Secure Architecture for Quantum Key Distribution Networks

Bruno Huttner

ID Quantique, Rue Eugène-Marziano 25, CH-1227 Les Acacias – Genève; Switzerland

bruno.huttner@idquantique.com

Abstract: We present a new architecture, designed to improve security of QKD networks. The Trusted Nodes are divided into Core Nodes, which XOR the keys from various QKD devices, and Edge Nodes, connected to key users. © 2024 ID Quantique

1. Introduction

QKD networks (QKDN's) expand the range of QKD systems and create complex topologies for end-to-end secure key distribution [1],[2],[3]. They are built from a potentially large number of QKD links, connected at so-called Trusted Nodes (TN's). With current technology, these TN's convert the quantum signals into classical keys, which must be stored, combined and protected. Therefore, all secret keys transiting through the QKDN are available at these TN's. This will change in the future, when quantum repeaters, which allow quantum signals to be exchanged end-to-end, will be built [4]. However, today, security of the classical keys at the TN's and during their processing to create end-to-end keys, represent weak points in any extended QKDN.

In this work, we propose a QKDN architecture, which reduces these weaknesses and improves the security. In this architecture the nodes are divided into two sets of nodes, the Core Nodes, which are the distribution nodes, and the Edge Nodes, where the key users are located. In the Core Nodes, the keys received from the QKD layer, which we call QKD-keys, are processed to create what we call XORed-keys, made from a bit-by-bit XOR of pairs of QKD-keys. The XORed-keys are not secret and are exchanged over insecure channels to generate the end-to-end keys at the Edge Nodes. The Edge Nodes only contains the secret local keys.

2. Current Architecture of QKD Networks

Functionally, a QKDN is built as a number of different layers, as shown in Figure 1 below.

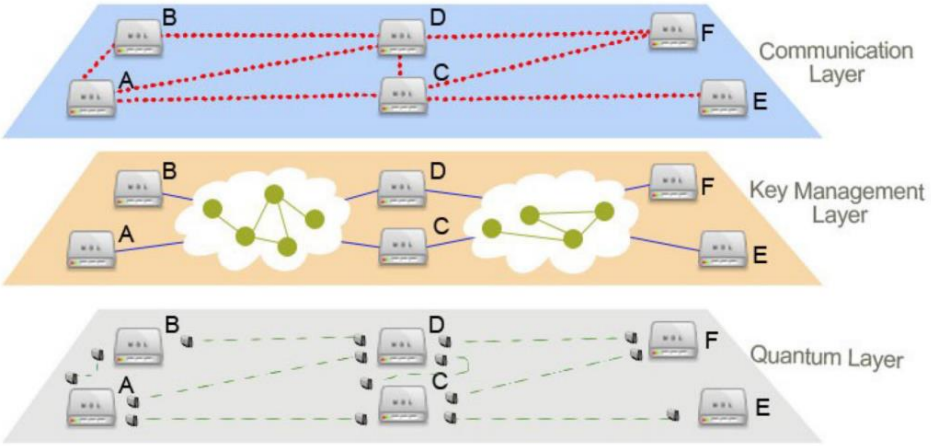

Figure 1: Schematic of a current QKDN, from reference [2]

It comprises:

- the Quantum Layer (QL), which includes the QKD modules and the TN's. The keys generated by the QKD modules are denoted by QKD-keys. They are sent up to the Key Management Layer (KML).
- the KML, which stores the QKD-keys, combines them to build the final keys, and distributes these final keys to the end users.
- and the Communication Layer, also referred to as the Service Layer (SL) or User Layer. It contains the applications or key users, which receive the final keys.

Similar structures have been proposed to different standardisation bodies and in the literature, with sometimes additional control layers [5],[6],[7]. However, the same basic layering remains, from Quantum Layer, up to Key Management Layer and finally to Service Layer.

In this architecture, the keys are transferred upwards locally, in the same node, from the QL to the KML and then to the Communication layer. The major attack vector is the KML, which needs to cover the complete QKDN, and is therefore delocalised over the whole network. A hacker penetrating the KML would get access to all the secret keys.

In our new architecture, the KML does not contain secret keys, but only processes the XORed-keys, made by the Core Nodes. A simple model is presented in the Section 3. The complete structure will be explained in Section 4.

