QUANTUM COMMUNICATION AND CRYPTOGRAPHY
Quantum Key Distribution

Marcomini Alessandro - 2024286

28/01/2022

1 Project goals

In the framework of communication it is of paramount importance to be able to rely on key distribution systems that allow two legitimate parties to exchange information-theoretically secure messages by the means of cryptography. In this sense, last decades have shown how Quantum Mechanics can enhance secret key distribution protocols thanks to its intrinsic randomness and its superposition laws. The most commonly used protocol of Quantum Key Distribution (QKD) is the BB84 proposed in [1], together with some proposed variations made by either controlling the intensity of an input source (decoy method, [4]) and/oror balancing the use of a "key basis" and a "control basis" in the choices of Alice and Bob while preparing and measuring (efficient BB84, [3]).

This project aims to explore an optical implementation of the three-state one-decoy efficient BB84 QKD protocol by analyzing the experimental data and giving quantitative results on the quality and security of the apparatus. The quantum communication took place in the Quantum Communication Lab in Via Trasea 7 (Padova) and lasted about one hour.

2 Theoretical picture

The analysis carried out in this work follows the steps indicated in [5] for the SKR estimation for one-decoy state QKD protocol with attention to finite-key effects. In particular, the ultimate goal of the analysis has been to relate the quality and quantity of data acquired to the Secret Key Length (SKL) that Alice and Bob can generate. In the framework described above this is upper bounded as:

$$SKL \leq s^l_{Z,0} + s^l_{Z,1}\left(1 - h_2\left(\phi^u_Z\right)\right) - \lambda_{EC} - 6\log_2\left(19/\epsilon_{sec}\right) - \log_2\left(2/\epsilon_{cor}\right) \tag{1}$$

therefore, by estimation of the quantities of the right hand side it is possible to give a (best-case) estimate of the SKL. These quantities are namely a lower bound on the number of detections when Alice sends vacuum states ($s^l_{Z,0}$) and single-photon states ($s^l_{Z,1}$), the binary entropy of the upper-bounded phase error rate (ϕ^u_Z), an error correction term depending of the QBER in the key basis (λ_{EC}) and an extra contribution from the security levels set by the user (ϵ_{sec}, ϵ_{cor}). As equation 1 suggests, the secret key length is enhanced if the majority of detections are not connected to multi-photon events, while the bound decreases while enlarging the security levels one wants to achieve and if the error rates in the channel are too high.

In the following paragraph we describe how each of the quantities in equation 1 has been estimated starting from the data, following the instructions of the appendix A of [5], which we refer to for theoretical justifications. However, it is important to stress that these formulas have been retrieved

in a finite-key scenario and for some of the quantities to be estimated the finite-key effects have been proved very strong. As a result, in order to avoid uninformative estimations (such as negative lower bounds on the number of detections), data have been grouped together to construct large enough datasets: for each of them the following quantities have been calculated. More details on the dataset size choice are discussed in the analysis section.

Counts and Quantum Bit Error Rate

First of all, once the dataset to work on is defined, one can immediately estimate the total number of cases in which the key basis Z has been used (n_Z, equal for A and B due to sifting) and the total amount of mismatches m_Z for data in the Z basis. Similar reasoning applies to the control basis X. Moreover, while focusing only on the cases in which Alice is preparing the state using intensity $k = \mu_1$ or $k = \mu_2$ it is possible to estimate $n_{Z,k}$ and $m_{Z,k}$ for both signal and decoy states. Again, same reasoning applies to $n_{X,k}$ and $m_{X,k}$.

From these, one can compute the Quantum Bit Error Rate (QBER) for each basis and in different conditions simply as:

$$QBER_Z = \frac{m_Z}{n_Z} \qquad QBER_X = \frac{m_X}{n_X} \tag{2}$$

$\epsilon-$security parameters

In the estimate of SKL (equation 1) there are explicit references to the values of $\epsilon_{sec}-$secrecy and $\epsilon_{cor}-$correctness we want the protocol to be subject to. These values are in principle to be set by the user and of course it holds that the more stringent these requirements are, the more bits we must deplete in post-processing operations. In particular, by setting ϵ_{cor} we are imposing an upper bound on the amount of mismatches we are willing to allow Alice and Bob's keys to have after the communication and post-processing. The value of ϵ_{sec} quantifies instead the (maximum) amount of information the Eavesdropper can have regarding the secret key.

