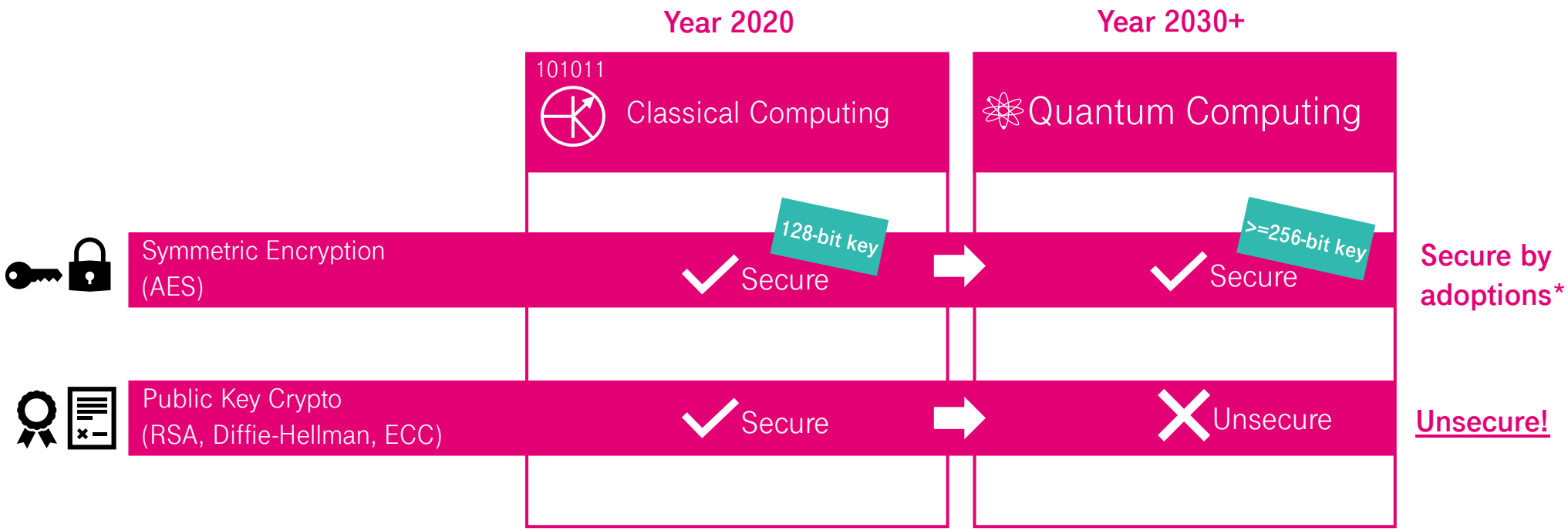
Improved Routing in QKD Networks

Daniel Giemsa, Matthias Gunkel, Tim Johann, Stephan Pachnicke, Robin Böhn, Falk Reuter

Leipzig | 09.05.2023

Motivation - Quantum Threat



Goal is to build a QKD Network as a Platform to distributed quantum-secure symmetrical keys between various sites

* NIST recommendation: 64Gbyte per **AES-256** key invocation according to https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf
→ example: about 4 seconds lifetime or 64bit/s refresh rate per 100G channel

Problem Definition

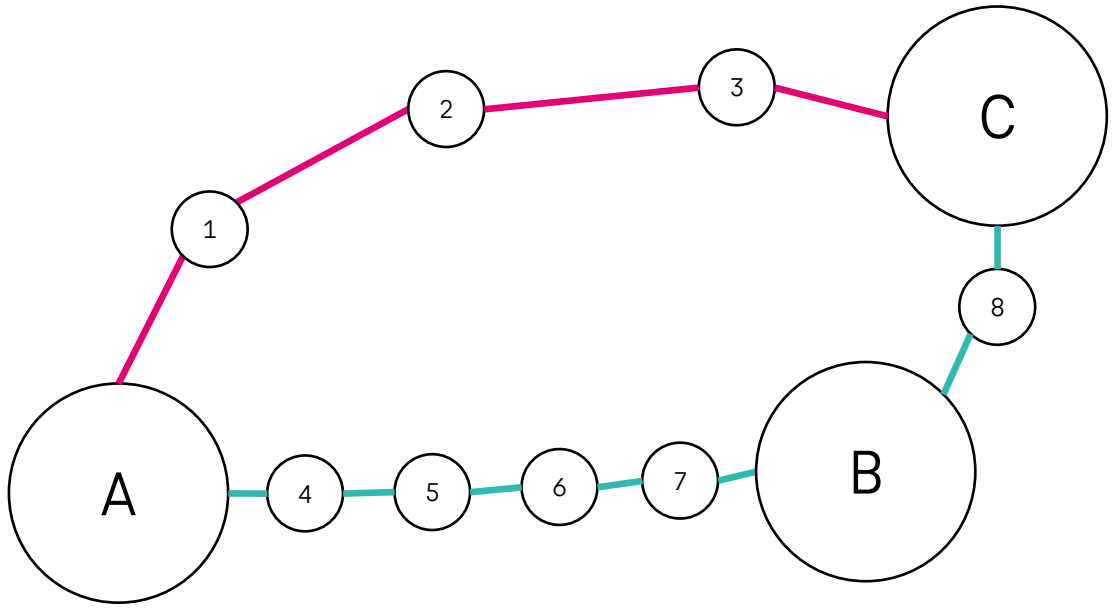
- Find a path to forward keys between two access nodes
- Considering two types of constraints:

