Optimizing Key Consumption in Switched QKD Networks

K. Christodoulopoulos, N. Makris, G. T. Kanellos, D. Syvridis
Department of Informatics and Telecommunications, University of Athens, Greece
kchristodou@di.uoa.gr

Abstract: We consider a switched QKD network and develop a novel scheduling algorithm that periodically configures the QKD links to optimize the generation and buffering of keys to maximize the key consumption rate across the network. ©2024 The Author(s)

1. Introduction

Quantum key distribution (QKD) technology has emerged to offer quantum-safe communication with information-theoretic security [1] and its maturity has recently led to several deployments worldwide [2]. China is leading the way with an ever-expanding QKD network of more than 2000km in 700 fiber QKD links interconnecting a few tens of nodes [3]. Scaling QKD networks from tens to hundreds of nodes that would seamlessly provide keys where needed is quite challenging and requires a powerful key management system (KMS) and integration of the QKD control layer with other networking equipment and applications. In this direction, several standards have been developed over the last few years, including the interface to extract key material, the QKD Control Interfaces for Software Defined Networks-SDN and orchestration (ETSI GS QKD 014, 015 and 018). Building upon such standards, an SDN-enabled dynamically switched QKD network was recently demonstrated with off-the-shelf QKD systems [4], leveraging the flexibility of SDN and of optical switches to offer enhanced resource utilization and key management.

The Dynamic QKD concept employs in every node one Alice and one Bob QKD device (can be generalized to include less or more) attached to a low-loss optical switch, allowing them to interface with any other dual device in the network [5]. A variation is the optical bypass concept [6][7]. For a fully meshed network of N nodes, this approach significantly reduces the number of QKD devices to ~N as opposed to ~N^2 required for establishing direct QKD links among all node pairs. Relayed QKD [6][8], creates keys for nodes that are not directly connected by relaying keys across a path. However, although it uses ~N devices, it consumes keys in every QKD link across the paths, which is inefficient for large networks. Moreover, it does not provide end-to-end security, and requires the use of trusted nodes.

A switched QKD network does not simultaneously have all the QKD links, they are dynamically configured, and generated keys are buffered in the KMS servers or Quantum Key Pools (QKP). The buffer is filled with keys while a QKD link is established between two nodes. These keys are consumed, typically continuously, by Security Application Entity (SAE) pairs residing at those nodes. When the QKD link is torn down to repurpose the QKD equipment (configure other QKD links), the network orchestrator needs to avoid the depletion of the buffered keys or else reestablish that QKD link. SDN-QKD [4,5] should enable the dynamic links configuration and keys management, and employ an effective algorithm to time-manage the resources and optimize the key generation and consumption rates across the network. [6] developed an algorithm to optimize key management that considers optical bypass, trusted relays, and QKP. However, the authors calculate a specific configuration for a period, which does not take full advantage of the dynamic reconfiguration capabilities of optical switches during a period. In [8], a routing, wavelength, and time-slot assignment algorithm for the co-existence of QKD and traditional optical connections is presented. However, the authors do not consider optical switches/bypasses. In addition, a dynamic QKD network must support the capability for any Alice to establish QKD links with any Bob. Non-optimal pairings of arbitrary QKD pairs can lead to lower key generation, which should be accounted for. In this paper we develop a novel scheduling algorithm for dynamically configuring the QKD links to maximize the key consumption rate across the network.

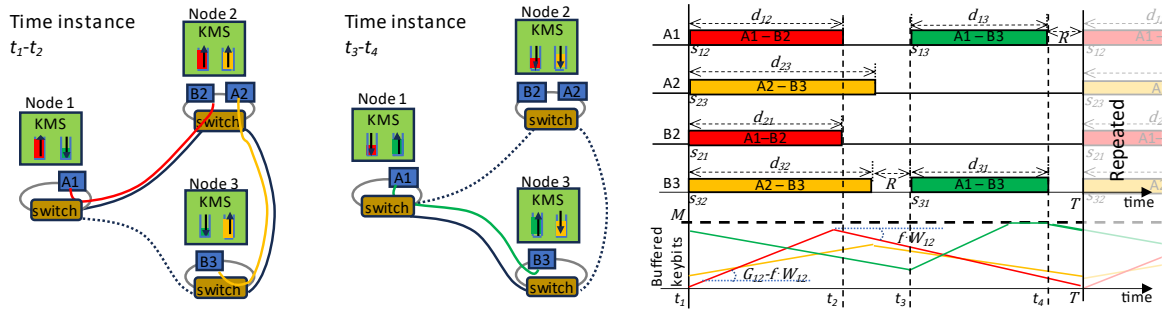

Fig. 1(a) Example of a dynamically switched QKD network (two instances), (b) periodic schedule example and buffers occupancies

