An Open-source Software Platform for Numerical Key Rate Calculation of General Quantum Key Distribution Protocols

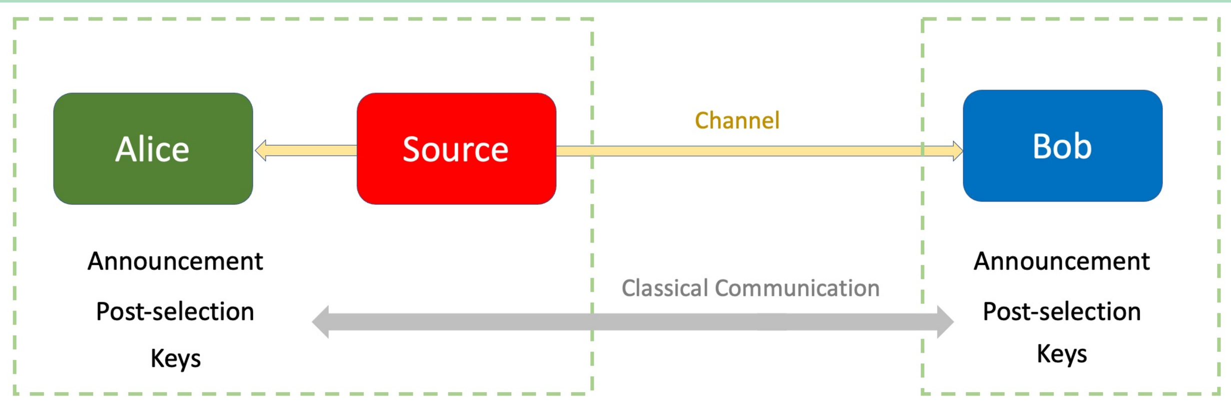
Wenyuan Wang[1], Jie Lin[1], Ian George[1], Twesh Upadhyaya[1], Adam Winick[1], Shlok A. Nahar[1], Kai-Hong Li[1], Kun Fang[1], Natansh Mathur[2], John Burniston[1], Max Chemtov[1], Shahabeddin M. Aslmarand[1], Yanbao Zhang[1,3], Christopher Boehm[4], Patrick Coles[1,5] and Norbert Lütkenhaus[1,*]

[1] *Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1*
[2] *India Institute of Technology Roorkee, Roorkee, India, 247667*
[3] *NTT Basic Research Laboratories and NTT Research Center for Theoretical Quantum Physics, NTT Corporation, 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa, Japan 243-0198*
[4] *University of Freiburg, Freiburg im Breisgau, Germany 79085*
[5] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, US*
[*] *email: lutkenhaus.office@uwaterloo.ca*

In this work, we present an **open-source software platform** that calculates key rate for general QKD protocols, building upon the numerical framework proposed by our group that can perform automated security proof of QKD protocols. The software platform is **fully modularized** with mutually independent modules for descriptions of **protocols/channels**, **solver modules** for bounding key rate, and **parameter optimization algorithms**. It currently supports **BB84 and measurement-device-independent QKD (including decoy states),** as well as **discrete-modulated continuous variable QKD**. It also supports **finite-size analysis** for non-decoy-state protocols. We hope that the open-sourcing can attract theorists to test new protocols and/or contribute to new solvers, as well as appeal to experimentalists who wish to analyze their data or optimize parameters for new experiments.

Background

Our group has proposed a novel **numerical approach** [1,2] for the security proof of general QKD protocols.



A **QKD protocol** can be described in a "prototypical" form [2] as above with the steps of:

- Alice and Bob perform measurements (**POVMs**);
- Alice and Bob make announcements and post-selection based on the state they receive, a process represented by a quantum channel (**Kraus operators**);
- Alice applies **key map** to obtain raw key;
- Alice passes classical information to Bob for **error-correction**;
- Alice and Bob perform privacy amplification to form final key

The key rate is:

$$R = \min_{\rho \in S} f(\rho) - p_{pass} \times leak_{obs}^{EC}$$

where $f(\rho) = D(\mathcal{G}(\rho)||\mathcal{Z}(\mathcal{G}(\rho)))$ is the quantum relative entropy, and maps $\mathcal{G}$ and $\mathcal{Z}$ are defined by the Kraus operator and key map, respectively. The term $p_{pass} \times leak_{obs}^{EC}$ is the leaked information during error-correction.

