funding

This research was partially funded by the European Union through the projects ALLEGRO (Grant Agreement No. 101092766) and HellasQCI (Grant Agreement No. 101091504).

Authors' contributions

Konstantinos Krilakis was responsible for the system design and implementation. Antonia Tsili was responsible for the evaluation and analysis. Katerina Mandilara

contributed to the integration of the QKD nodes. Dimitris Syvridis was responsible for the conceptualization and coordination of the project.