2. Dynamic QKD scheduling

We assume a given network topology with N nodes and denote by N_i^A and N_i^B the number of Alices and Bobs at node i, respectively. Typically, we would have one Alice and one Bob per node, but we allow for a higher number to satisfy high key rates and/or large networks. Each node has an optical switch and fiber links to any other node (direct or bypassing other nodes). The switch at a node can be configured to interface an Alice/Bob to a selected fiber; at the other end the respective switch interfaces a Bob/Alice, creating thus the QKD link. We assume that we are given (measured or modelled) the average generation secret key rate (SKR) G_{ij} (key-bits/sec) for each node pair (i,j), capturing the reduced performance of pairing arbitrary QKD equipment and the impairments of the respective fiber connecting the each pair. Dynamically switching an Alice to another Bob, including optical switch configuration and initialization of the newly established QKD link, takes R time.

We operate the switched QKD network in a periodic manner with a period T. We can repeat a scheduler every T, or measure the average generation rates G_{ij} during a period, and calculate a new schedule for next period. When configuring a QKD link (i,j), key material is produced, extracted and stored in buffers of size M_{ij} at the respective KMS servers of nodes i and j. We assume that each QKD link is configured once in a period. The proposed solution is generic and requires that the KMS and QKD control plane provides a key buffer per key generation/consumption pair. We assume a set of SAE pairs; a SAE pair S_{ij} has one SAE at nodes i and one at j that consume QKD keys (e.g. for data plane encryption) from the respective buffers. In practice we could have several SAE pairs between two nodes, but we consider only one (could represent a virtual SAE pair with the aggregated key consumptions). We assign a relative weight W_{ij} to the key consumption rate of S_{ij}. This allows for the normalization of the achieved consumption rates, e.g. a SAE pair that uses keys to encrypt data over a 100 Gb/s data link could have 10 times the weight of another with 10 Gb/s, so that the same amount of data would be encrypted with a single key in both cases, before changing/consuming a new key. The consumption rate of S_{ij} is given by $C_{ij} = f \cdot W_{ij}$, where f is a common variable across all SAE pairs that we maximize. This is done to ensure fairness so that all pairs get equal (weighted) consumption rates. Otherwise, if we maximized the sum of consumptions rates, we would favor and spent more time in QKD links with higher generation rates. Note that the consumption weights and generation rates are symmetric ($W_{ij}=W_{ji}$, $G_{ij}=G_{ji}$). Considering (i,j), the keys are generated for duration d_{ij} in the period T, and they are continuously consumed. Thus, in order to avoid keys starvation, we need to ensure a higher key generation than consumption: $d_{ij} \cdot G_{ij} \geq T \cdot C_{ij}$. Specifically, when the QKD link (i,j) is configured, keys are generated for duration d_{ij}, and the buffers at nodes i and j grow by a rate G_{ij}-C_{ij}. When the QKD link is torn down (the QKD devices are repurposed), for duration T-d_{ij}, the keys are consumed, and the respective buffers empty with a rate C_{ij}. If the buffer of size M_{ij} is filled during the generation duration, the additional keys are discarded. Thus, we also require that $(T-d_{ij}) \cdot C_{ij} \leq M_{ij}$. Given the maximum key consumption rate C_{ij}^{max} (keys bits/sec), we have: $T \leq M_{ij}/C_{ij}^{max} + d_{ij}$. For simplicity in our results we assumed that all buffers have the same size M, all SAE pairs have the same maximum consumption rate C^{max} and set the period $T=M/C^{max}$, which satisfies the above constraint. Note that constraining T and M captures the common security practice of avoiding keeping the keys for a long time. Finally, C_{max} also has a physical meaning, it is the minimum of the speed of the key extracting interface or the maximum consumption rate of the SAE.

