A Components Based Framework for Quantum Key Distribution Networks

V. Martin[1], D. Lopez[2], A. Aguado[1], J.P. Brito[1], J. Setien[1], P. Salas[1], C. Escribano[1], V. Lopez[2], A. Pastor-Perales[2] and M. Peev[3]

[1]*Center for Computational Simulation, Universidad Politécnica de Madrid, Madrid, Spain*
[2]*Telefonica Investigación y Desarrollo y gCTIO/I+D, Madrid, Spain*
[3]*Quantum Communications Lab. Huawei Research Germany, Dusseldorf, Germany*

ABSTRACT

The difficulties related to operating basic quantum communication technologies and the inherent limitations of their direct transmission distances, be these be a consequence of absorption or optical aperture, leads to the fact that current generations of Quantum Key Distribution (QKD) systems are essentially designed to work in a stand-alone mode on a single link. This limits their utility and the potential impact on the market. To avoid these barriers, it is necessary to advance towards systems built for a network. QKD systems have to perform as network devices, and ideally on an equal footing with other telecom devices, without requiring any specialized or ad hoc tuning. To achieve this "zero-touch" integration, it is necessary to build a software ecosystem. This software should take a bare QKD system and provide all the information, so that the network can manage and the applications use the system. Here we present a SW architecture, based on components with a well-defined functionality, to build the SW ecosystem needed to deploy QKD systems in networks. The components support traditional and Software Defined Networks (SDN), as well as separated or integrated (sharing infrastructure) networks to improve their industrialization.

Keywords: Quantum Key Distribution, Quantum Networks, Software-Defined Networking, Application Program Interface

1. INTRODUCTION

Although quantum communications is arguably the most mature among the new generation of quantum information technologies, it is built on extremely low yield processes, e.g.: the production of single photons using attenuated laser pulses are typically attenuated by 90% or more, practical detectors to be deployed in the field have an efficiency of around 20%, a good optical fibre halves the probability of transmission of a quantum signal every 15 km, etc. The difficulties of building Quantum Key Distribution (QKD) devices derived from these processes had as a consequence that the first generations were designed to work in a stand-alone mode, assuming a dedicated infrastructure just to support them. While this situation could be acceptable for certain, cost no object, situations, it is an obvious problem to create a broader market. This situation is slowly changing, with the design of systems and protocols that are more resilient to noise [1] and tolerate larger losses [2]. However, they are still built mostly to work on a single link as isolated devices that need a careful set-up and calibration procedure when they are installed and this, only in places meeting the most stringently advantageous requirements on losses or noise, for example. In short, the expectation is that the working environment has to be modified and fitted for the devices and not the other way around.

It is clear then that to improve the situation and make QKD useful to a broader market, the systems need to be designed for networks. Systems that work under situations that are more taxing on their performance, and that are better integrated in a more realistic network environment. The systems have to be aware of the network and the network has to be able to control them, ideally in an automatized way, such that optimal performance can be achieved within the technological and physical limits, always so close in quantum technologies.

The objective of this paper is to present a general networking framework to support this view. The framework is based on components that implement a well-defined functionality, this functionality defines the flow of information among the components to implement a given architecture. This flow, in turn, define the interfaces. Well-defined interfaces and clear component functionality allow for a modular view, which allows a disaggregated building of the network in the sense that different manufacturers can provide components that can work together. In the case of QKD, this is important because it opens the field to new players beyond the extremely specialized ones, brings needed know-how from other important fields, like security or networking, making possible a tighter integration and adding credibility through the use of tried and tested technologies. It is to be noted that a healthy ecosystem, where products can be bought from different manufacturers, is a must for the security market, where a user wants to make sure that there is always a backup source in case of trouble.

