Secure NFV Orchestration over an SDN-Controlled Optical Network with Time-Shared Quantum Key Distribution Resources

A. Aguado, E. Hugues-Salas, P. A. Haigh, J. Marhuenda, A. B. Price, P. Sibson, J. E. Kennard, C. Erven, J. G. Rarity, M. G. Thompson, A. Lord, R. Nejabati and D. Simeonidou

Abstract— **Quantum key distribution (QKD) is a state-of-the-art method of generating cryptographic keys by exchanging single photons. Measurements on the photons are constrained by the laws of quantum mechanics, and it is from this that the keys derive their security. Current public key encryption relies on mathematical problems that cannot be solved efficiently using present-day technologies; however, it is vulnerable to computational advances. In contrast QKD generates truly random keys secured against computational advances and more general attacks when implemented properly. On the other hand, networks are moving towards a process of softwarization with the main objective to reduce cost in both, the deployment and in the network maintenance. This process replaces traditional network functionalities (or even full network instances) typically performed in network devices to be located as software distributed across commodity data centers. Within this context, network function virtualization (NFV) is a new concept in which operations of current proprietary hardware appliances are decoupled and run as software instances. However, the security of NFV still needs to be addressed prior to deployment in the real world. In particular, virtual network function (VNF) distribution across data centers is a risk for network operators, as an eavesdropper could compromise not just virtualized services, but the whole infrastructure.**

We demonstrate, for the first time, a secure architectural solution for VNF distribution, combining NFV orchestration and QKD technology by scheduling an optical network using SDN. A time-shared approach is designed and presented as a cost-effective solution for practical deployment, showing the performance of different quantum links in a distributed environment.

Index Terms—**Network Functions Virtualization, Quantum Key Distribution, Software Defined Networking.**

Manuscript submitted October 28, 2016. This work has been supported by the UK government under the EPSRC grants EP/M013472/1: UK Quantum Technology Hub for Quantum Communications Technologies and EP/L020009/1: Towards Ultimate Convergence of All Networks.

A. Aguado was with the High Performance Networks group, Faculty of Engineering, University of Bristol, Bristol, BS8 1UB, UK and is now with the Center for Computational Simulation - UPM. Campus de Montegancedo. Boadilla del Monte, 28660 Madrid. Spain (a.aguadom@alumnos.upm.es)

E. Hugues-Salas, J. Marhuenda, R. Nejabati and D. Simeonidou are with the High Performance Networks Group, Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1TH, U.K. (e-mail: {e.huguessalas; jaume.marhuenda; reza.nejabati; dimitra.simeonidou} @bristol.ac.uk).

P. A. Haigh was with the High Performance Networks group, Faculty of Engineering, University of Bristol, Bristol, BS8 1UB, UK and is now with the Communications and Information Systems Group, University College London, London, WC1E 6BT, UK (p.haigh@ucl.ac.uk)

A. B. Price, P. Sibson, J. E. Kennard, C. Erven, J. G. Rarity and M. G. Thompson are with the Centre for Quantum Photonics, School of Physics, University of Bristol, Bristol BS8 1TH, U.K. (e-mail: {alasdair.price; philip.sibson; jake.kennard; chris.erven; john.rarity; mark.thompson} @bristol.ac.uk). A. B. Price is also with Quantum Engineering Centre for Doctoral Training, University of Bristol, Bristol BS8 1TH, U.K.

A. Lord is with BT, Polaris House, Martlesham Heath, Ipswich IP5 3RE, UK (e-mail: andrew.lord@bt.com).

I. Introduction

NETWORK function virtualization (NFV) promises significant network infrastructure simplification as current hardware appliances are replaced with software running on standard servers. NFV is complemented by software-defined networking (SDN), provisioning the required network connectivity to respond to newly instantiated appliances by aligning heterogeneous network topologies in an automated manner. In addition, NFV provides an on-demand instantiation of the required network functions without installation of any new equipment and distributes the functions via the fastest network service provisioning possible. However, as in other network softwarization processes, there are security risks associated in the deployment of an NFV solution. In an NFV-enabled network infrastructure, network functions are stored centrally as software images in a remote data center (DC) where they can be cloned, transferred and deployed as virtual functions on commodity servers (replacing network appliances) across the network. This transfer of network functions must be secured, as any attempt to tamper with NFV can create a significant security breach. For instance, if a transmitted software image of a network function contains any sensitive information, such as a firewall, its interception and/or alteration can compromise an entire network.