The scheduling problem is as follows. We assume as input: the number of Alices N_i^A and Bobs N_i^B per node, the key generations rates G_{ij} and consumption weights W_{ij} per node pairs, the reconfiguration time R, and the period T ($=M/C^{max}$, M is the buffer size, and C^{max} the maximum key consumption rate). We denote by $L \geq T$ a big number. The algorithm finds the periodic schedule: the starting time s_{ij} with respect to the period start, the duration d_{ij} and whether Alice is used in i and Bob in j, or opposite, for the (dynamic) formation of QKD link (i,j). The objective is to maximize a weighted sum of key consumption rates. The calculated schedule is repeated every T, or we measure the average generation rates G_{ij} and create a new schedule for next period. We developed an ILP formulation to solve this problem [9]:

Variables:	ILP formulation:	
f: float, key consumption multiplication factor	max: $\Sigma_{i,j} f W_{ij}$, subject to:	
s_{ij}: float, start time of QKD link (i,j) configuration	For all nodes i, j, with $W_{ij}>0$: $T \geq s_{ij} + d_{ij} + R$	C1
d_{ij}: float, duration of QKD link (i,j) configuration	For all nodes i, j, with $W_{ij}>0$: $Tf \cdot W_{ij} \leq d_{ij} \cdot G_{ij}$, $f \cdot W_{ij} \leq C^{max}$	C2
x_{ij}: binary, equal to 1 if Alice in i and Bob in j, or 0 if inverse	For all nodes i, j, with $W_{ij}>0$: $x_{ij}+x_{ji}=1$, $s_{ij}=s_{ji}$, $d_{ij}=d_{ji}$	C3
u_{ij}: binary, equal to 1 if the start of QKD link (i,j) configuration is before the start of (i,k) that is, if $s_{ij}<s_{ik}$	For all nodes i, j, $k \neq j$, with $W_{ij}>0$, $W_{ik}>0$: $s_{ij} - s_{ik} \leq u_{ijk} \cdot M$, $s_{ij} - s_{ik} \geq (u_{ijk}-1) \cdot M$, $u_{ijk}+u_{ikj}=1$	C4
v_{ij}: binary, equal to 1 if the start time of QKD link (i,j) configuration is before the end of (i,k), that is, if $s_{ij}< s_{ik}+d_{ik}$	For all nodes i, j, $k \neq j$, with $W_{ij}>0$, $W_{ik}>0$: $s_{ik}+d_{ik}+R-s_{ij} \leq v_{ijk} \cdot M$, $s_{ik}+d_{ik}+R-s_{ij} \geq (v_{ijk}-1) \cdot M$	C5
a_{ijk}: binary, equal to 1 if pair (i,j) and pair (i,k) both use Alice in i and overlap in time	For all nodes i, j, $k \neq j$, with $W_{ij}>0$, $W_{ik}>0$: $a_{ijk} \geq x_{ij} + x_{ik} + u_{ijk} + v_{ijk} - 3$, $b_{ijk} \geq x_{ji} + x_{ki} + u_{ijk} + v_{ijk} - 3$	C6
b_{ijk}: binary, equal to 1 if pair (i,j) and pair (i,k) both use Bob in i and overlap in time	For all nodes i, j, with $W_{ij}>0$: $N_i^A \geq x_{ij} + \Sigma_{k,k \neq j} a_{ijk}$, $N_i^B \geq x_{ji} + \Sigma_{k,k \neq j} b_{ijk}$	C7

C1 constrains that the end of each QKD link configuration is less than the period. C2 constrains the consumed to be lower than the generated keys per period and that the consumption rate is less than the maximum. C3 captures the symmetries in Alice-Bob communication. C4-C6 identify the QKD links that overlap in time and use same equipment type (Alice/Bob). C7 constrains the number of simultaneous QKD links to be lower than the available.

