Open QKD Network
- An Open Source QKD Network Project

Dr. Xinhua Ling
Project Goals

• Show the feasibility of integrating QKD technology with classical communication networks

• Provide a reference implementation of the 4-layer architecture

• Make it easy for people to showcase QKD technologies at a network scale

• Make it easy for people to deploy pilot QKD networks
Integrate Communication Networks with QKD

Conventional communication network

QKD network

KMS: Key Management Server
Proposed QKD Network Architecture to ETSI

As an input from Canadian QKD network community
In Scope

4-Layer Architecture: QKD technology-independent design

User
(classical commun. net.)

KMS
(Key Management Service)

QNL
(QKD Network Layer)

QLL
(QKD Link Layer)

Requests and uses keys
Manages and Issues Keys to User applications
Extends QKD from Point-to-Point Links to a Network
 Produces Raw Key Bits using QKD technologies

Produces Raw Key Bits using QKD technologies

5
## Out-of-Scope

### User Layer

User utilizes QKD keys at its wish

- can be *any* of the layers 2-5 entities in classical networks, e.g.
- enterprise applications (L5)
- TLS (L4)
- IPsec (L3)
- link encryptor (L2)

Note: It’s the enterprise’s responsibility to develop User Layer, thus beyond the scope of the project.

### QLL (QKD Link Layer)

**All about QKD Technologies**

- A plethora of protocols and platforms to choose from
- Optical fiber based QKD
- Satellite QKD
- …

---

Out of Scope
Main Functionalities Implemented

- **KMS**
  - Interface with user applications, basically aligned to ETSI GS QKD 014
    - Note: further changes will be needed once the QKD-014 is finalized for full conformity
  - Basic key management capabilities
  - Interface with the QNL

- **QNL**
  - Establish shared keys across a network via trusted relay nodes
  - A routing module that can react to QKD network topology dynamics and hide it from KMS
  - Key relaying hop by hop via OTP using QLL generated keys
Demo Applications Provided

- To showcase how (easy it is) to get and use QKD-generated keys
  - Not the main effort of this project

- Demo 1: TLS with PSK
  - A TLS tunnel is established between a client and a server, with QKD keys as the pre-shared key (PSK)
  - A file is transferred over the TLS tunnel

- Demo 2: video chat
  - using qTox* between two users
  - using QKD Keys to protect the confidentiality of the session

* qTox is an open source peer-to-peer communication tool
Source Code

- Hosted at Github.com, actively managed by IQC team
- MIT License
- Commits: 60+
- Attraction: 4 forks to date
## Future Tasks

<table>
<thead>
<tr>
<th>Proposed Work</th>
<th>Potential Contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Enhance the implementation to further improve performance in scalability and robustness</td>
<td>• IQC team</td>
</tr>
<tr>
<td>• Implement standard APIs once they are finalized</td>
<td>• Other research teams</td>
</tr>
<tr>
<td>• Design and implement administrative/operational functionalities such as a dashboard</td>
<td>• QKD-as-a-service provider</td>
</tr>
<tr>
<td>• Investigate container options using Docker etc. for easier deployment</td>
<td>• Classical communication network owners/operators</td>
</tr>
<tr>
<td>• Implement QKD/PQA integration options if PQA candidate is chosen by e.g. NIST</td>
<td>• Standards bodies</td>
</tr>
</tbody>
</table>
Contact

Dr. Michele Mosca  
University Research Chair  
519-888-4567 x37484  
michele.mosca@uwaterloo.ca

Dr. Norbert Lütkenhaus  
Professor  
519-888-4567 x32870  
lutkenhaus.office@uwaterloo.ca

Webpage: https://openqkdnetwork.ca/
GitHub page: https://github.com/Open-QKD-Network