A leading candidate for protecting against this type of security threat is quantum key distribution (QKD), a contemporary approach to the generation of symmetric keys [1]. In QKD, keys are distributed by transmitting single photons from a sender (QKD-Alice) to a receiver (QKD-Bob) over a quantum channel. This channel can be implemented using fibre optics or in free space. Fundamental laws of physics prevent an eavesdropper (Eve) from learning the key, as any attempt made to gain information about the photons will irreversibly change them in a manner that can be readily detected. An additional benefit of QKD is that the keys it generates can be considered future-proof from hacking. As

they are truly random, no information can be gained from a mathematical attack, regardless of any computational advances.

Building on these principles, we propose and experimentally demonstrate the inclusion of QKD to tackle NFV's security problems. Utilizing SDN technology, a cost-efficient method for time-sharing the QKD systems is presented, demonstrating the ease of which these systems can be integrated with an NFV platform. This paper extends the description presented in [2] with a more thorough description of the experimental work and the scheduling process. Moreover, this paper enhances [2] by adding trusted node functionality for longer distances.

The paper is organized as follows: section II provides a short introduction to the NFV MANO framework, shows our architectural approach and introduces the integration with QKD; section III focuses on how SDN can reduced costs in a QKD network by time-sharing the available resources, as well as demonstrating a trusted node architecture; section IV describes the experimental test-bed; section V presents our experimental results and section VI concludes the paper.

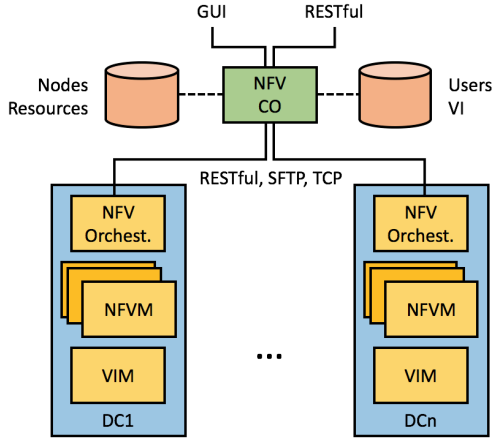

Fig 1 NFV MANO architecture for distributed DCs

II. NFV MANO Architecture

The NFV management and orchestration (MANO) specification document has been developed within the European Telecommunications Standards Institute (ETSI) [3]. The architecture is organized into three layers: the orchestrator, virtual network function (VNF) managers, and the virtual infrastructure manager (VIM). Existing solutions approach the functional distribution in several different ways and/or are not complete (they provide implementations of different functional modules), due to a lack of standardization.

In this paper, we define and implement a prototype of the ETSI NFV architecture for distributed DCs (Fig. 1) within an emulated metro area network (MAN). Our approach splits the architecture into a centralized orchestrator (NFV CO) acting as a master node and different ETSI NFV stacks operating in slave mode for each data center. The NFV CO is composed of a Python backend core, mySQL database maintaining information regarding users, virtual infrastructures and distributed nodes, a graphical user interface (GUI) and representational state transfer (RESTful) interfaces to facilitate platform users and administrators to manage their infrastructures. Network function images are stored within the centralized orchestrator DC, which is responsible for their distribution based on service demands. When a service/infrastructure request arrives at the orchestrator, the service chain is split into network requirements (that will be pushed into an SDN controller) along with network nodes and endpoints (distributed among the DCs).

The slave-mode stack implements the MANO architecture composed of three layers, as previously mentioned:

- The orchestrator (NFV MANO gateway) acts as a gateway between the NFV CO and the DC, taking care of importing and, when required, decrypting the VNF images, orchestrating the different endpoints within a data center.
- The VNF managers provide lifecycle management and monitor VNFs to control startup, scaling and termination of the functions (not implemented on the first prototype).
- The VIM manages the resources to create VNF instances (VNFI), orchestrating the virtual machines (VMs) and Linux containers (based on OpenStack and Docker, respectively).