3. Dynamic QKD networks performance – indicative results

We performed simulations to evaluate the efficiency of a dynamic QKD network using the proposed ILP scheduling algorithm. If indicated otherwise, we assumed a ring topology of N=10 nodes, adjacent node distances of 10 Km, non-adjacent nodes distances were taken as the chord distances, fiber attenuation coefficient of 0.21 db/km, one Alice N_i^A=1 and one Bob N_i^A=1 per node, buffer M=100 MByte for each node pair, key size of 256 bits, maximum key consumption rate C^{max}= 300 keys/sec (and thus a default period of T=173 min), weighted consumption rate W_{ij}=1 for all pairs, switching time R=5 min, and a realistic SKR as a function of attenuation [10] (decoy state, pulse rate 1GHz, detectors efficiency 12%, dark counts 6E-5, detectors error rate 5E-3, error correction factor 1.22). We considered a 5 dB additional penalty due to unmatched QKD pairs and optical switches. As performance metric we considered the weighted key consumption rate $f W_{ij}$(b/s) which was the same for all pairs, since we set W_{ij}=1 for all. We used CPLEX ILP solver, stopped its execution after 10 min and collected the best solution up to that. We averaged the results over 10 random instances for each set of parameters.

Table 1 shows the consumption rate as a function of various network parameters. We observe that for 5 nodes we achieve the maximum consumption rate (300 Kb/s) per node pair, indicating that one Alice and Bob per node is more than enough in such a relatively small network. The consumption rate decreases as the number of nodes N increases, since Alices and Bobs are time-shared, while we also pay time for switching. Considering different buffer sizes M, we observe a small increase of the achieved consumption rate. In our simulations the buffer size is connected to the period (T=M/C^{max}). So for higher buffer sizes, periods are longer, the keys are generated for longer and consumed for longer durations with the same rates, and the improvement comes from the benefit of a more efficient schedule where switching overhead becomes negligible. As expected, shorter links increase the generation rate and the achievable consumption rate, while longer links have the opposite effect. Similarly adding more equipment is beneficial. However, with 3 devices for topologies with 15 nodes and default buffer size and distance, we achieve the max consumption rate; adding more equipment did not yield any improvement.

Table 1: Dynamic QKD network performance

Number of nodes N	5	10	15	20
Keys consumption rate (Keys/s)	300	276.1	49.4	15.9
Buffer size M (MByte)	50	100	200	400
Keys consumption rate (Keys/s)	267.8	276.1	277.5	283.2
Neighbouring nodes distance (Km)	5	10	15	20
Keys consumption rate (Keys/s)	300	276.1	125.9	54.8
Number of A and B per node (N=15 nodes)	1	2	3	4
Keys consumption rate (Keys/s)	49.4	141.7	300	300

4. Conclusions

In a switched QKD network, QKD links are dynamically established, keys are generated for a portion of time, buffered, and continuously consumed. The network orchestrator should replenish the keys, re-establish a link before the respective buffer is emptied. We developed a novel algorithm that defines a periodic schedule for the QKD links, avoids keys depletions and maximizes the key consumption rate across the network.

5. References

[1] M. Mehic, et al. “Quantum Key Distribution: A Networking Perspective”, ACM Comput. Surv., 53(5):1–41, 2020
[2] A. Lewis, M. Travagnin, “A Secure Quantum Communications Infrastructure for Europe”, JRC Technical Reports, EC, 2022
[3] Y.A. Chen, et al, “An integrated space-to-ground quantum communication network over 4,600 kilometres”, Nature 589, 214–219, 2021.
[4] R. Tessinari, R. Woodward, A. Shields, "Software-Defined Quantum Network Using a QKD-Secured SDN Controller and Encrypted Messages", OFC 2023, paper W2A.38
[5] O. Alia, et al., “Dynamic DV-QKD Networking in Trusted-Node-Free Software-Defined Optical Networks”, J. Lightwave Technol., 2022
[6] Q. Zhang, et al, “Joint Routing, Channel, and Key-Rate Assignment for Resource-Efficient QKD Networking”, Globecom, 2022
[7] K. Dong, et. al., “Auxiliary graph based routing, wavelength, and time-slot assignment in metro quantum optical networks with a novel node structure”, Optics Express, 28 (5), 2020.
[8] Yuan Cao, et al, “Time-Scheduled Quantum Key Distribution (QKD) Over WDM Networks” J. Lightwave Technol., 2018
[9] C. Papadimitriou, K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity", Dover Publications, 1998
[10] Y. Zhao et al., “Simulation and Implementation of Decoy State Quantum Key Distribution over 60km Telecom Fiber,” IEEE ISIT, 2006