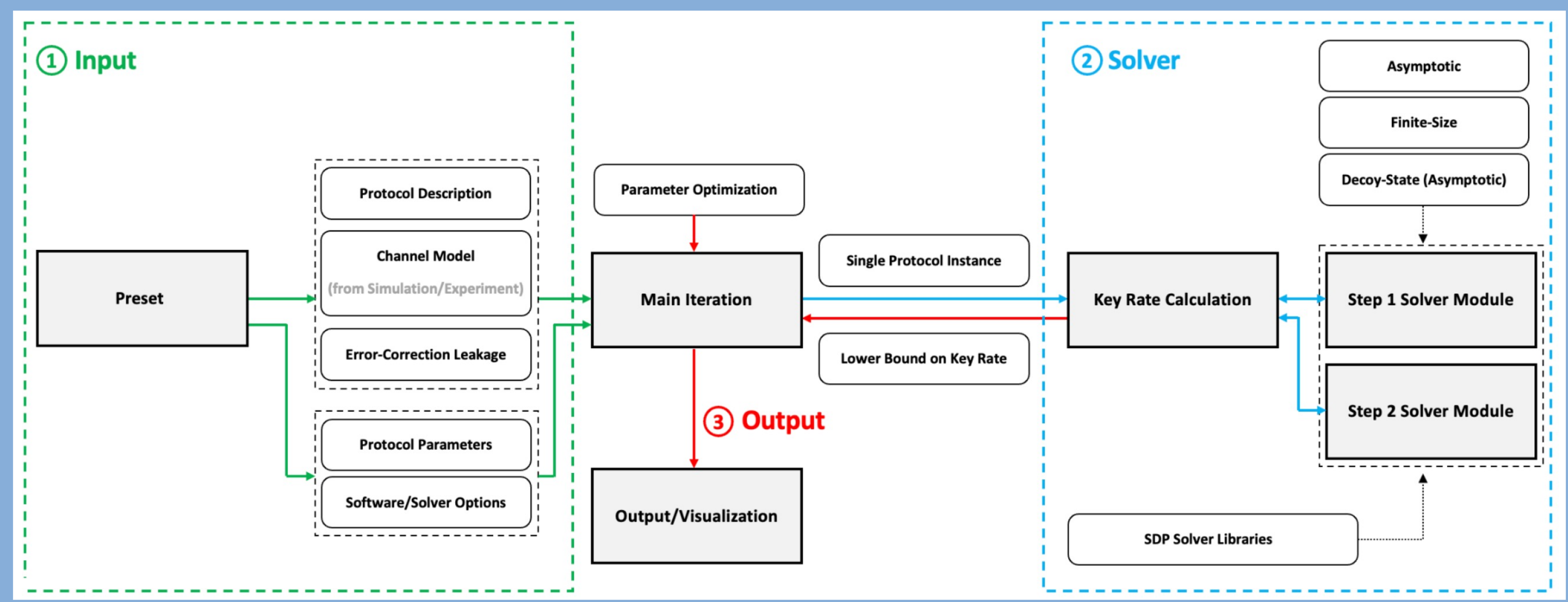
The calculation of key rate comes down to minimizing the privacy amplification part $\boldsymbol{f(\rho)}$, given that ρ satisfies the constraints S given by POVMs $\{\Gamma_k\}$ and their observed **expectation values** $\{\gamma_k\}$.

We can lower-bound the key rate of a protocol, such as using a "two-step approach" to break the optimization into multiple semidefinite-programming problems [2], once we know these information below:

- Kraus operators
- key maps,
- POVMs $\{\Gamma_k\}$,
- expectation values $\{\gamma_k\}$
- error-correction leakage

So far the framework has been **successfully applied to various protocols** such as BB84 and measurement-device-independent QKD [1,3], discrete-modulated continuous-variable QKD [4], **as well as side-channels** such as detector-efficiency mismatch [5] and unbalanced encoding [6]. Finite-size analysis [7] has also been successfully combined with the framework.