Coming to the specifics of the analysis below, the numerical values for the security parameters have been set to $\epsilon_{sec} = 10^{-9}$ and $\epsilon_{cor} = 10^{-15}$, which are typical values as specified in [5]. To be fully precise, the ϵ_{sec} term is indeed the composition of the secrecy parameters of all the single components of the post-processing routine, and to fully characterize the final outputs one would have to specify in detail all these terms and give an estimate of ϵ_{sec} with higher accuracy. For the sake of simplicity and in absence of further details, here the approach mimics the one of *Rusca et al.* by setting all error terms equal to an identical value ϵ_1 which satisfies $\epsilon_{sec} = 19\epsilon_1$ (the 19 factor comes from the number of times approximated bounds are used in the definition of the secret key length formula, see [2]).

Zero and one-photon detection bounds

The SKL estimate requires an indication on the number of single-photon and vacuum events. However, being practically impossible to measure the number of incoming photons together with the quantum state of light, one has to simply evaluate on a statistical basis a reasonable quantity to estimate the number of detection for zero and one-photon pulses. In [5], *Rusca et al.* suggest two approaches for the upper bound on the vacuum case. The first is given by:

$$s^u_{Z,0} = 2\left(m_Z + \delta\left(n_Z, \epsilon_1\right)\right) \tag{3}$$

where n_Z,m_Z are defined above and $\delta\left(n, \epsilon\right) \equiv \sqrt{n \log\left(1/\epsilon\right)/2}$. The second instead holds for data related to a single intensity ($k = \mu_1$ or $k = \mu_2$) of the ones Alice can choose to prepare her state

with:

$$s^u_{Z,0} = 2\left(\tau_0 \frac{e^k}{p_k}\left(m_{Z,k} + \delta\left(m_Z, \epsilon_1\right)\right) + \delta\left(n_Z, \epsilon_1\right)\right) \tag{4}$$

where $\tau_n = \sum_{k\in\kappa} p_k e^{-k} k^n/n!$ is the total probability of sending a $n-$photon state (sum of Poissonian probabilities for all the possible preparation intensities κ) and p_k is the probability for Alice to choose k among the possibilities in κ.
These formulas allow to collect three different estimates of the upper bound on $s_{Z,0}$, which can work better or worse according to the size of the dataset they are calculated on. For further analysis one can thus consider the more stringent one (i.e., the minimum of the three bounds).

In the finite-key scenario it is possible to provide also a lower bound to vacuum detection in the measure basis:

$$s^l_{Z,0} = \frac{\tau_0}{\mu_1 - \mu_2}\left(\mu_1 n^-_{Z,\mu_2} - \mu_2 n^+_{Z,\mu_1}\right) \tag{5}$$

where $n^\pm_{Z,k} = n_{Z,k} \pm \delta\left(n_Z, \epsilon_1\right)$ for $k \in \kappa$.

The upper bound on the vacuum detection in the Z basis is needed in the estimate of the lower bound on the single-photon states:

$$s^l_{Z,1} = \frac{\tau_1\mu_1}{\mu_2\left(\mu_1 - \mu_2\right)}\left(n^-_{Z,\mu_2} - \frac{\mu_2^2}{\mu_1^2} n^+_{Z,\mu_1} - \frac{\left(\mu_1^2 - \mu_2^2\right)}{\mu_1^2}\frac{s^u_{Z,0}}{\tau_0}\right) \tag{6}$$

Phase error rate and error correction

As for the other terms in equation 1, the phase error in the Z basis can be upper bounded as:

$$\phi^u_Z = \frac{v_{X,1}}{s_{X,1}} + \gamma\left(\epsilon_{sec}, \frac{v_{X,1}}{s_{X,1}}, s_{Z,1}, s_{X,1}\right) \tag{7}$$

where $s_{X,1}$ can be estimated applying equation 6 to the data referring to X basis measurements, $v_{X,1}$ represents the number of bit errors in the X basis for single-photon events (see below) and the γ function is defined as:

$$\gamma\left(a, b, c, d\right) = \sqrt{\frac{\left(c+d\right)\left(1-b\right)b}{cd\log 2}\log_2\left(\frac{c+d}{cd\left(1-b\right)b}\frac{21^2}{a^2}\right)} \tag{8}$$

The value of $v_{X,1}$ can be upper bounded by the means of information on the X basis measurements, as shown in [2]:

$$v^u_{X,1} = \frac{\tau_1}{\mu_1 - \mu_2}\left(m^+_{X,\mu_1} - m^-_{X,\mu_2}\right) \tag{9}$$

Where $m^\pm_{Z,k} = m_{Z,k} \pm \delta\left(m_Z, \epsilon_1\right)$ for $k \in \kappa$.