The current implementation interfaces the different modules of the architecture using RESTful APIs to send instructions. For file transmission it uses the secure file transfer protocol (SFTP) or the transmission control protocol (TCP), both of which are tested. Regarding the network configuration, the CO to SDN controller communicates through an SSH-based interface, following the policy based approach shown in [4].

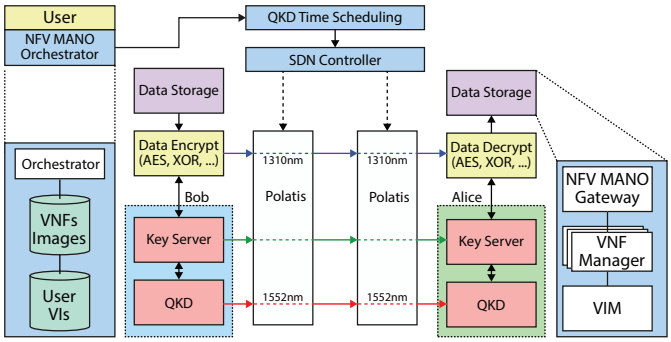

Fig 2 Point to point link integration of our distributed NFV MANO architecture and the QKD system

A. Integration with QKD

In the proposed NFV architecture, the NFV CO holds a catalogue of the network functions hosted in its trusted DC. To make use of these virtualized network functions, the images containing them must be cloned and securely transmitted to remote data centers. Conventionally, secure channels are established using public key cryptography, which relies on the presumed computational difficulty of solving certain mathematical problems. Although the public keys do not always encrypt the data directly, they can be used for symmetric key transport (e.g. key exchange using Diffie-Hellman). To evolve our NFV MANO architecture, we have integrated commercial ID Quantique QKD systems (ID3100 Clavis^2) [5] as shown in Fig. 2. The network that connects the

QKD pair comprises a 25ms-switching, large-port count, programmable beam-steering optical switch (Polatis™), logically partitioned into two smaller interconnected switches to enable multiple link emulation. The optical network and the switch are used for directing both standard and quantum signals between the nodes.

When an image needs to be transmitted, the NFV CO asks for a key from the QKD-Bob using the proprietary IDQ3P protocol (based on the Q3P specification [6]), before the Advanced Encryption Standard (AES) symmetric key algorithm is used, together with key, to encrypt the image, informing the remote platform of the transmission and the key ID needed for image decryption. While we use the standard AES algorithm in this experiment, the one-time pad (OTP) algorithm could easily be used instead for mission critical transmissions (e.g. small size configuration files or management commands.). The workflow is shown in fig. 3.

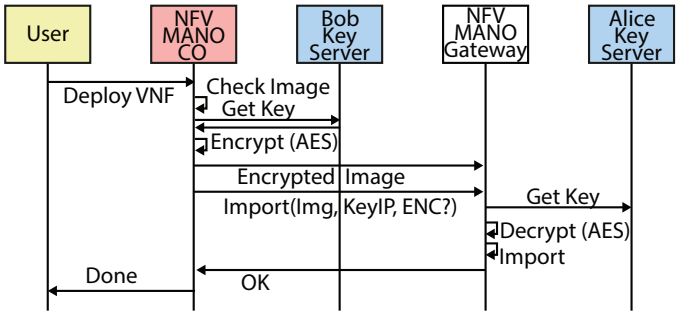

Fig 3 Workflow for the end-to-end distribution of VNFs

III. Time-shared QKD network based on SDN

QKD networks have previously been demonstrated in field experiments [7-8]. However, the technology deployed is yet to reach a stage where it can be mass produced at low cost. Losses and channel noise also present deployment limitations for QKD. Current approaches for quantum networks focus on the deployment of point-to-point links, where a pair of QKD devices is required for each connection. In order to reduce the financial expenditure of deploying a QKD network, we propose a time-scheduled solution. Based on the specific requirements of our architecture we can reduce the number of physical devices required by time-sharing QKD resources.

