Optimization and CV-QKD Post-Processing in the Open Source AIT QKD Software R10

Oliver Maurhart[1,*], Christoph Pacher[1], Chi-Hang Fred Fung[2], and Momtchil Peev[2]

[1]Security & Communication Technologies, Center for Digital Safety & Security,
AIT Austrian Institute of Technology GmbH, Donau-City-Str. 1, 1220 Vienna, Austria
[2]Optical and Quantum Laboratory, Munich Research Center,
Huawei Technologies Düsseldorf GmbH, Riesstr. 25-C3, 80992 Munich, Germany
[*]oliver.maurhart@ait.ac.at , https://sqt.ait.ac.at/software/projects/qkd

This presentation discusses optimization techniques applied to increase data throughput in the general purpose Open Source AIT QKD Software "R10" for QKD post processing. Derived from the trusted repeater network SECOQC effort and results this software provides a set of building blocks to integrate arbitrary sifting, error estimation, error correction, confirmation, and privacy amplification up to full network integration. Accompanying the basic QKD post processing is the Quantum Point-to-Point Protocol (Q3P) node which enforces information-theoretically secure network peer-to-peer communication for classical applications. An outlook at planned SDN and NFV activities at AIT will be given.

At the heart of the QKD design lies the "QKD Module" realized as a UNIX process which reads in key material and performs various tasks on this data stream ranging from simple BB84 protocol implementation to LDPC algorithms and beyond. As all QKD modules share the same input/output interface provided by the AIT QKD library written in C++11, integration of protocols or algorithms is simple. Furthermore this attempt enables different parallelization techniques by forking and joining the stream of key material. Modules, e.g. error correction, can run concurrently on multiple CPU cores to boost overall key reconciliation throughput. This idea can be applied to a whole series of modules forming several "QKD Pipelines" pushing keys to a single key database.

We examined the throughput bottlenecks in the framework part of the software and addressed these with a complete overhaul of the input and output implementation on the very low levels of the software. A guiding principle has been minimizing the impact of the structural changes and the design resulting in an non-intrusive solution.

This poster will highlight the changes and outlines the process creating QKD post processing pipelines which can handle more than two hundred million samples per second. Parallel to this development we could improve maturity of the software.

Alongside the performance gains introduced in the framework we also added functionality for QKD post processing of continuous variables protocols. We extended the key stream metadata to also include CV-QKD information, we added a module that creates simulated QPSK measurements, an channel estimation module capable of estimating the channel transmittance, the SNR, and the access noise and enriched the software suite to work on key data with different encoding structures like QPSK measurement values.

The design of the AIT QKD R10 software has been done with extensibility, flexibility and robustness in mind. Each module in the pipeline can be easily substituted with implementations of alternative algorithms by independent teams without touching the rest of the pipeline. The core algorithms can be used to implement Quantum Bit-Commitment, Quantum Coin-Flip and Quantum Oblivious Transfer protocols.

The AIT QKD "R10" QKD software is bundled with management tools, partly GUI oriented, based on Distributed Bus (DBus) message exchange technology to simply script QKD modules or AIT QKD based applications with Python or even Bash. Utilities and tools as well as a boilerplate setup for QKD module coding projects can be used to create new QKD post processing modules or QKD based user applications.

The software is mainly available under GPLv3 and AIT welcomes suggestions from academic groups or industry to co-operate.