Quantum Key Management System with Dynamic Routing for Meshed QKD Networks

Mario Wenning[1,2,*,†], Jonas Berl [1,3,†], Tobias Fehenberger[1], Ciarán Mullan[1], Helmut Grießer[1], Piotr Rydlichowski[4], Laurent Schmalen[3], Carmen Mas-Machuca[2,5]

[1] *Adva Network Security, Fraunhoferstrasse 9a, Munich, Germany*
[2] *Chair of Communication Networks, Technical University of Munich, Munich, Germany*
[3] *Communications Engineering Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany*
[4] *Poznań Supercomputing and Networking Center, Poznań, Poland*
[5] *Chair of Communication Networks, University of the Bundeswehr Munich, Munich, Germany*

[*] *mario.wenning@advasecurity.com* [†] *These authors contributed equally.*

Abstract: For an emulated QKD network, a decentralized key management system is automatically deployed as VNF. We show that dynamic key re-routing overcomes failures in the key distribution layer of meshed QKD-secured OTNs under realistic conditions.

1. Overview

One of the most mature applications of quantum technologies is quantum key distribution (QKD) for securing sensitive data against powerful quantum computers. Over the past decade, QKD vendors have developed devices for commercial use, and the first QKD networks (QKDNs) have been deployed in the field [1]. To operate long-haul meshed QKDNs, trusted nodes (TNs) are required to overcome the limited reach of current generation QKD devices [2]. However, with a TN-based network architecture comes the need for a quantum key management system (QKMS) that distributes end-to-end (e2e) keys. In contrast to QKMSs with a central controller [3], we propose a scalable decentralized approach that follows the paradigm of network function virtualization. Being distributed, our proposed demonstration does not suffer from a single point of failure and allows zero-touch provisioning using management and orchestration (MANO) tools.

Within our demonstration, we emulate the multi-layer network architecture depicted in Figure 1. The architecture is composed of three layers: (i) the QKDN, (ii) the key management network (KMN) and (iii) the optical transport network (OTN) with encrypted lightpaths (LPs). As shown in Figure 1, we simplify the QKDN by using single point-to-point (p2p) QKD links between OTN nodes. In a TN-based architecture, p2p QKD links distribute keys only between neighboring nodes. To establish e2e keys between any pair of OTN nodes, key relays between key management entities (KMEs) must be performed in the KMN. The relayed e2e keys are consumed by secure application entities (SAEs) that encrypt LPs in the OTN.

In the considered multi-layer architecture, the routes of LPs in the OTN and the routes of e2e keys may differ. The independence of both layers allows dynamic re-routing of e2e keys while leaving the LPs of the OTN untouched. Given sufficient path diversity, we demonstrate re-routing to overcome link failures or null secret key rates (SKRs) on the QKD layer, thereby improving network resiliency. To verify the operation of the proposed QKMS under realistic circumstances, we emulate p2p QKD links with recorded monitoring data from the QKD testbed deployed in Poznań. By introducing a KMN-wide accessible monitoring, we record performance data, e.g., routing metrics, buffer filling, and latencies.

The main contributions of our proposed demonstration are as follows:

- Automated deployment and zero-touch provisioning using state-of-the-art MANO tools
- Dynamic decentralized routing of e2e keys and interoperability with on-premise encryption devices
- Playback of recorded QKD monitoring data for testing under realistic conditions

2. Innovation

Decentralized routing is the de-facto standard in large IP-based networks, e.g., the Internet. Due to its proven scalability, we apply this paradigm to solve the e2e key routing problem. Additionally, a decentralized approach does not require a central controller which reduces the potential attack surface and prevents a single point of failure. For the application of QKD to meshed OTNs, automated deployments and virtual network functions (VNFs) simplify operation and setup. We implement the functionalities of e2e key routing and key relaying as VNFs utilizing containerization. With the help of Docker, we streamline its use, update procedures, and maintenance. Furthermore, the interfaces between QKD devices, KMEs, and SAEs that handle the exchange of secret keys