As for the error correction term λ_{EC}, the approach used is the one described in [2]: in a complete implementation (quantum communication together with post-processing) its value should be set equal to the number of bit publicly exchanged between Alice and Bob for the sake of error correction (which are of course removed from the total length of the secret key). However, in absence of this piece of information an estimate can be achieved by considering a fixed error-correction efficiency factor f (usually between 1.3 and 1.8 in the literature, here arbitrarily set to 1.5 for the sake of simplicity) to multiply the binary entropy of the QBER in the key basis. As a result, in the computation of equation 1 we consider $\lambda_{EC} = f \cdot h_2\left(m_Z/n_Z\right)$.

The whole set of equations above allows to compute the SKL starting from the total number of counts and errors in a dataset, eventually filtered on the basis choice and/or the preparation intensity. Ultimately, we can translate the SKL into the Secret Key Rate (SKR) simply considering $SKR = SKL/\Delta_t$, where Δ_t is the acquisition time corresponding to data used to compute the SKL (for chunks of multiple seconds, the SKR returned is an average).

3 Data analysis

The whole set of operations described below have been carried out in Python by direct implementation of the formulas described above.

Experimental settings

From the experimental point of view, the implementation follows the instructions for the 3-state 1-decoy efficient BB84 protocol. The key basis is Z, the control basis is X. In the preparation stage, Alice selects the preparation basis at random with probabilities $P_Z^A = 90\%$, $P_X^A = 10\%$. Moreover, she prepares and sends only three states: $|H\rangle$ or $|V\rangle$ in the Z basis and just $|D\rangle$ in the X basis. Bob instead performs a measure in the X or Z basis chosen at random with $P_X^B = P_Z^B = 50\%$.

As for the signal and decoy intensities, these have been set respectively to $\mu_1 = 0.4699$ and $\mu_2 = 0.1093$ photons per pulse. Alice selects at random an intensity $k \in \kappa \equiv \{\mu_1, \mu_2\}$ to prepare her pulse with probabilities $P_\mu(\mu_1) = 70\%$ and $P_\mu(\mu_2) = 30\%$.

Loading and cleaning

The whole set of data to analyze consists of three binary files containing information on the states prepared by Alice, the ones measured by Bob and the intensity level chosen (signal, strong intensity μ_1 or decoy, weak intensity μ_2). The raw keys have been already pre-processed, i.e. events have been synchronized and non-received qubits have been removed. The acquisition time of these data is slightly less than one hour (3519 seconds): following the encryption schema used to construct the binary files, at each second an indicated number of bytes has been read and each byte has been turned into a four-state tuple. The endianness of the binary-to-int conversion has been selected by comparing the first outcomes to known results.

	A	B	D	N	A_basis_Z	B_basis_Z
0	V	V	S	1	True	True
1	H	H	S	1	True	True
2	V	V	S	1	True	True
3	H	H	S	1	True	True
4	H	H	S	1	True	True
...	...	...	...	...	...	...
419775	V	V	S	50	True	True
419776	V	V	S	50	True	True
419777	V	V	S	50	True	True
419778	V	V	S	50	True	True
419779	H	H	S	50	True	True

419780 rows × 6 columns

Figure 1: Example of sifted data. A: Alice's state; B: Bob's state, D: decoy intensity (either Strong or Weak), N: block number (corresponding to the second of arrival).