Resource Utilization:

- Balance the constraints to improve resource utilization
- Taking detours to avoid heavy frequented links or links with lower key rates
- Limiting detours to avoid overload of the network considering the total key consumption
- Latency can be seen as uncritical → keys can be stored

Security aspects:

- Paths must fulfill certain security aspects to ensure secure key forwarding
- Considering the links (keys) and the nodes



Example Topology of a QKD network

Implementation - Path Computation

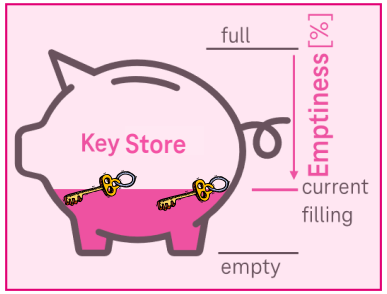
General assumptions:

- Centralized control architecture
- Online Routing: every request will be routed considering the current state of the network

The challenge is to define a method to implement the detour:

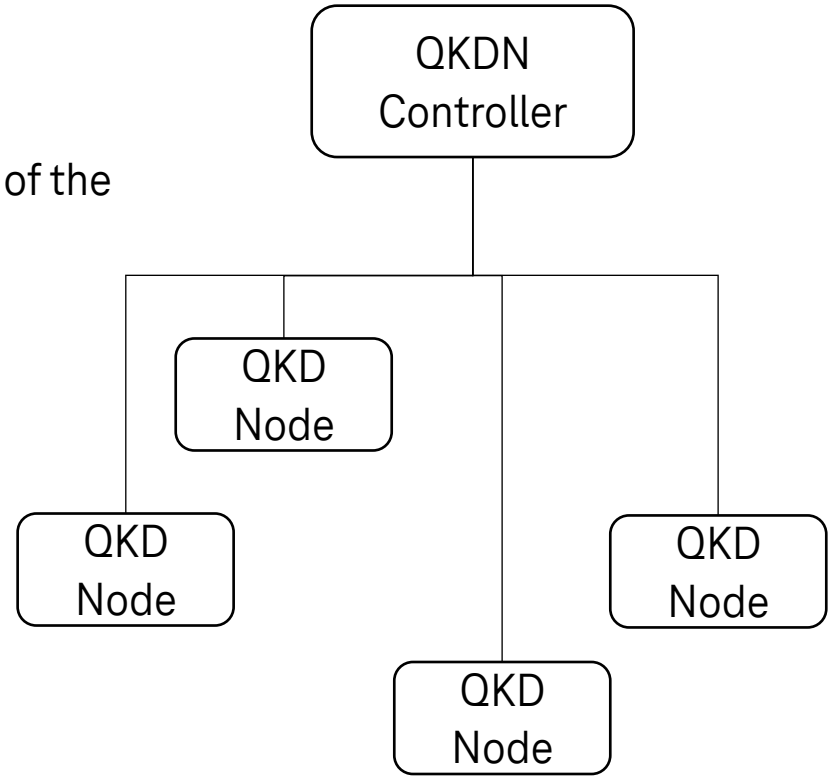
Metrics:

- Secret Key Rate (SKR)
- Emptiness of the Key Store (eks/cap)



Formula:

$$F(SKR, eks) = x * \left(e^{\left(\frac{eks}{cap} * a\right)}\right) + b * \frac{eks}{cap} + (1 - x) * c * \frac{1}{SKR}$$



High Level Architecture according to ETSI 015 (https://www.etsi.org/deliver/etsi_gs/QKD/001_099/015/02.01.01_60/gs_QKD015v020101p.pdf)

