Software for numerical calculation of key rates

Patrick J. Coles, Jie Lin, Adam Winick, Yanbao Zhang, Eric M. Metodiev,
Shouzhen Gu, Electra Eleftheriadou, Filippo Miatto, and Norbert Lütkenhaus
Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, N2L3G1 Waterloo, Ontario, Canada

Background.—Quantum key distribution (QKD) will play an important role in quantum-safe cryptography. The main theoretical problem in QKD is to calculate how much secret key can be distributed by a given protocol. A crucial practical issue is that the QKD protocols that are easiest to implement with existing optical technology do not necessarily coincide with the protocols that are easiest to analyze theoretically [1]. Currently, calculating the secret key output of a protocol is typically extremely technical, and hence only performed by skilled experts. Furthermore, each new protocol idea requires a new calculation, tailored to that protocol. Ultimately the technical nature of these calculations combined with the lack of universal tools limits the pace at which new QKD protocols can be discovered and analyzed. Here, we address this problem by developing a user-friendly software for calculating the secret key output, with the hope of bringing such calculations "to the masses" [2].

The secret key output is quantified by the *key rate* - the number of bits of secret key established divided by the number of distributed quantum systems. Analytical simplifications of the key rate calculation can be made for special protocols that have a high degree of symmetry [3], such as the BB84 [4] and six-state protocols [5]. However, in practice, lack of symmetry is the rule rather than the exception. That is, even if experimentalists try to implement a symmetric protocol, experimental imperfections tend to break symmetries. Furthermore, it is sometimes desirable due to optical hardware issues to implement asymmetric protocols, e.g., as in Ref. [6]. We refer to general QKD protocols involving signal states or measurements that lack symmetry as "unstructured" protocols. Some recent work has made progress in bounding the key rate for special kinds of unstructured protocols, such as four-state protocols [7, 8] and qubit protocols [9]. Still, there is no general method for computing tight bounds on the key rate for arbitrary unstructured protocols. Yet, these are the protocols that are most relevant to experimental implementations.

This motivates our present work, in which we develop a robust, numerical approach to calculating asymptotic key rates [2]. We employ this approach in a computer software, where Alice and Bob input a description of their protocol (e.g., their signal states, measurement devices, sifting procedure, and key map) and their experimental observations, and the software outputs the key rate for their protocol. This software allows for any discrete-variable protocol, including those that lack structure.

At the technical level, the key rate problem is an optimization problem, since one must minimize the well-known entropic formula for the key rate [10] over all states ρ_{AB} that satisfy Alice's and Bob's experimental data. The main challenge here is that this optimization problem is inefficient, with the number of parameters growing as $d_A^2 d_B^2$, for a state ρ_{AB} with $d_A = \dim(\mathcal{H}_A)$ and $d_B = \dim(\mathcal{H}_B)$. For example, if $d_A = d_B = 10$, the number of parameters that one would have to optimize over is 10000. In this work, we give a novel insight that transforming to the *dual problem* (e.g., see [11]) resolves this issues, hence paving the way for automated key rate calculations. In the dual problem, the number of parameters is the number of experimental constraints that Alice and Bob choose to impose. For example, in the generalization of BB84 to arbitrary dimensions [12], Alice and Bob typically consider two constraints: their error rates in the two mutually-unbiased bases. So, for this protocol, we have reduced the number of parameters to something that is constant in dimension. We therefore believe that our approach (of solving the dual problem) is ideally suited to calculate key rates in high dimensions.

Software.—We have developed a software package for implementing our numerical approach. We intend to make this software publicly available in the very near future, likely within the next month. To develop our software, we created a Graphical User Interface using MATLAB, which will work on Mac, Windows, or LINUX. Furthermore, the user does not need a MATLAB license in order to use our software. Figure 1 shows a screenshot. To use our software, the user inputs four pieces of information: (1) the key map (i.e., how the key is derived from the data), (2) the constraints (the observables measured and their expectation values), (3) the post-selection map (e.g., sifting), and (4) the amount of error correction. This information defines the protocol. Once this information is entered, the user can click on the "Calculate Key Rate" button. The computer will then perform the calculation, save the data, and generate a plot of the data. In the "Controls" menu, we provide the option of calculating the key rate using either the dual problem or the primal problem. We also provide the option to perform parallel computing. In the "Help" menu, we provide a list of example protocols, such as the BB84, B92, and six-state protocols. When the user clicks on one of these examples, the parameters associated with that protocol will be loaded into the software.