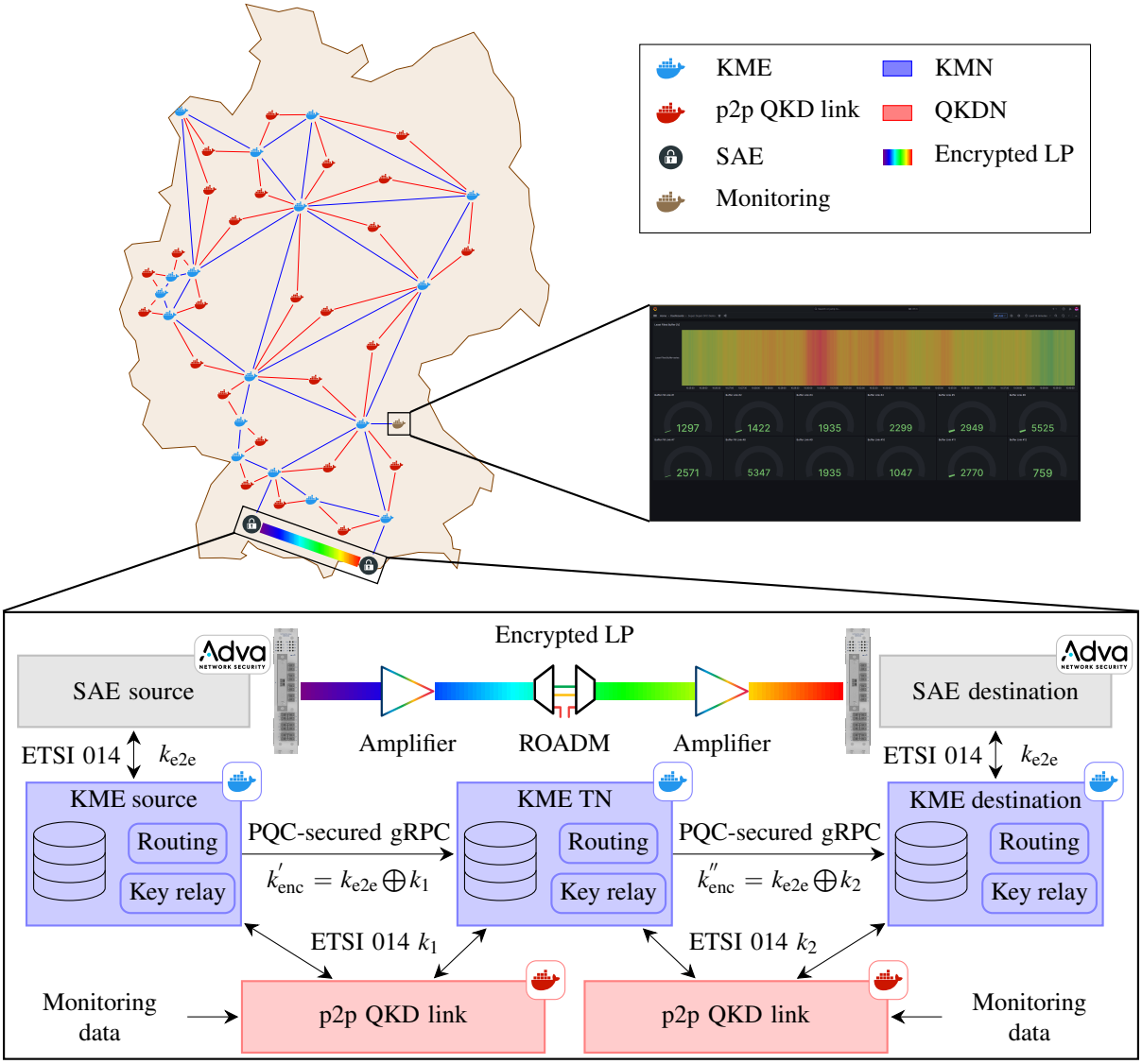

Fig. 1. Emulated multi-layer network based on the Nobel-DE topology with the demonstration setup.

follow industry standards. Hence, we support multi-vendor OTNs and QKDNs. Lastly, the key management traffic is secured by post-quantum cryptography (PQC). PQC adds another layer of security as it prevents meta-data, e.g., auxiliary routing information, from being leaked to an attacker.

3. OFC Relevance

Although quantum computers do not provide sufficient computational power yet to break state-of-the-art public-key cryptosystems, eavesdroppers can store data now and decrypt it later. Hence, sensitive data should be secured with quantum-safe techniques as soon as possible. In this demonstration, we show how a meshed QKD network can be integrated into existing OTN infrastructure and how secure e2e keys can be distributed with the help of a QKMS. Therefore, we outline a possible path towards QKD-secured optical communication in large meshed optical networks.