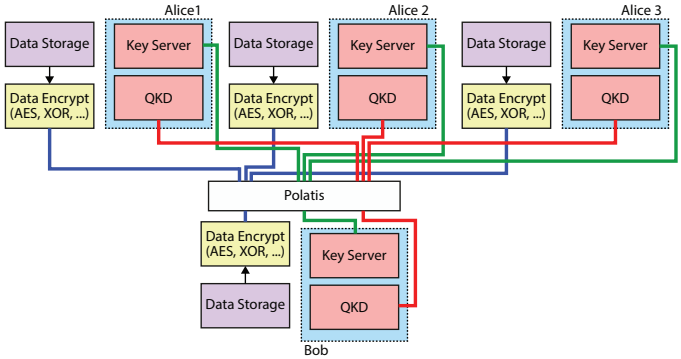

Fig 4 Logical representation of the timeshared QKD network

Fig. 2 shows our novel QKD-enabled NFV architecture in an SDN controlled optical network, while Fig. 4 shows the one-to-many QKD network approach. Our implementation of the network scheduler enables a single physical QKD-Bob to be time-shared between multiple endpoints (QKD-Alice) allowing the establishment of multiple secure connections using fewer QKD devices than would normally be required. This method reduces the cost of the proposed secure NFV architecture because of the significant reduction in hardware and, for improved systems that utilize plug-and-play QKD devices, allows the most sophisticated components to be contained within a single time-shared unit.

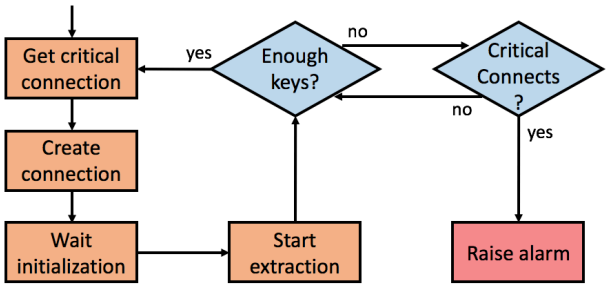

Fig 5 Scheduler's workflow for timesharing the QKD network.

In addition, the optical connections can be reconfigured quickly using SDN, also integrated with the scheduler. The scheduler synchronizes with the QKD units allocated in the NFV stacks and the NFV CO domains. This scheduling is undertaken assuming that the QKD channel and the classical data channels are multiplexed in space (i.e. carried over different fibers). The scheduling process can be summarized as follows:

1) The scheduler identifies the QKD-Alice nodes that need to share a particular QKD-Bob node (information provided by the NFV CO).

2) The scheduling mechanism utilizes information extracted from the network SDN controller to create the required end-to-end connectivity for the QKD-Alice/Bob pairs. It establishes optical connections between the QKD-Bob and every QKD-Alice, sequentially one connection at a time.

3) The key servers are started and put on hold until the QKD initialization process has finished.

4) the scheduler extracts the generated keys from both QKD units, storing them alongside the key IDs extracted from the response message.

The scheduler follows the flow chart shown in Fig. 5. It analyses which connections are most critical based on the average of requested bits per second and current number of keys available, and tries to keep a balance between the connections. The scheduler keeps important information of each link: average of requests per second, number of available keys, initialization times and the key generation speed. It is important to notice that the scheduler is subject to a number of constraints when planning the switching:

1) At the start, it has to consider which connection is most critical, i.e. which end-to-end connection will require key extraction soonest (the lower time "t" where "t" is equal to the current number of keys in bits divided by the average number of requests for a given connection in bits/second).

2) Once the connection has been stablished and the key extraction process initiated, the scheduler checks whether the number of keys already extracted are enough to cope with the number of requests for a given time (as an example, we

consider twice the initialization time of each of the remaining connections).

3) While one connection is active, the scheduler monitors if the others are in a critical situation by: a) sorting the connections from lowest to highest priority, b) for the nth connection in the list, check if it can remain live for time T, where:

$$T = \sum_{i=0}^{n-1} 2t_i + t_n$$

where t_i is the initialization time for a given connection "i".