Applications.—We have a long list of scenarios that we

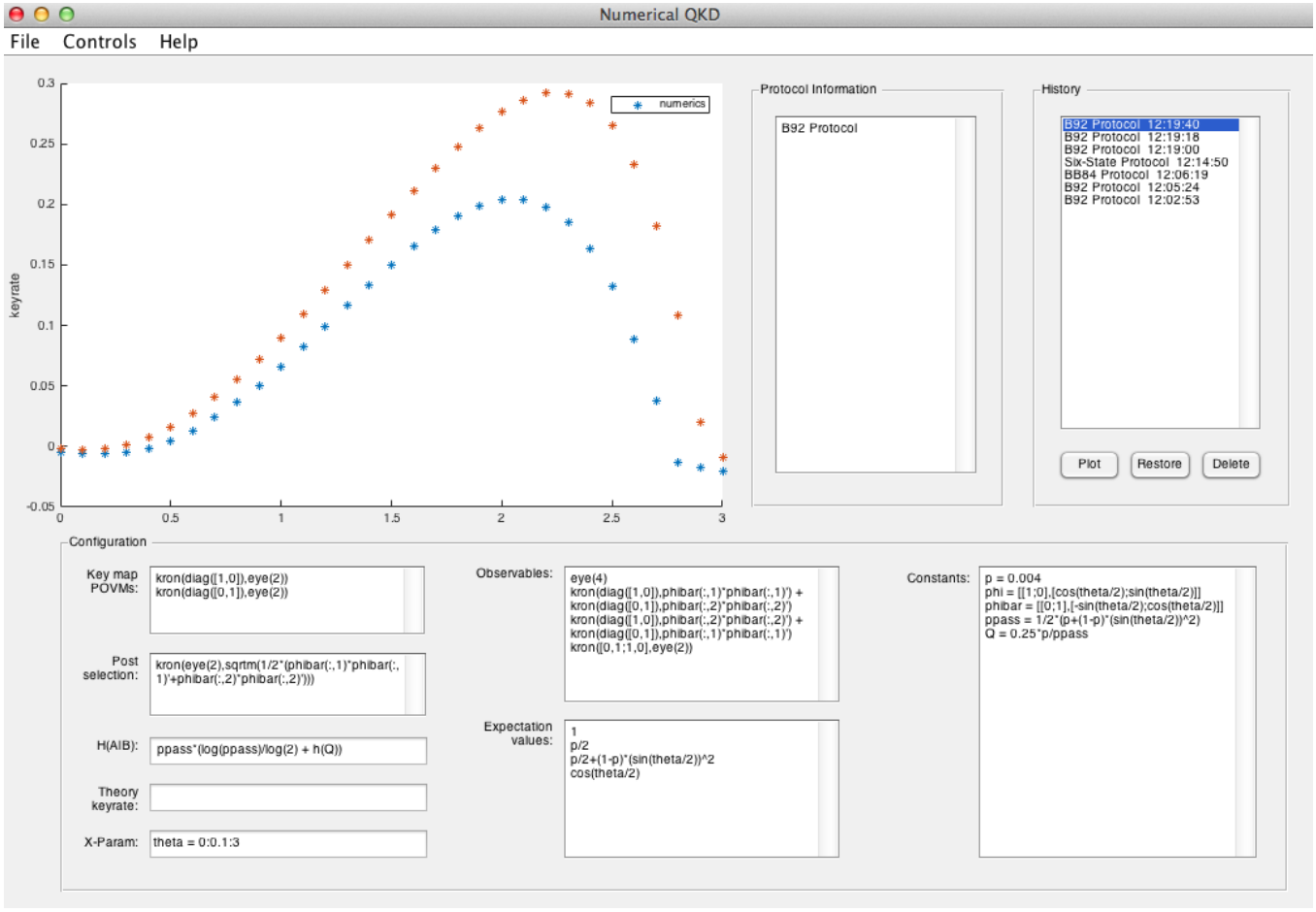

FIG. 1: A screenshot of our software for calculating key rates. The user inputs the following information: (1) the key-map POVM, (2) the constraints, (3) the post-selection, and (4) the amount of error correction. This information defines the protocol, and the software will calculate the key rate based on it.