Architecture



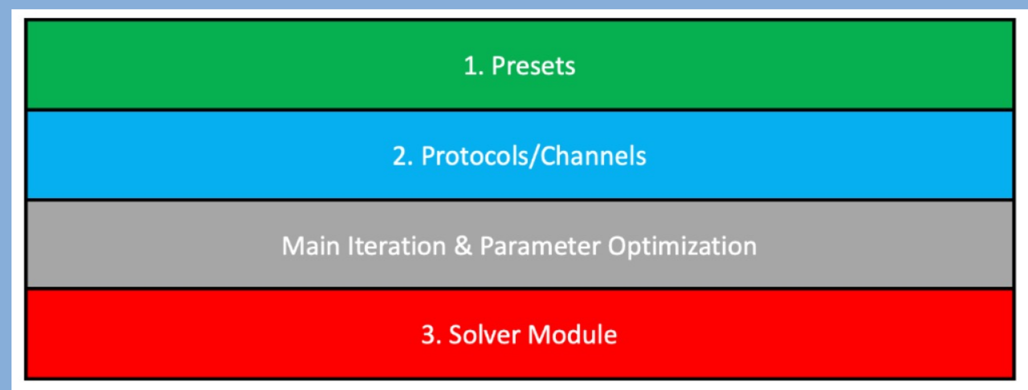
Based on our group's previous works, we present an **open-source platform** to calculate the key rate of general QKD protocols.

The platform is **fully modularized**, with three main types of modules, each independent from the rest and is easily swappable between different modules.

1. The **user-supplied input data:** provides the protocol description and channel model, parameters and solver settings
   - Description file easily caters for various QKD protocols and side-channels
   - Channel model can be from theoretical simulation, can also be from real experimental data
2. The **backend solver module:** takes in a set of data and calculates its key rate;
   - The solver follows the two-step numerical approach to bound key rate for a given instance of protocol. Both asymptotic and finite-size solvers are included.
3. The **main iteration**: iterates or optimizes over a range of parameters. It views the solver module as a black box.
   - The optimization of parameter is decoupled from the protocol/solver. Any number and any combination of parameters can be specified as optimizable (or iterable).
   - User can choose between various optimization algorithms, including e.g. efficient local-search algorithms.

Our platform is also structured such that there are multiple abstraction levels exposed to users with different purposes:

- A **casual user** can pick up one of the *presets* to easily perform simulations or optimize parameters for existing protocols.
- A **theorist** can choose to test key rates of new types of protocols or channels by supplying new *description files*. An **experimentalist** can also replace the channel model with real data to calculate key rate.
- An **expert user** can opt to replace existing *solver modules* with one of their own, so long as it follows the interface of accepting one set of protocol/channel data and returning a key rate.



Current Package Contents

Protocols:

- BB84 (supports decoy states) [3]
- MDI-QKD (supports decoy states) [3]
- Discrete-Modulated CV-QKD [4]

Solvers:

- Asymptotic solver module [2]
- Finite-size solver module (for all non-decoy protocols) [7]
- Gauss-Newton solver (*will be part of future release) [8]

Parameter optimization algorithm (e.g. local search) available for any protocol

Vision

With the **open-sourcing** of the platform, we hope that contributors can bring in **more protocols for testing**, as well as **newer solvers** with better efficacy or accuracy, such as the ongoing collaboration [8], which will be part of the package in the future.

We also hope that the platform will interest **experimentalists** using existing protocol descriptions in the package for analysis of experimental data or optimization of experimental parameters.

References

Project website: openqkdsecurity.org

[1] PJ Coles, EM Metodiev, and N Lütkenhaus. Nature Communications 7 (2016): 1-9.
[2] A Winick, N Lütkenhaus, and PJ Coles. Quantum 2 (2018): 77.
[3] W Wang, N Lütkenhaus, preprint in preparation. Also see **poster #229.**
[4] J Lin, T Upadhyaya, and N Lütkenhaus. Physical Review X 9 (2019): 041064.
[5] Y Zhang, PJ Coles, A Winick, J Lin, N Lutkenhaus, Physical Review Research 3 (2021): 013076.
[6] NKH Li, and N Lütkenhaus. Physical Review Research 2 (2020): 043172.
[7] I George, J Lin, N Lütkenhaus. Physical Review Research 3 (2021): 013274.
[8] H Hu, J Im, J Lin, N Lütkenhaus, H Wolkowicz, arXiv:2104.03847 (2021).