If step 3) finds connections in a critical situation and condition 2) is satisfied, the scheduler will change the current configuration. If during these steps some conditions are not satisfied (not enough keys to cope with all the requests in the links), the scheduler should raise an alarm to a network management system either to increase the number of devices or to dynamically activate existing ones to cope with the demand for keys. To avoid generating an excess of keys for a single connection while others are consuming, we have set up an adaptive threshold to stop the current active connection. This threshold is calculated with the remaining live time of the other connections. Some of the values (e.g. time considered for a connection to be alive) have been hardcoded for this test. In order to calculate more accurate values, the scheduler should perform data analytics over previously monitored data. The scheduling process works offline, so that at any time the NFV CO requires keys to encrypt an image, there will be keys available for any combination of endpoints.

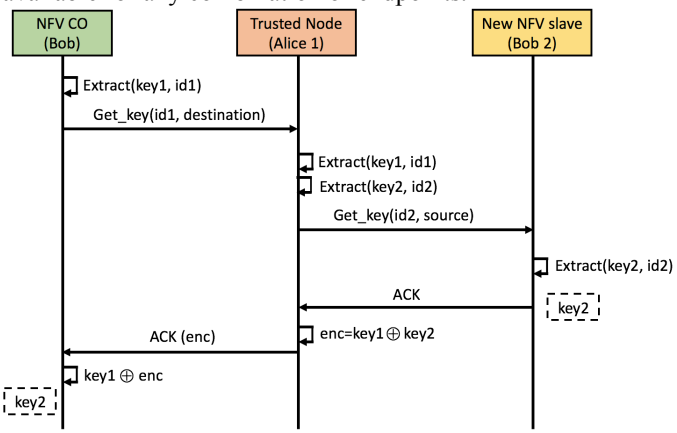

Fig 6 Example of workflow for multi-hop key distribution (single trusted node)

A. Key distribution in a multi-hop QKD network.

While it may provide much stronger security, QKD still has its limitations. Current state-of-the-art demonstrations can reach up to 307 kilometers [9], although the secret key rate drops significantly. If the separation between nodes is large, a multi-hop approach (conventionally referred to as trusted nodes) will be required, utilizing one or more intermediate trusted nodes. These nodes have to manage keys associated with different end points, while maintaining the secrecy of the keys and retaining an awareness about certain pieces of topological information (neighboring nodes, that may also be used for routing requests).

We have logically extended the topology presented in Fig. 4 adding a fourth node (NFV stack). In order to populate a key from the new domain to the NFV CO, the intermediate node (Alice 1) has to coordinate both endpoints, extract two different keys (one per connection), and finally use one to encrypt the other before transmitting it to the appropriate Bob. More explicitly, the workflow is as shown in Fig. 6:

1) The NFV CO (Bob) extracts a key and key ID from the QKD boxes (in our case, from the scheduler)

2) The NFV CO sends a message to the trusted node, including the key ID and the destination ID to be reached (in this example, an IP).

3) The trusted node (Alice 1) extracts the key for the first connection using the first key ID.

4) The trusted node extracts a key and key ID for the second connection from the scheduler.

5) The trusted node sends a message including the key ID of the second connection, as well as the ID of the source node.

6) The destination node (new NFV slave, Bob 2) extracts the key, pairing it with the source ID, and it returns an ACK when successfully finished.

7) The trusted node performs an XOR operation with key1 and key2 to encrypt key2 and sends the result to the NFV CO.

8) The NFV CO performs an XOR with the encrypted key and key1, obtaining key2.

Note that this is just one example of how to propagate keys in a trusted node QKD network (the trusted node could have sent *key1* to the destination node or both endpoints could have the other endpoint key). By carrying out this set of operations, the NFV CO and the new deployed NFV slave can successfully start encrypting their communications.