4. Objective and Configuration of the Demo

4.1. Point-to-point QKD link

After successful key distribution, both endpoints of a p2p QKD link can access shared key material. Therefore, a single container suffices to abstract p2p QKD links, as depicted in Figure 1. To verify our QKMS under realistic circumstances, we use SKRs from a deployed QKDN in Poznań. According to the recorded SKRs, we generate keys within the container and store them temporarily in an in-memory database. Additionally, the container comprises an ETSI GS QKD 014 server for exchanging keys with the KME [4]. To emulate a link failure between two

Fig. 2. Flowchart for the proposed demonstration.

QKD devices, e.g., due to an actual hardware failure or a null-SKR, we set the key generation rate for this p2p QKD link to zero.

4.2. Key Management Entity

As shown in Figure 1, every KME fulfills the following functionalities: fetching keys, routing, key relaying, and delivering keys. To support varying key delivery rates, we buffer e2e keys at the source and destination KMEs using an in-memory database. Implementing ETSI GS QKD 014 as northbound (NB) and southbound (SB) interfaces, we realize a vendor-agnostic KME that allows interoperability between different QKD and encryption devices [4]. The inter-KME communication builds on the gRPC framework [5] and is secured by transport layer security (TLS) with PQC support.

The source KME of an LP generates a random number, k_{e2e}, which is used as an e2e key after a successful key relay. After generating k_{e2e}, the source KME fetches the key k_1 that is shared with the next hop on the route and XORs both keys. The result of this operation, i.e., k'_{enc}, is sent to the next KME on the route. We refer to this operation as key relaying. Concatenating this principle along the route leads to a secure e2e key, assuming that all intermediate nodes are trusted [6]. The routing metric of each link depends on the fill state of the associated key buffer. Potential QKD-link failures or null-SKRs affect the generation of keys for this p2p QKD link and influence the routing metric. Automated re-routing will overcome this bottleneck by recalculating the current shortest path to the destination SAE according to the routing metric. The decentralized routing is based on the open shortest path first (OSPF) protocol [7].

4.3. Secure Application Entity

The relayed e2e keys are consumed by SAEs that encrypt LPs in the OTN. As depicted in Figure 1, the keys are exchanged using the ETSI GS QKD 014 interface between SAEs and KMEs [4]. Depending on the data rate of an LP, we renew keys at different rates.

5. Components and Demonstration Procedure

The proposed demonstration consists of the emulated multi-layer topology and on-premise Adva Network Security encryption devices. The on-premise devices implement the ETSI GS QKD 014 interface and use the obtained keys to encrypt LPs with the Advanced Encryption Standard. The key generation in the QKDN is emulated with the help of recorded monitoring data from a deployed QKDN.

Figure 2 summarizes the demonstration procedure. Using MANO tools, we automatically deploy the QKMS in the first step. Then, through key relaying and QKD, the key buffers begin to fill up. In the next step, we use the on-premise hardware to provision an encrypted LP that uses the relayed e2e keys. We monitor the resulting steady state to analyze the key routes and buffer fill states before we emulate a QKD failure. Lastly, we demonstrate automatic key re-routing until a new steady state is reached.

Acknowledgements

The work has been partially funded by the German Federal Ministry of Education and Research in the project QuNET+ISQKMS (16KISQ104). Carmen Mas-Machuca acknowledges the support by the German Research Foundation (DFG) under grant numbers MA 6529/4-1 and KE 1863/10-1.

References

1. H. Hübel et al., "Deployed QKD Networks in Europe," in *OFC*, 2023, pp. 1–3.
2. "ID Quantique Products," https://www.idquantique.com/quantum-safe-security/products/, [Online; On 31-10-2023].
3. R. Bassi et al., "Quantum Key Distribution with Trusted Relay using an ETSI-compliant Software-Defined Controller," in *DRCN*, 2023, pp. 1–7.
4. "Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API," https://www.etsi.org/deliver/etsi_gs/QKD/001_099/014/01.01.01_60/gs_qkd014v010101p.pdf, [Online; On 31-10-2023].
5. "gRPC A high performance, open source universal RPC framework," https://grpc.io/docs/guides/auth/, [Online; On 31-10-2023].
6. "ITU-T Y.3803, Quantum key distribution networks - Key management," https://www.itu.int/rec/T-REC-Y.3803-202012-I/en, [Online; On 31-10-2023].
7. J. Moy, "OSPF Version 2," https://www.rfc-editor.org/info/rfc2328, [Online; On 31-10-2023].