Simulation - Initialization

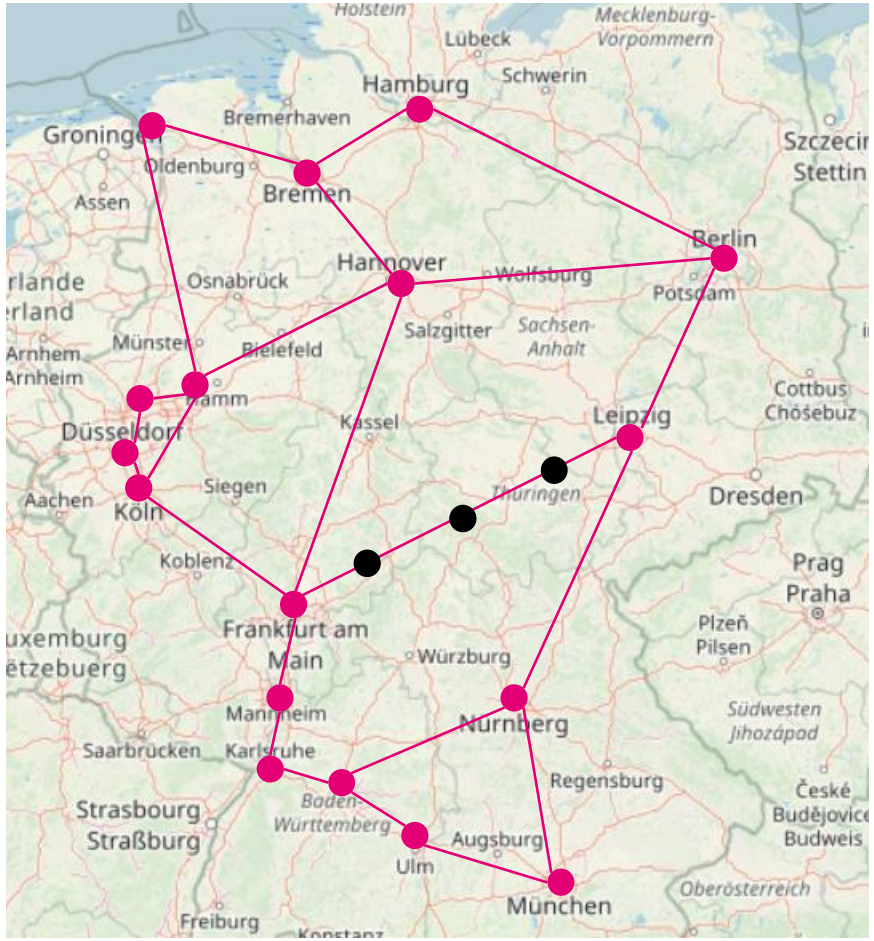
Data:

- Topology from the SND LIB „nobel-germany“
- Contains: Topology, lengths und demands in relation

Assumptions:

- Split longer edges evenly to reasonable lengths for QKD Modules (about 80 km)
- Realistic secret key rates from the model of a QKD system from HHI*
- Generate a request list for individual key requests based on the demands of the SNDlib
- Uniform sizing of the key storage with the aim of achieving a meaningful simulation duration

*Cooperation in the QuNET+ML Project



SDNLib Nobel-Germany (http://sndlib.zib.de/home.action)

Simulation - Procedure

Load initial state in the network

Load the request table

Process the requests by consuming keys on the computed path

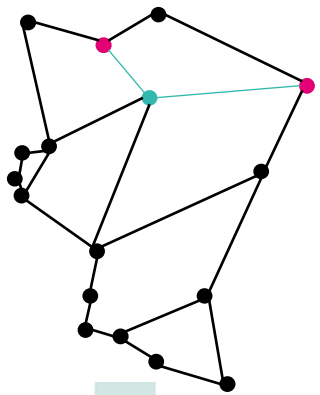
Generate keys on every edge of the network

End Simulation after 140 seconds

Loop

Node A	Node B	Store [Keys]	required rate [Gbit/s]	required keyrate [key/s]	enc_data[Gbit]
Berlin	Bremen	0	600	0	0