are interested in applying our software to:

- Side-channel attacks (e.g., trojan-horse attacks)
- Protocols with detector efficiency mismatch
- Protocols involving discrete sets of coherent states
- Measurement-device-independent (MDI) protocols
- Protocols involving decoy states with partial phase randomization
- Distributed-phase-reference protocols, such as the coherent-one-way (COW) protocol

Thusfar we have only investigated the first four topics, as we now discuss. However, we anticipate having results for all the above topics by the time of QCRYPT.

Trojan-horse attack.—Consider a BB84 protocol where Alice encodes using a phase modulator. A well-known hacking attack is the trojan-horse attack [13], which exploits the fact that Alice's phase modulator is not isolated from Eve. Eve sends a pulse of light, some of which passes through Alice's phase modulator and reflects back to Eve, carrying the information about Alice's encoding. Let α denote the amplitude of the light reflected back to Eve. Figure 2 shows the key rate versus α obtained from our numerics, for the case where the signal is a single photon. Our key rates are slightly higher than a recent analytical result from [14], shown as solid curves in Fig. 2. We are currently working on extending our analysis to multi-photon signals.

Efficiency mismatch.—Detector efficiency mismatch is an important issue in QKD because it leads to hacking attacks if not accounted for [15, 16]. Ref. [17] gave an analytical lower bound on the key rate in the case of efficiency mismatch, assuming Bob receives at most a single photon. However, in practice, it is common for Bob to receive a signal with multi-photon contributions. It remains an open problem in QKD theory to ask how the key rate depends on efficiency mismatch, for the general scenario where the photon number distribution is arbitrary. This is precisely the sort of problem on which our

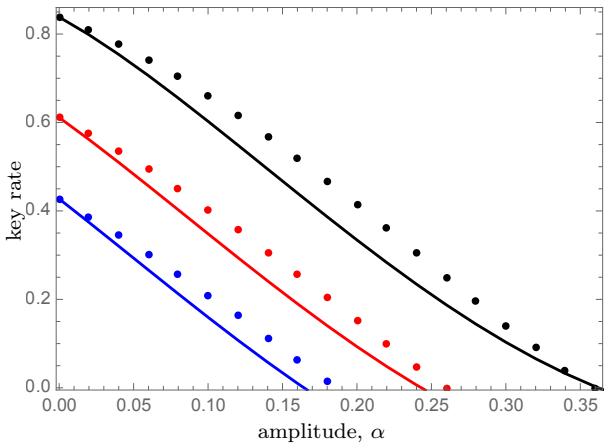

FIG. 2: Key rate versus amplitude α of back-reflected light for the Trojan-horse attack, for error rates of 0.01 (black), 0.03 (red), 0.05 (blue). Our numerics (circular dots) give higher key rates than previous analytical results (solid curves) [14].

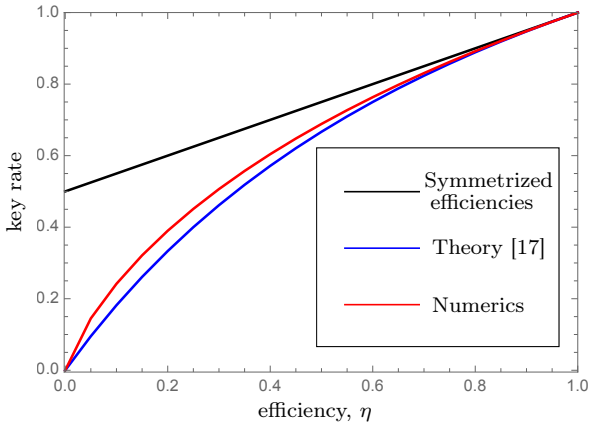

FIG. 3: Key rate versus efficiency η under the single-photon assumption, setting the error rate to zero. We consider a BB84 protocol where Bob's polarization measurement does an active basis choice. The black curve is the case of no mismatch, where both detectors have efficiency $(1+\eta)/2$. The red and blue curves correspond to efficiency mismatch, where one detector has perfect efficiency and the other has efficiency η. The blue curve is the analytical lower bound from Ref. [17]. The red curve is the result of our numerics, via the primal problem. Our result shows that the true key rate is slightly higher than the previously known lower bound.