2. A GENERAL FRAMEWORK BASED ON COMPONENTS

There have been many QKD networks implemented up to now [3]. Starting from the DARPA in Boston (2004), Vienna (2008), Tokyo (2010) and more recently in China (2016-17) and Madrid (2018) to name a few of them. They have been built from single point to point quantum links and devices that many times were closer to laboratory prototypes than to commercial systems and have been implemented as a separated network, without sharing physical or logical infrastructure, with the telecommunications network. The Madrid network [4] tried to bring both together by using systems installed in production facilities of Telefónica, the main telco provider in Spain, and using Software Defined Networking as an integration driver to implement in the same logical structure the control of classical and quantum channels. The Huawei QKD systems used, allowed for a certain control from the network.

Although the published descriptions of these networks are usually done in terms of layers with different functionalities (quantum layer, key management layer, etc) and their definition is not very precise, allowing for a different number of layers, there are several common functionalities that can be extracted. Moreover, these can be grouped in different abstract components so that their complexity is large enough to be attractive as product implementations, which would allow for the disaggregation mentioned earlier. It is interesting to define here the quantum forwarding plane. This structure encapsulates all the capabilities that are added to a standard telecommunications network to make it quantum. To some extent is an attempt to clearly define, using a terminology from networks, the quantum layer seen in previous networks. A clear boundary between the quantum forwarding plane and the rest would allow to discern which parts are new and which ones can be essentially taken from networking or service-oriented technologies. In our definition, all the tasks that have to be done to obtain the final product of the quantum protocols belong to the QFP. In a QKD network, the final product are the symmetric secret keys, hence the QKD protocols, including all the manipulation and production of the quantum states, but also all the classical postprocessing and relay protocols, belong to the QFP. Note that the QFP can be easily extended to networks dedicated to entanglement distribution, where the final product is to have registers in distant locations in the network that store correlated quantum states. These components can be aggregated to create a QKD node as the basic building block of the network, which is shown in Fig. 1, where the QFP components are enclosed in the light red squares.

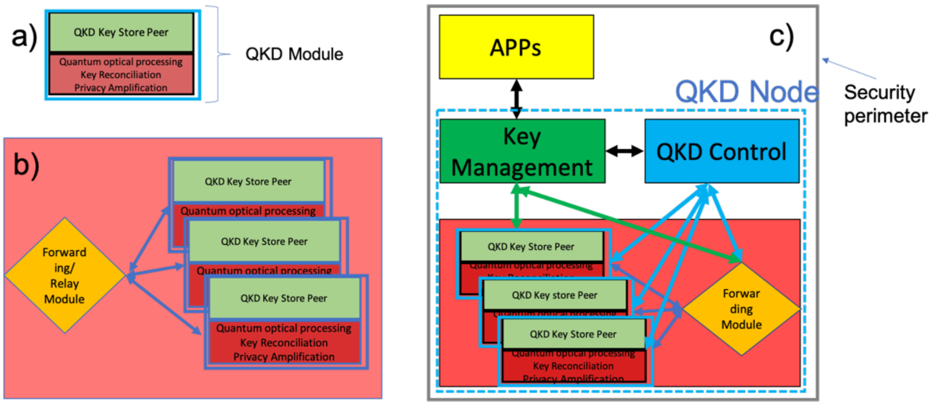

Figure 1: Set of components used to create a QKD network node. On the top left a) is the QKD Module, which contains the SW and HW needed for the execution of a QKD protocol. This includes the processing of quantum states and the results of their measurements that end in a secret symmetric key. This key is indexed and stored, ready to be delivered to a key management system (or an application) in the QKD key store. The components can be connected among them to relay the key to another node in a multi-hop link through the network. This relay function, depicted in the bottom left panel b) consumes keys only for this purpose, keys that are then not available for the applications. On the right, panel c), a full QKD network node is shown with a key manager and a control entity. The applications are not considered as part of the QKD node, although they have to be inside the same security perimeter. The arrows show the connections among the components, i.e. the interfaces among them.