Node A	Node B	Requests[Keys]
Berlin	Bremen	1
Berlin	Bremen	1
Berlin	Bremen	1

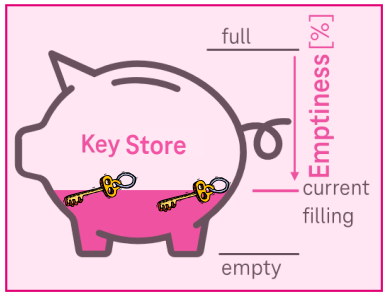


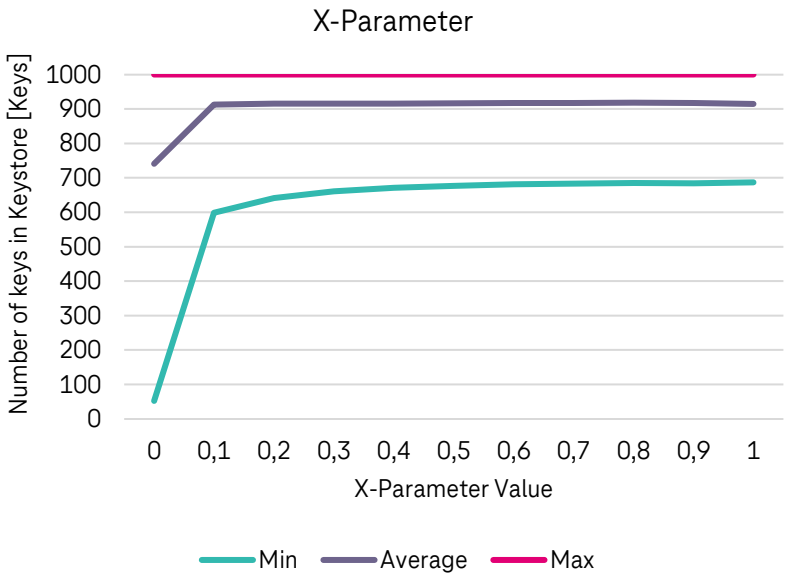
Node A	Node B	Store [Keys]	required rate [Gbit/s]	required keyrate [key/s]	enc_data[Gbit]
Berlin	Bremen	331	600	2,366906	84000

Simulation - Results

Formula:

$$F(SKR, eks) = x * (e^{\left(\frac{eks}{cap} * a\right)} + b * \frac{eks}{cap}) + (1 - x) * c * \frac{1}{SKR}$$