At this point it is possible to have a full view of data: each event (coincident measure of A and B) is characterized by the state sent and the preparation basis, the state measured and the measurement basis, together with Alice's intensity choice. Following the post-processing procedure of the protocol, the first step to perform is the sifting: since Alice is randomly choosing preparing the state in basis X or Z there are no useful correlations when Bob and she select different bases. Therefore, a first filter over the dataset is applied by only keeping coincidences where the same basis is used for both preparation and measure. This procedure almost halves the total length of the raw keys:

the amount of shared bits surviving the sifting is of the order of $40'000\ bit/s$. Finally, data manipulation has been concluded by tagging each sifted acquisition block (corresponding to one second of quantum communication) with a unique label and saving the datasets to file: this allows to enhance performances for the analysis. An example of sifted data is reported in figure 1, while an estimate of the average number of sifted bit per second has been carried out over chunks corresponding to 200 s of acquisition time and is displayed in figure 2.

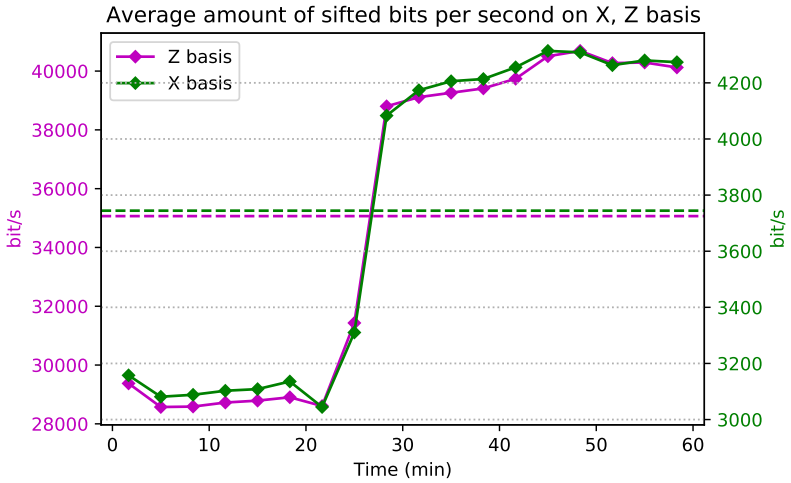

Figure 2: Amount of sifted bit per second on the X, Z bases.

The trends in figure 2 show a sudden increase in the flux of bits after about half an hour: this is likely due to some changes in the efficiency of the channel. Moreover, by looking at the scale of the rates for the X and Z basis one can notice how the ratio between average values of these quantities if $X : Z \sim 1 : 9.3$, in accordance to the nominal probabilities that Alice uses to prepare her states $(10\% : 90\%)$. A final note: having considered chunks of 200 s over a total of 3519 s (see below) resulted in a final chunk of solely 119 s: this, properly normalized, shows that the bit flux between Alice and Bob is almost steady in the final part of acquisition, but the total number of bits for the last chunk is eventually smaller. This has some side effects in the estimate of the SKR.

QBER computation

At this point it is straightforward to compute some quantities of interest: first of all, an index on the quality of the communication held by Alice and Bob is given by the Quantum Bit Error Rate. As described in equation 2, it is sufficient to consider for each block and basis the total coincidences number and the amount of errors among them. Results have been computed for each second and reported in figure 3.

As one can infer from the graph, the QBER on the control basis X is smaller on average but displays larger fluctuations. As for the one in the key basis, we observe a quite defined trend which is changing faster in the median region of the data acquisition. This is probably related to the large increment of the total number of sifted bits that is visible in figure 2. Nevertheless, absolute values of both QBERs remain in a reasonable range.