3. The XOR model for key distribution

The XOR model was presented in [6] as an alternative key relay scheme for a single trunk line with extended range. We provide a simplified version as an example in Figure 2.

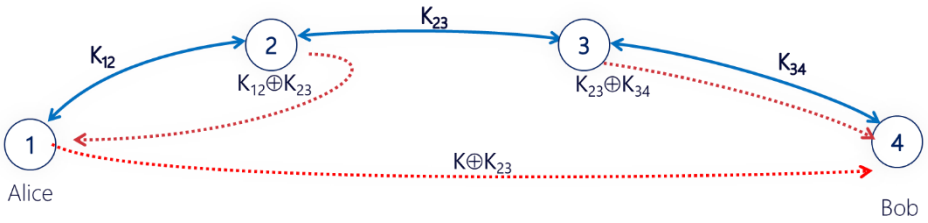

Figure 2: The XOR model of a simple QKDN

In this model, the Core Nodes are the intermediate nodes, 2 and 3. They calculate the XORed keys, here $K_{12} \oplus K_{23}$ and $K_{23} \oplus K_{34}$ respectively. When Alice, at Edge Node 1, and Bob, at Edge Node 4, request a common key, the XORed-keys are sent to them by the Core Nodes. In Figure 2 for example, node 2 sends the XORed-key $K_{12} \oplus K_{23}$ to Alice and node 3 sends $K_{23} \oplus K_{34}$ to Bob. Alice and Bob then share K_{23}. It is also possible to send all XORed-keys to one side, or to choose the middle point for longer links. The key provided by the network, here K_{23} ,can be used as shared key by Alice and Bob. Preferably, Alice generates her own key, K, generally by means of a Quantum Random Number Generator, encrypts it with an OTP with K_{23} and sends the resulting $K \oplus K_{23}$ it to Bob, who can decrypt it with K_{23}.

The main advantages of this model are already clear:

- There is no need for long-term storage of the QKD-keys in the intermediate nodes. The XORed-keys can be computed as soon as the QKD-keys are available. They can be stored at these nodes, but are not secret.
- The intermediate nodes do not receive the final key, as is the case in the hoping model, where the final key, K is sent from node to node. The final key K can be exchanged between Alice and Bob out-of-band, through an insecure channel.

In Section 4, we extend the model to more complex topologies and demonstrates its advantages.

4. The New Architecture

The new architecture, based on the XOR model is presented in Figure 3.

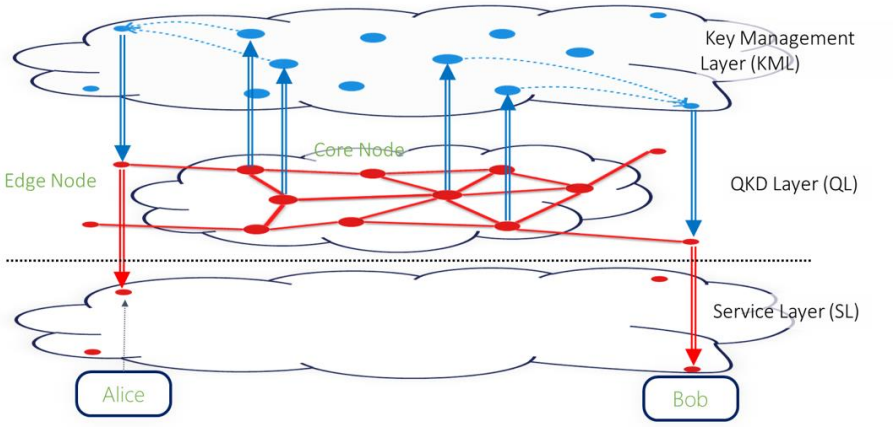

Figure 3: The new architecture, where the KML does not contains any secret key. The QKD layer (in red) comprises both Core Nodes (large red dots inside the cloud), connected by several QKD links (red lines) and Edge Nodes (small red dots outside the cloud). The blue double arrows represent XORed-keys. They are transferred upwards from the Core Nodes of the QL to the KML. When requested by the end-users, Alice and Bob, the KML sends them towards the Edge Nodes (dotted blue arrows). They are combined with the QKD-keys at the Edge Nodes. The final end-to-end keys, represented by red double arrows, are then sent down to the SL to Alice and Bob.