The broad functionalities that have to be fulfilled by a node include the execution of the QKD protocol: all the processing of the quantum states, including their creation and measurement and the postprocessing of the classical information obtained after measurement. This results in symmetric keys that are tagged and stored in the local memory of the module, awaiting to be delivered. This conforms the QKD module, which is able to create a key between the two ends of the quantum channel, i.e. a single-hop link. Since the maximum distance in a single link is heavily penalized because of the absorptions in the optical fiber, the network has to be capable of doing multi-hop links, where the key is relayed from one node to the next in a chain of nodes. To do this, part of the key is consumed and this key is not available for the final application. This task is just key transport done on behalf of the network and is carried out by the forwarding (relay) module that has to be connected with all the QKD modules installed in the node. It is to be noted that in many of the QKD networks implemented, this key relay function is ascribed to the key management. While this is an implementation decision, the functional distinction has to be made, since key routing is certainly not considered a key management task. The key forwarding module is the smallest one considered and can be also used to build interoperability in the network among QKD module manufacturers at the link level (i.e. a multi-hop link can be composed from single hops, each one built by a different manufacturer).

The next component is the QKD control entity, whose functionality is to control the QKD devices and their connections -since we consider the general case, when optical switches allow the reconfiguration of the network- on behalf of the requests done by the application. Note that we do not consider that the application issues its requirements (e.g. demanding a given flow of key between two specific nodes in the network and with certain characteristics attached – Quality of Service) directly to the control, but they go through the key manager (KM), which is the last component. If the key manager cannot satisfy the requirements, it will ask for the connections to be created by the network control. This design means that there is a unique entry point of the application in the QKD node, which facilitates both the implementation of the applications and the collection of all the requests so that the key management can decide on the priorities and preemptively ask for the creation of keys among nodes to the QKD control that, in turn, can start the production of keys or demand from network control an optical path connecting the requested nodes. Note that while we assume only one KM per node (although it could be a simple one redirecting the requests to others, that might be proprietary and serve single-manufacturer sub-networks, for example) we do not place that request on QKD control, where we assume that each manufacturer could have its own to manage their QKD modules in, possibly, a proprietary way. However, the control communications among nodes should be standardized if some degree of interoperability among manufacturers is required (i.e. a network served by QKD modules from different manufacturers). In the case of a central controller, like in the SDN case, this is relatively easy to achieve by standardizing the South Bound Interface from the SDN central controller to the node controller (SDN Agent).

This simple set of components also satisfy the rule of implementing clearly distinct and non-overlapping functionalities that require a substantial know-how. The functionalities that are above the quantum forwarding plane, are essentially classical functionalities that can be considered extensions of components that are already in the market. For example, there are manufacturers of key management systems that would just need to add some specialized modules to their products to deal with the continuous supply of symmetric keys produced in a QKD network. In the same way, existing SDN controllers (e.g. OpenDaylight) can be extended with modules to deal with the control needs of a QKD network. From the very simple perspective of a network with a fixed set of exclusive use quantum channels to the more sophisticated one with a reconfigurable, shared infrastructure used to transmit both, quantum and classical signals and where finding the situations in which both types of signals can be transmitted simultaneously could require completely new modules, like path computational elements with algorithms that take into account all the variables -possibly including others associated to security, like avoiding certain nodes or requesting the production of a single key out of others that have travelled through different paths.

Building QKD networks using this component model have, thus, the opportunity to open the nowadays very limited QKD ecosystem to new players, which will also bring new capabilities, solid products and enlarge the market. However, to achieve this it is crucial to have well defined and stable interfaces among the components. In Fig. 1 the set of arrows define the points where communication among the components is needed, these are the interfaces that must be considered for a possible standardization. The ones in Fig. 1 correspond to the intra-node interfaces: those that connect components inside the same node, but inter-node flows of information, connecting several nodes have to be considered together with the associated interfaces.