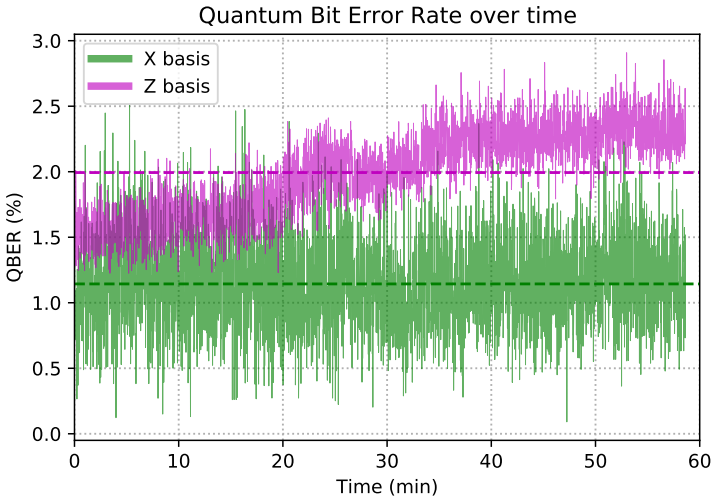

Figure 3: Quantum Bit Error Rate per second

Finite-key effects and Secret Key Length

As mentioned above, while trying to perform the analysis described in section 2 some of the bounds proposed by *Rusca et al.* fail to provide meaningful information. For example, if we try to calculate the SKL for the data corresponding to an acquisition time of one second the bounds for $s^l_{Z,0}$ and $s^l_{Z,1}$ result in negative values. This is useless for the analysis: the number of detections made by Bob is in any case lower-bounded by 0. Even by flooring to zero the bounds for negative outcomes, the analysis still failed due to an unreasonable estimate of ϕ_Z.

This problematic behaviour can be related to the finite-key effects that become extremely relevant for short keys, such as the one Alice and Bob can extract from a single second of quantum communication. Therefore, we gradually enlarged the amount of data to process by aggregating datasets for multiple seconds of acquisition. Ultimately, for large enough datasets ($\mathcal{O}(10^7)$ bits, corresponding to ~ 150 s) the finite-key limits start to be less heavy and the bounds begin to return non-trivial results.

Therefore, the estimate of the SKL has been carried out over data chunks of 200 s. The results are reported in figure 4. Of course, analyzing chunks of data comes with a cost: we have more events (which allows to weaken finite-key effects on the SKR bound), but on the other hand we allow for higher acquisition and computational time. The value 200 s has been chosen according to some trials and the capabilities of our machinery.

The trend of the SKR follows the one of the raw key stream and presents a steady behaviour in the first and second region of the plot (the last point drops in value due to the smaller number of counts in the last chunk of reduced size, as mentioned). While we did not find similar results in the literature to compare the goodness of our estimates and check the quality of the implementation, we can affirm that the behaviour of the SKR follows the theoretical expectations.

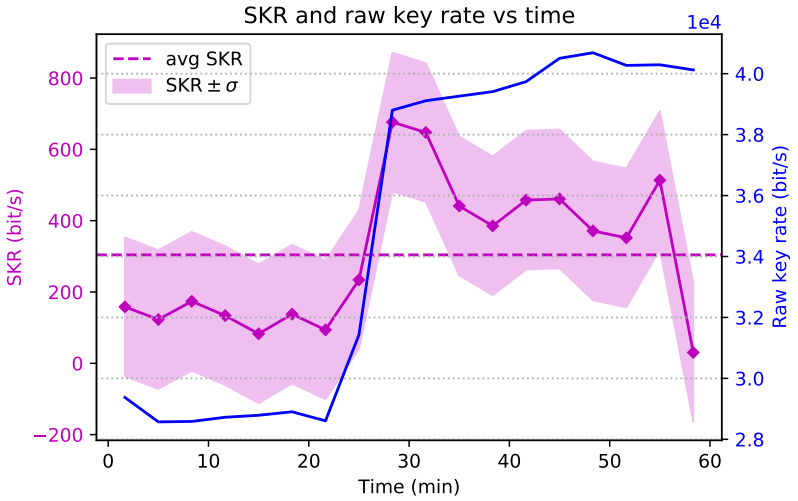

Figure 4: Secret Key Rate (purple) and raw key rate (blue) per second (average over 200 s)

Cumulative analysis and asymptotic behaviour

Other interesting results arise while taking a cumulative approach and performing the SKL bound estimation over sets of increasing size (namely, considering the first 200 s chunk, then the first two chunks altogether and so on and so forth). It is known that, due to the security requirements and natural noise of the channel, there is only a limited SKR we can achieve in the limit of infinitely many sifted bits in the key basis. The idea is that if the input key is long enough, the bounds provided in [5] become the ones for the $\infty-$key case. Results are reported in figure 5.