The QL contains all the QKD modules and generates the QKD-keys. In the Edge Nodes, which contains all the key users, the QKD-keys stay in the QL, except to be distributed locally to the users in the SL below. The Core Nodes comprise at least two QKD modules (this expands the concept of intermediate nodes of Section 2). The QKD-keys from the various links connected to the Core Nodes are XORed in pairs. The resulting XORed keys are sent to the KML above. The corresponding QKD-keys are then destroyed. Since the KML cannot request any QKD-key from the QL, an attacker, who manages to hack the KML and to control it, cannot get access to the QKD-keys. As an extra safety measure, one could add the requirement that the Core Node confirms the destruction of the QKD-keys before the corresponding XORed-key is used. The Edge Nodes are only linked to one Core Node. They receive the XORed-keys from the KML and build the end-to-end keys, which are provided to the local user in the SL below.

The main advantages of this new architecture are the following:

- The Core Nodes only receive QKD-keys from the local QKD devices. They forward XORed-keys to the KMS. The KMS cannot request any secret key from the Core Nodes. The Core Nodes can erase the QKD-keys before the end keys are created.
- The KML does not process any secret QKD-key. It only receives XORed-keys from the Core Nodes, processes them and forwards the relevant ones to the Edge Nodes. In principle, the KMS does not need confidentiality. In practice, one can add one layer of security by adding any conventional cryptography.
- The QKD-keys stored at the Edge Nodes are local. There is no intermediate keys available there. The final keys are only known to the corresponding Edge Nodes, which will use them.
- The SL comprises local end-users, directly connected to the Edge Nodes in a secure location. Obviously, physical security of the end users (Alice and Bob) must be provided.

These advantages are most relevant for large QKDN's, where different entities may control the Core Nodes. They are also important for applications of QKDN to critical infrastructures. Indeed, the end keys, which will be used by Alice and Bob, are made of three elements: one QKD-key stored at Alice; a combined XORed key from the KML, which does not need to be confidential; and one QKD-key stored at Bob. Since the QKD-keys are only available locally, any hacker penetrating the KML cannot create the correct final key, without attacking the end points as well.

References

[1] Miloslav Dusek, Norbert Lutkenhaus, and Martin Hendrych, "Quantum cryptography", Progress in Optics Vol. 49. Elsevier, 381–454. (2006) DOI: http://dx.doi.org/10.1016/S0079-6638(06)49005-3.

[2] Miralem Mehic, Marcin Niemiec, Stefan Rass, Jiajun Ma, Momtchil Peev, Alejandro Aguado, Vicente Martin, Stefan Schauer, Andreas Poppe, Christoph Pacher, Miroslav Voznak, "Quantum Key Distribution: A Networking Perspective", ACM Computing Surveys, Vol. 53, No. 5, Article 96. (2020) DOI: https://doi.org/10.1145/3402192 .

[3] Louis Salvail , Momtchil Peev , Eleni Diamanti, Romain Alléaume, Norbert Lütkenhaus, Thomas Länger, "Security of Trusted Repeater Quantum Key Distribution Networks", Journal of Computer Security, vol. 18, no. 1, pp. 61-87, (2010) DOI: http://dx.doi.org/10.3233/JCS-2010-0373 .

[4] R.Alléaume, C. Branciard, J.Boudad, T.Debuisschert, M.Dianatif, N.Gisin, M.Godfrey, P.Grangier, T.Länger, N.Lütkenhaus, C.Monyk, P.Painchault, M.Peevi, A.Poppe, T.Pornin, J.Rarity, R.Renner, G.Ribordy, M.Riguidel, L.Salvail, A.Shields, H.Weinfurter, A.Zeilinger, "Using quantum key distribution for cryptographic purposes: A survey", Theoretical Computer Science, Volume 560, Part 1, p.62-81 (2014); DOI: https://doi.org/10.1016/j.tcs.2014.09.018

[5] ITU-T Recommendation Y.3800 "Overview on networks supporting quantum key distribution"; https://www.itu.int/rec/T-REC-Y.3800-201910-I/en

[6] ITU-T Recommendation Y.3803 "Quantum key distribution networks - Key management"; https://www.itu.int/rec/T-REC-Y.3803/en

[7] ITU-T Recommendation X.1710 "Security framework for quantum key distribution networks"; https://www.itu.int/rec/T-REC-X.1710-202010-I