numerical approach could make progress. Indeed we have some preliminary results for the multi-photon case that look promising. Here we show only the single-photon case in Fig. 3. For simplicity, we assume one detector has perfect efficiency and the other has efficiency η. The red curve in Fig. 3 shows the result of our numerics for this scenario. We find that, for all values of η, our numerics give slightly higher key rates than the result of Ref. [17], which is shown as the blue curve in Fig. 3.

Coherent-state QKD.—We are currently using our software to investigate protocols using weak coherent states with non-randomized phases. For example, we considered one such protocol introduced in [18] and analyzed in [19], and we found that we obtained slightly higher key rates than those of Ref. [19].

MDI QKD.—Using our software we have reproduced the known key rate for MDI QKD with BB84 signal states [20]. We are currently using our software to investigate MDI QKD protocols involving coherent states.

Conclusions.—Our software is robust, efficient, and user-friendly. It has the potential to be a widely used tool in the QKD field. It can even be used to as an educational tool for undergraduate researchers, hence bringing QKD analysis to a wider community.

[1] V. Scarani, H. Bechmann-Pasquinucci, N. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, Reviews of Modern Physics **81**, 1301 (2009).

[2] P. J. Coles, E. M. Metodiev, and N. Lütkenhaus, Nature Communications (in press) (2016), arXiv:1510.01294.

[3] A. Ferenczi and N. Lütkenhaus, Physical Review A **85**, 052310 (2012).

[4] C. H. Bennett and G. Brassard, in *International Conference on Computers, Systems & Signal Processing, Bangalore, India* (1984), pp. 175–179.

[5] D. Bruss, Physical Review Letters **81**, 3018 (1998).

[6] C.-H. F. Fung and H.-K. Lo, Physical Review A **74**, 042342 (2006).

[7] O. Maroy, L. Lydersen, and J. Skaar, Physical Review A **82**, 032337 (2010).

[8] E. Woodhead, Physical Review A **88**, 012331 (2013).

[9] K. Tamaki, M. Curty, G. Kato, H.-K. Lo, and K. Azuma, Physical Review A **90**, 052314 (2014).

[10] I. Devetak and A. Winter, Proceedings of the Royal Society A **461**, 207 (2005).

[11] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2004).

[12] L. Sheridan and V. Scarani, Physical Review A **82**, 030301 (2010).

[13] N. Jain, B. Stiller, I. Khan, V. Makarov, C. Marquardt, and G. Leuchs, IEEE Journal on Selected Topics in Quantum Electronics **21** (2015).

[14] M. Lucamarini, I. Choi, M. B. Ward, J. F. Dynes, Z. L. Yuan, and A. J. Shields, Physical Review X **5**, 1 (2015).

[15] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov, Nature Photonics **4**, 5 (2010).

[16] I. Gerhardt, Q. Liu, A. Lamas-Linares, J. Skaar, C. Kurtsiefer, and V. Makarov, Nature Communications **2**, 349 (2011).

[17] C. C.-H. F. Fung, K. Tamaki, B. Qi, H.-K. H. Lo, and X. Ma, Quantum Inf.˜Comput. **9**, 0131 (2009), 0802.3788.

[18] B. Huttner, N. Imoto, N. Gisin, and T. Mor, Physical Review A **51**, 1863 (1995), 9502020.

[19] H. Lo and J. Preskill, Quantum Information and Computation **7**, 431 (2007).

[20] H.-K. Lo, M. Curty, and B. Qi, Physical Review Letters **108**, 130503 (2012).