We perform the analysis calculating for each cumulative chunk the SKL as the SKR over the equivalent time in seconds. Finally, we estimate the asymptotic limit by considering the average value of counts and errors ($\langle n_Z \rangle$, $\langle m_Z \rangle$, etc) over the single chunks and multiplying them for a large enough factor χ. Indeed, we are modeling a communication system between Alice and Bob which is exchanging bits at an higher rate and with a QBER which is similar to the (almost) steady ones we observe (i.e., we are simulating an experiment with the same preparation and measurement settings but with less losses). Plugging these numbers into the code to retrieve the bounds for the SKL, we find that for $\chi = 10^3, 10^5, 10^8$ the SKR estimate does not change anymore, meaning we reached the $\infty-$key limit. This asymptotic value is displayed in blue in figure 5. As we can notice, the SKR has a sudden growth for short raw key lengths while the slope becomes smaller and smaller by enlarging the input (the relative gain between the last two points is 0.6%). The trend seems to be compatible with the asymptotic limit and the last simulated point has a relative distance of $\sim 13\%$ from the infinite case.

While figure 5 shows clearly how a larger input raw key can increase the SKR at the output, it is not to be forgotten that this does not come for free. Firstly, it is non-trivial (and sometimes non-feasible) to enlarge the generation rate of raw bits to high values and reach long enough raw keys. Secondly, one should also take into account the amount of time needed to concretely perform the post-processing operations, together with the machinery at disposal. In fact, by augmenting the length of the raw key we are increasing both the secret key rate and the computational time required to compute the secret key. Therefore, in practice one must find a compromise between these factors.

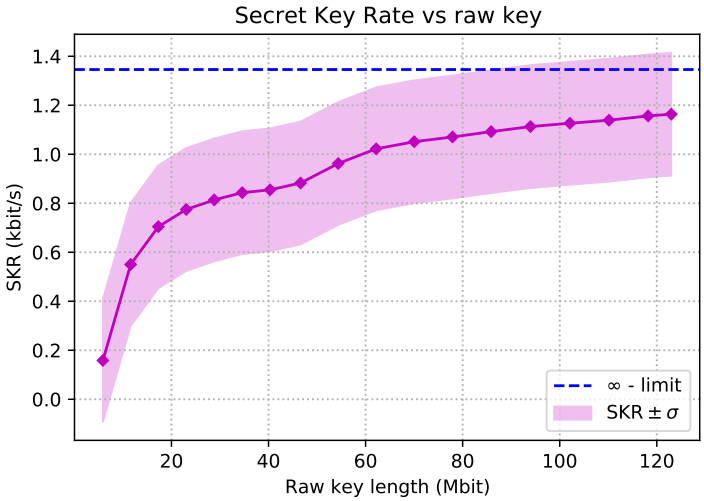

Figure 5: SKR for different raw key lengths, corresponding to the secret key rate and the number of sifted Z bits for chunks of data of increasing size.

4 Conclusions

In conclusion, we managed to compute the quantities of interest and to provide meaningful insights regarding both the quality of the channel (by the QBER) and the bound for the secret key rate for the polarization-based QKD implementation under review. The former happens to display reasonable behaviour for the noise (control basis) and an acceptable error rate on the key basis which is shifting as a consequence of the internal changes of the channel taking place halfway the acquisition process. As for the SKR, this happens to fluctuate a lot over time: in the first section it is steady around a value of 200 bit/s while in the second region fluctuations are more marked.

Moreover, an ulterior analysis has been carried out to study the behaviour for large raw key blocks. The trend is here well defined and in line with theoretical expectations, with a controlled increment that points to the asymptotic value.

Future development could investigate larger amount of data and higher order statistics, together with more detailed information regarding post-processing operation and timing.

References

[1] Charles Bennett and Gilles Brassard. Withdrawn: Quantum cryptography: Public key distribution and coin tossing. volume 560, pages 175–179, 01 1984.

[2] Charles Ci Wen Lim, Marcos Curty, Nino Walenta, Feihu Xu, and Hugo Zbinden. Concise security bounds for practical decoy-state quantum key distribution. *Phys. Rev. A*, 89:022307, Feb 2014.

[3] Hoi-Kwong Lo, H.F. Chau, and M. Ardehali. Efficient quantum key distribution scheme and proof of its unconditional security. *Journal of Cryptology*, 18:133–165, 04 2005.

[4] Xiongfeng Ma, Bing Qi, Yi Zhao, and Hoi-Kwong Lo. Practical decoy state for quantum key distribution. *Physical Review A*, 72, 03 2005.

[5] Davide Rusca, Alberto Boaron, Fadri Grünenfelder, Anthony Martin, and Hugo Zbinden. Finite-key analysis for the 1-decoy state qkd protocol. *Applied Physics Letters*, 112(17):171104, 2018.