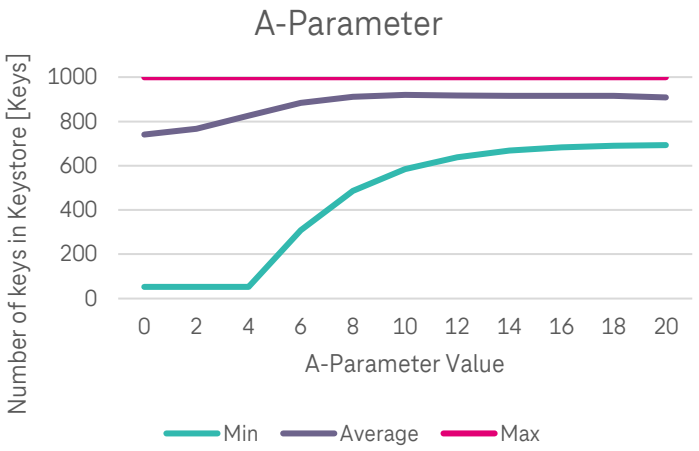


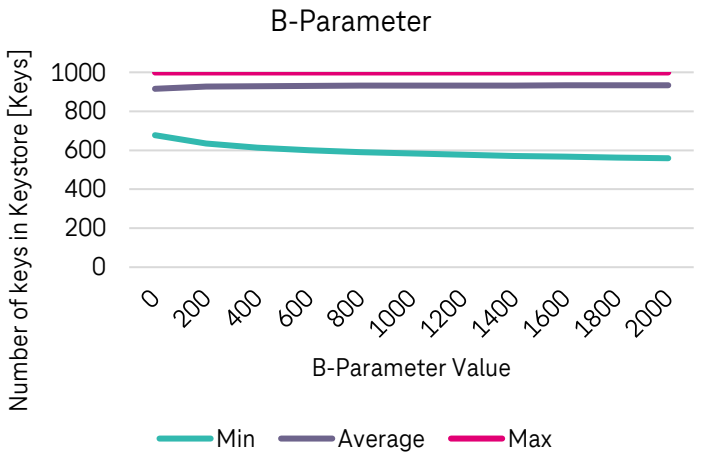


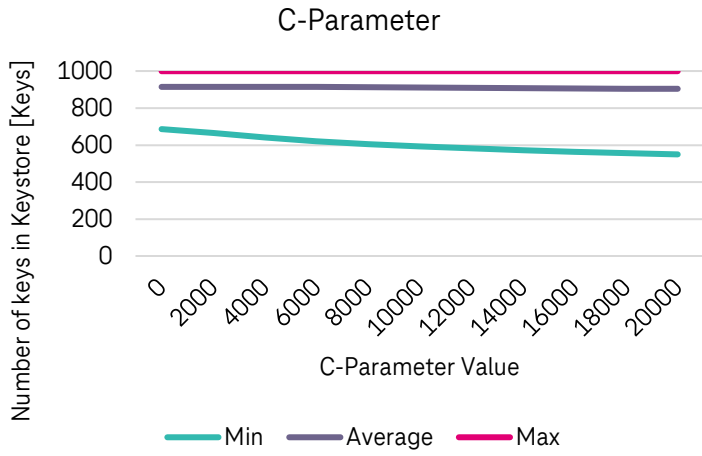
Simulation - Results

Formula:

$$F(SKR, eks) = x * \left(e^{\left(\frac{eks}{cap} * a\right)}\right) + b * \frac{eks}{cap} + (1 - x) * c * \frac{1}{SKR}$$



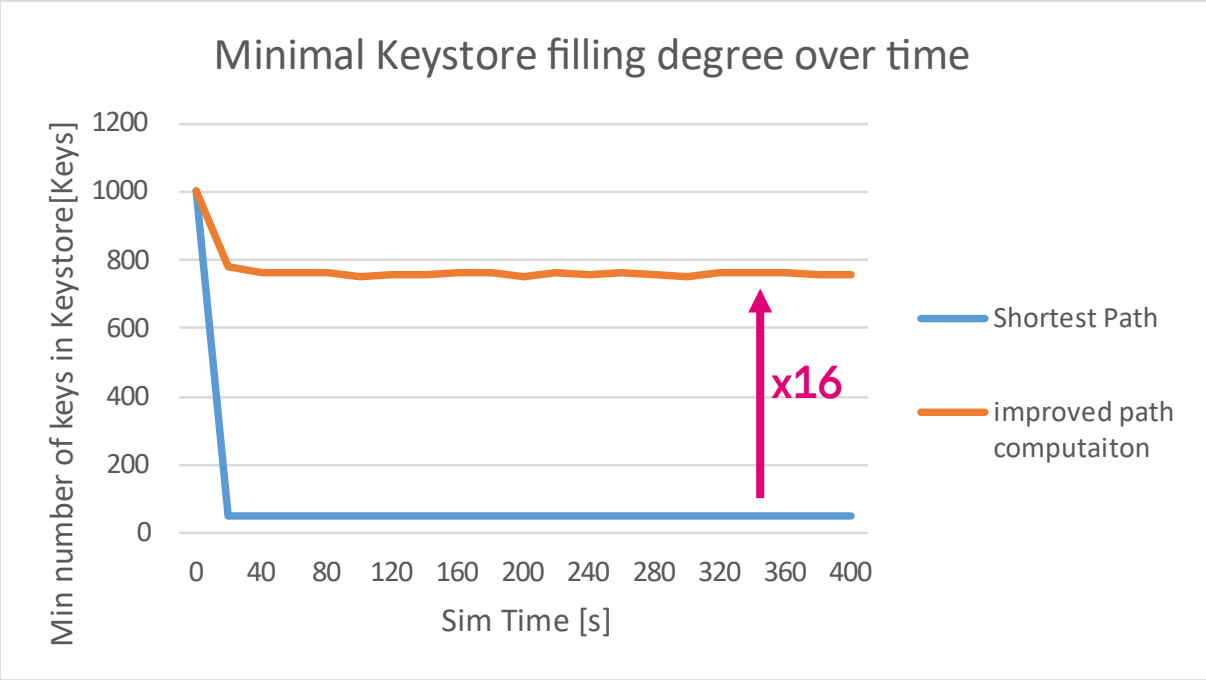




Simulation - Results

Formula:

$$F(SKR, eks) = x * \boxed{(e^{\left(\frac{eks}{cap} * a\right)}} + b * \frac{eks}{cap}) + (1 - x) * c * \frac{1}{SKR}$$



Summary:

- Our simple adapted link metric allows persistent use of Dijkstra without draining individual buffers. It also supports load-balancing as each single key demand is optimally routed.
- "Emptiness" of key buffer is crucial, SKR is only implicitly relevant.

→ Result: $M_{opt} = a\text{^}E$ with $b=0;\ c=0$

OUR MISSION

WE BUILD UP THE QUANTUM SECURITY OF THE FUTURE

Acknowledgements:

DemoQuanDT

QuNET+ ML