IV. Experimental test-bed

Our time-shared QKD test-bed logically comprises four nodes, as shown in Fig. 4. Node 1 contains a QKD-Bob device and a key server. These are time shared with three emulated QKD-Alice devices (Nodes 2 to 4) and their key servers. In each node, keys can be extracted for use in a number of different protocols, such as AES or OTP. The QKD-Alice/Bob nodes are connected via a dynamic, and reconfigurable circuit-switched optical network. The QKD-Alices are emulated in a way such that the same physical device is used, but the network path changes for each emulated node. Note, for the purposes of this demonstration, an initial step changing the authentication keys between QKD devices was omitted and will be implemented in future work. Our optical network is composed by the abovementioned SDN-enabled Polatis switch, emulating the different logical nodes. The switch is loopback connected using several fibre lengths to emulate the different links. The QKD-Alice/Bob units, operating at 1551.7 nm, are controlled by separate servers using a software suite designed for automated hardware operation and key distillation. In this experiment, the QKD and classical data channels are always confined to separate fibers. This test-bed is equipped with two Dell T630 servers running the proposed NFV architecture stack, the SDN controller for the optical network (OpenDaylight) and the QKD device scheduling mechanism.

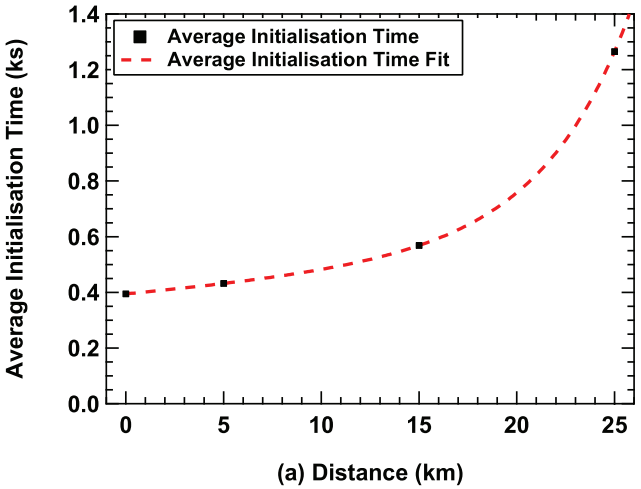

Fig 7 Average of initialization time for each link

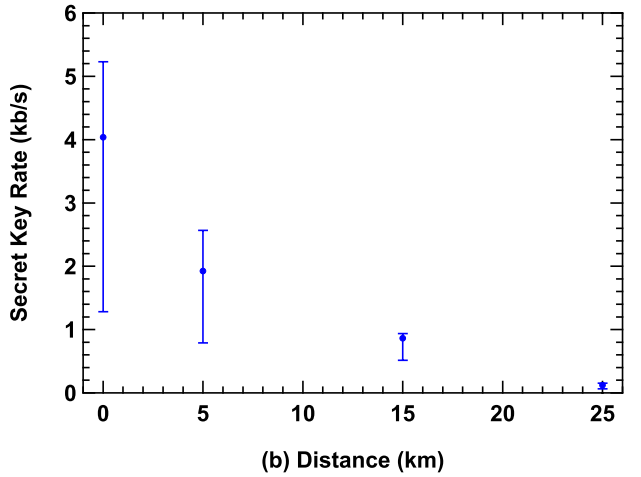

Fig 8 Measured key rates for the different links

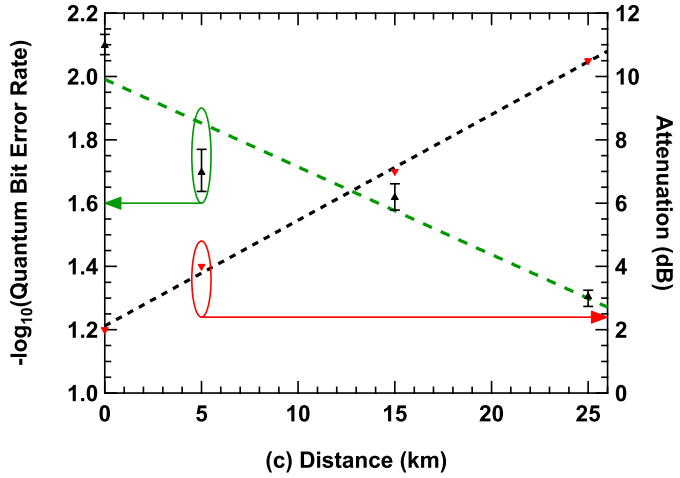

Fig 9 Measure QBER and attenuation for the different links

V. Experimental Results

The results from our time-shared QKD network are shown in Fig. 7-9. Fig. 7 shows the measured average initialization time for each QKD-Alice/Bob pairs, i.e. the time taken for synchronization, characterization of the setup and generation of the first set of keys. The measured average initialization time increases with the distance between QKD-Alice/Bob. Compared to the back-to-back (BtB) case, where the average initialization time is 400s, a threefold increase to 1265s at 25 km is observed, indicating that as the distance (and power penalty) increases, the time taken to initialize becomes substantially larger. This result is particularly useful when time-sharing the QKD-Bob, as it allows a sensible estimate of the delays encountered when cyclically communicating between different QKD-Alice units. Fig. 8 shows the secret key rate as a function of distance. The mean values are represented by the circular marker, while the maximum and minimum values are shown as error bars. The secret key rate in the BtB scenario is ~4 kb/s, while