Table 1: List of the inter-node and intra-node interfaces that appear naturally in our model with the selected components

Intra-node Interfaces	Inter-node interfaces
• App – Key manager interface • Key manager – QKD module interface • Key manager – Control interface • Control – QKD module interface • QKD Module – Forwarding Module interface • Control – Forwarding Module interface	Key management related node interfaces • Node KM to node KM interface • Node KM to Central KM interface [optional] Control Node interfaces: • Node KM to Network Control Interface [optional] • Node Control to network Control Interface • QKD Network Control peer interface (alternative to the one above) • QKD Network Control to QKD Network Control

In our model, the interfaces that appear naturally are listed in Table 1. When considering this list, several aspects have to be taken into account. First is that it is as general as possible: It tries to cover all network implementations done up to now and some foreseen ones. This means that several architectures are covered and that not all interfaces have to be present in the same network. For example, it does not make sense to have an interface to a centralized controller in a distributed control network. In other cases, an implementation makes one of the interfaces disappear because they coalesce several components. For example, in some network implementations the key manager also deals with the key routing (forwarding), making the interface to the forwarding module disappear, since its functionality becomes internal to the component. In our view, this is bad practice because key routing is not among the tasks considered part of the key management. The keys used for key forwarding are not available to the application, hence there is no need to supply them to a key manager and it is a general security rule to avoid unnecessary operations on secret key material. Lastly, ascribing this functionality to the KM would be equivalent to ask a hypothetical manufacturer specialized in key management to include key routing issues, which are completely foreign to the usual know-how of a KM company. This obviously could complicate the participation of the company in the QKD ecosystem.

To implement this model, it is then crucial to locate the interfaces that are critical for a given network implementation and make them stable, so that the developers feel confident in that their investment will pay off. This can be achieved through standardization. Interestingly, several of the interfaces listed in Table 1 have been already standardized in the European Telecommunications Standards Institute [5], in particular the key supply interfaces and also the control interface between the node control agent and a central controller in the SDN case.

3. CONCLUSIONS

In this contribution we have presented a model for Quantum Key Distribution networks that is based on components. The minimal set of components chosen implement functionalities that are clearly defined and distinct among components, then the information flow among them define naturally the interfaces. The model is broad enough to cope with all network implementations up to now, although not all implementations need all the interfaces. The functionality implemented by some of the components is actually classical and quite close to functionalities that are already implemented for network services and network infrastructures, mainly key management systems and network control and management products. These can be extended to cope with the products (secret keys) and needs of the new QKD networks (specialized control to optimize the transport of quantum signals and maximize the flow of secret keys in the network), which potentially opens the QKD ecosystem to new players. This would bring benefits to QKD networks in several aspects: more confidence in the products coming from established players in the security and networking arena, a varied ecosystem that is not tied to a few manufacturers and the increased competition that is usually so beneficial for the customers. To achieve this program, good and stable interfaces have to be produced, a clear role for standardization. Stable and clear standards together with research prototypes showing that the scheme works, would be good drivers to broaden the QKD market.

ACKNOWLEDGEMENTS

Authors would like to thank the Madrid's regional government, Comunidad Autonoma de Madrid, for the project Quantum Information Technologies Madrid, QUITEMAD-CM P2018/TCS-4342, the FET Flagship on Quantum Technologies, European Union Horizon 2020 research and innovation programme under grant agreement No 820466: Continuous Variable Quantum Communications (CiViQ) and ICT grant agreement No 857156: Open European Quantum Key Distribution Testbed (OpenQKD) and the team of Transport and IP Connectivity in Telefónica Spain for their support to this activity.

REFERENCES

[1] F. Karinou et al. "Towards the integration of CV quantum key distribution in deployed optical networks," IEEE Photonics Technology Letters, vol. 30, no. 7, pp. 650–653, April 2018.

[2] M. Lucamarini et al. "Overcoming the rate-distance barrier of quantum key distribution without using quantum repeaters" Nature 557, 400-403 (2018)

[3] M. Mehic et al. "Quantum Key Distribution: A Networking Perspective" ACM Computing Surveys, to be published (2020)

[4] A. Aguado et al. "The Engineering of a SDN Quantum Key Distribution Network", IEEE Comms. Mag. 57, 20-26 (2019) "The Future of Internet" doi:10.1109/MCOM.2019.1800763; arxiv.org/abs/1907.00174

[5] ETSI GS QKD 004 "QKD Application Interface", 014 "QKD Application Delivery" and 015 "QKD Control Interface for Software Defined Networks (draft)" See portal.etsi.org