A Quantum Key Distribution simulator for BB84-type protocols with decoy states

Florian Prawits, Christoph Pacher and Hannes Hübel
Security & Communication Technologies, Center for Digital Safety & Security
AIT Austrian Institute of Technology GmbH, Giefingasse 4, 1210 Vienna, Austria
florian.prawits@ait.ac.at, christoph.pacher@ait.ac.at, hannes.huebel@ait.ac.at

Abstract
The complex relationship of parameters in security proofs for Quantum Key Distribution (QKD) protocols often precludes intuitive approaches, thus constituting a high barrier to entry when trying to reason about the performance of QKD systems. We present a software tool with a graphical user interface (GUI) which can aid in interactive evaluation, design and optimization of BB84-type decoy state QKD systems.

Introduction
Contemporary implementations of BB84-type DV-QKD protocols utilize weak coherent laser pulses as the carrier for the encoded information, which however imposes a severe limitation in the maximally achievable transmission distance due to the inherent threat of photon number splitting (PNS) attacks. This potential weakness can be elegantly eliminated by the adoption of the protocol to include so-called decoy states (DS) in the transmission.

The additional degrees of freedom in deciding when to send signal/decoy states and which intensities to use for them however further complicates the already complex task of anticipating protocol performance and finding a set of suitable parameters to achieve optimal secret key rates (SKR). In order to predict protocol performance, as a function of characteristics of the QKD setup like channel losses and device imperfections, state preparation fidelity, decoy state parameters and finite size effects, the software simulator pyDSsim has been developed. The tool is written in Python and implements the recent security proof framework introduced in [1,2]. The software can be scripted from the command line or used via a graphical user interface (GUI: QT5 framework) for easy exploration via parametrized x-y plots of over 40 different variables, allowing for a comprehensive evaluation of their interdependencies.

The main feature however is the option to numerically compute the set of protocol variables for a given QKD-setup which maximizes the secret key rate under constraints typical for practical implementations: fixed block sizes or fixed acquisition times. To this end two different algorithms (differential-evolution [3] and L-BFGS-B [4]) are utilized, allowing for a cross-check of the acquired results and choice between speed and accuracy of the approach.

Simulator: GUI overview and workflow
1. The user specifies the QKD setup under investigation and fixes the x-axis out of the set \( \{ n, R, t_{\text{on}}, \text{DCR}, \beta_3 \} \). Important input parameters include:
   a) Protocol variables:
      - laser intensities, probabilities for choosing signal/decoy state and X/Z basis
   b) Modes of operation:
      - optimized vs. fixed inputs, optimization algorithm, protocol constraint (blocksize vs. acquisition time)
   c) QKD system parameters:
      - source rate, detection window (i.e. pulse width and jitter), alignment error, detector properties, channel transmittance
   d) Post-processing parameters
2. After hitting the RUN button, any of the over 40 variables of the security proof can be selected as the y-axis from the plot controls.

3. The canvas is automatically updated for the given x-y pair. For direct comparison, plots can be stacked on the canvas and retained between runs of different inputs.
4. The toolbar offers fine grained customisation of the plot, saving figures and exporting calculations to *.csv
5. Additional information about the state of the simulator can be gleaned from the log and progress-bar.

Use case examples

i. Operating a given QKD system: optimal protocol variables for maximum SKR

ii. Detector comparison: superconducting nanowire (SNSPD) vs. InGaAs single photon avalanche (SPAD)

iii. Sensitivity of secure key rate to non-optimal signal (\( \mu_s \)) and decoy (\( \mu_d \)) laser pulse intensities

References

Acknowledgements
This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 820474 (UNIQORN).