in the 25km case the rate drops to ~0.1 kb/s, due to the additional attenuation. These values slightly differ from the ones exposed in the Clavis2 specification file (higher rates than 0.5 kbps over 25 km). This is due to additional attenuation introduced in the link by the two cross connections (between 1 and 2 dB loss per cross connection) plus small losses in each connector (around 0.5 dB). In optimal conditions, a 25 km SMF has losses between 5 and 6 dB. In our case, we measured a total loss above 10dB, which reduces the measured rate. These factors clearly reduce the optimal performance of the optical link. Fig. 9 shows the quantum bit error rate (QBER). In line with the other results shown, the longer the distance between QKD–Alice/Bob nodes the higher the QBER, affecting the secret key rate with linear proportionality. A QBER of 5.3% is reached for a QKD pair of over 25km of standard single-mode fibre (SSMF). The linear fit in Fig. 9 (green) is calculated using the measured data with a $-\log_{10}$(QBER) to show the exponential relation between QBER and distance (and, similarly, attenuation).

For the multi-hop setup, we emulated two links using the same pair of devices, having the same distance for the two hops (5 km each). In this case, the workflow was programmed in Python, running the three nodes in three different servers spread across the lab. The interfaces used for this use case were IDQ3P for key extraction and HTTP for communication among the nodes. The HTTP response from the intermediate node (ACK(enc) in Fig. 6) contains the encrypted key (key1 ⊕ key2) as raw data, which is used by the NFV CO to extract key2 using key1. An example of this execution for a 32 byte key exchange is:

Key1:0x063c394625fb316d899e849265ca04b25fbd3eaf12cf51a48871f9c7a548ae40
Enc:0x07098e88af3dcc5f0c558d0b751028553031c6e613146196ae7b11cb4cb43bc9
Key2:0x0135b7ce8ac6fd3285cb099910da2ce76f8cf84901db3032260ae80ce9fc9589

Fig. 10 shows the encrypted key (data) transmitted in the HTTP response.

```
▼ Data (32 bytes)
    Data: 07098e88af3dcc5f0c558d0b751028553031c6e613
    [Length: 32]

0000  07 09 8e 88 af 3d cc 5f  0c 55 8d 0b 75 10 28 55
0010  30 31 c6 e6 13 14 61 96  ae 7b 11 cb 4c b4 3b c9
```

Fig 10 Encrypted key transmitted from the trusted node to the NFV CO node

Regarding image transmission between the nodes, our setup was composed of three Dell PowerEdge T630 servers (one hosting the orchestrator, one hosting the ETSI NFV stack in slave mode and one hosting the SDN controller and scheduling mechanism), with 10Gb/s small form-factor pluggable interfaces allowing the servers to communicate through our optical network at 1310nm. In here, any wavelength can be used since the data and quantum channels are carried over separate fibres (Fig. 4).

	Windows VM	Ubuntu VM	Centos OVS_LC
Size	15925 MB	191 MB	769 MB
Encryption	02:06.870	00:01.71	00:05.730
Decryption	02:24.970	00:01.57	00:05.890
SFTP	06:45.060	00:04.42	00:17:930
Socket	00:33.626	00:00.39	00:01.260

Fig 11 Different times for image encryption, transmission and decryption

As shown in Fig. 11, in our implementation the transmission of the 16GB Windows Server image including encryption (126s), transmission through a standard socket (33s) and decryption (144s) took ~305s. At the same time, using secure file transfer protocol (SFTP) for the transmission takes ~405s. Note, these timings are intended to be illustrative as they vary depending on the computational load of the server and the specific implementation details of sockets and SFTP.

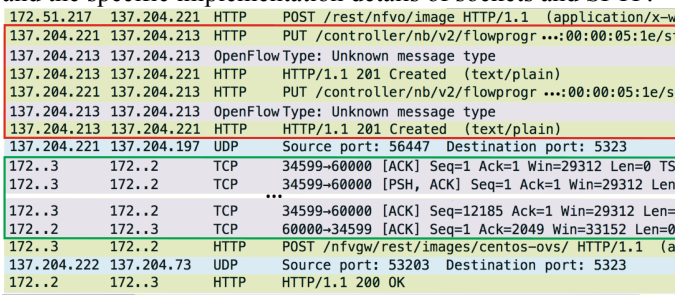

Fig 12 Wireshark capture showing the chain of messages

Fig. 12 shows the capture of messages exchanged for the VNF transmission to the remote DC, as shown previously in Fig. 3. The first message initiates the workflow, before the orchestrator creates the required network connectivity for the transmission by sending CFlow_mod OpenFlow messages to the Polatis switch through OpenDaylight (highlighted in red). The orchestrator requires the key from the QKD-Bob unit to encrypt the image and sends it to the remote DC by using a standard TCP socket (highlighted in green). We also performed this transmission using SFTP. To finalize the process, the orchestrator informs the image and key IDs to be used for the decryption, sending an acknowledgement ACK (HTTP 200) when the process is finished.

VI. Conclusions

In this paper, the integration of an NFV orchestration platform over SDN-controlled optical networks with quantum-key-distribution systems is shown for the first time. A particular ETSI-NFV inter-DC architecture has been designed on a QKD compatible optical network test-bed to provide enhanced security capabilities for VNF distribution across DCs. A novel SDN-based resource scheduling method for time-sharing a single QKD-Bob between multiple QKD-Alice units. Furthermore, we show that our solution could be applied in longer distances by applying a multi-hop approach to distribute keys on the network. Results demonstrate that an SSMF link of up to 25 km can be secured using quantum encryption for NFV MANO operations with a simultaneous SDN control, showing a minimum QBER of 5.3%.

References

[1] H.K Lo, M Curty, and K Tamaki, “Secure quantum key distribution” Nature Photonics 8, 595-604 (2014).

[2] A. Aguado *et al.* “First Experimental Demonstration of Secure NFV Orchestration over an SDN-Controlled Optical Network with Time-Shared Quantum Key Distribution Resources” in European Conference in Optical Communications W.1.F, Düsseldorf, Germany 2016.

[3] ETSI GS NFV 002 V1.2.1 “Network Functions Virtualisation (NFV); Architectural Framework” (2014-12).

[4] J. Marhuenda *et al.* “Experimental Demonstration of Policy-based Dynamic End-to- End Provisioning over Multi-Layer Network using SDN”; in European Conference in Optical Communications Tu.1.B, Düsseldorf, Germany 2016.

[5] IDQuantique ID3100 Clavis2: http://www.idquantique.com/

[6] S. Ghernaouti-Hélie and M. A. Sfaxi “Quantum Key Distribution within point to point protocol (Q3P)”, IETF draft, 2005.

[7] M. Sasaki, *et al.* “Field test of quantum key distribution in the Tokyo QKD Network” Opt. Express 19 (11), pp10387--10409 (2011)

[8] M. Peev, et. al, “The SECOQC quantum key distribution network in Vienna,” New J. Phys. 11(7), 075001, pp. 2-37 (2009).

[9] B. Korzh, *et al.* “Provably secure and practical quantum key distribution over 307 km of optical fibre”, Nature Photonics 9, 163-168 (2015).