1-decoy 3-state efficient BB84 implementation and security analysis

Nicole Zattarin

1 Introduction

Quantum Key Distribution (QKD) is an approach for sharing symmetrical keys between distant users, usually referred as Alice and Bob, in an information-theoretically secure way. The implementation is based on establishing an optical link between Alice and Bob. Since implementations are based on employing optical cables that are already operating worldwide, at the state of art, QKD is considered one of the leading protocols in quantum cryptography, thus such field is considered mature enough for real world applications. Moreover, unconditional security has been proved for several QKD protocols, which makes this strategies extremely powerful, because they can guarantee security without imposing any restriction on the power of the eavesdropper.

The first proposal is the well known BB84 Bennett and Brassard [1984], originally meant to work with true single-photons. Nevertheless, from a practical point of view, deterministic single-photon sources are still not available. Therefore, nowadays applications employ weak coherent laser pulses. In this work we first briefly introduce the main theoretical background, then we discuss the implementation of a QKD protocol based on the usage of coherent states of lasers, the so called *decoy states*.

2 Background

In this section we briefly introduce the generic setting of QKD, with particular reference to the BB84 protocol proposed by Bennett and Brassard [1984]. We discuss in particular the parameters of the protocol and the generic steps of post-processing.

2.1 BB84

Let us consider a system depicted in Figure 1, where we refer to the two legitimate users as Alice and Bob. They employ a classical authenticated channel to share classical information, and a quantum channel which is open to any manipulation from an adversarial. Therefore, an eavesdropper, let us call her Eve, can listen to all communication that takes place on the first channel, while she can take part in the quantum communication, by manipulating the shared states. Before the beginning of the communication, Alice and Bob select two basis that they are going to use to generate and measure the states. We will discuss the details of such mechanism in the next sections.

2.1.1 The protocol

In the classical BB84 Alice sends to Bob a sequence of photons prepared in different polarization states, chosen randomly from two complementary bases. To be more specific, one can choose to work with the basis $\{|H\rangle, |V\rangle\}$ of horizontal/vertical polarization, and $\{|D\rangle, |A\rangle\}$ for diagonal/antidiagonal photons. For each photon received, Bob selects either H/V or D/A randomly with equal probability, and he

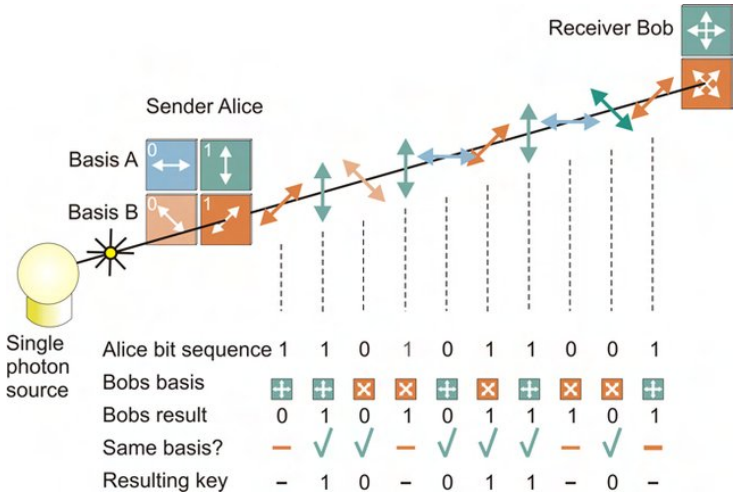


Figure 1: Schematic representation of BB84. Alice sends a photon encoded in a specific basis to Bob, who chooses a basis and performs a measurement. Then, Alice and Bob broadcast their measurement bases on the classical channel: if they chosed different bases they discard the bit. Source Stock [2011].

measures in such basis. The next step is usually referred as sifting: Alice and Bob broadcast their measurement bases on the classical channel. If they chosed different bases they just discard the bit, while the remaining bits compose the so called *sifted keys* κ_A^S and κ_B^S, which have same length, but are assumed to be different in general. Eventually, sifted keys must be post-processed in order to retrieve two identical secure keys $\kappa_A''^S = \kappa_B''^S$. The main steps of post-processing are:

1. Information reconciliation: sharing information over the public channel Alice and Bob produce two equal keys $\kappa_A'^S = \kappa_B'^S$, by leaking the less possible information to Eve;
2. Error verification: after information reconciliation we can assume that $\kappa_A'^S = \kappa_B'^S$ with high probability, thus error verification is meant to detect if there are residual errors in the keys;
3. Privacy amplification: final step to produce $\epsilon-$unconditional secure keys that are independent from Eve.

2.1.2 Intercept-resend attack and error rate

The advantage of the quantum protocol is particularly evident once we take into account Eve attack. Let us consider the simple scenario in which Eve intercepts the photon sent by Alice, she measures it and then she sent the collapsed state to Bob. In the quantum framework the action of an attacker may be detected because its presence introduces an error in the final key, due to a fundamental property of quantum mechanics: a state subjected to measurements collapses. Indeed, if Eve measures in the same basis chosen by Alice the state does not change, but when she chooses the wrong one she sends to Bob a state in her own measurement basis. Thus, in this case Bob will get an error with 50% probability even when he measures in the same basis as Alice.

Therefore, the action of Eve introduces an error in Bob's key, whose rate is usually referred as Quantum Bit Error Rate (QBER). This error can be employed to compute secret key rates, a quantity that quantifies the amount of secure bits available, thus the level of compression needed to make the key private. In formula secure key rates reads:

$$r = \max\{I_{(A,B)} - I_E, 0\}, \tag{1}$$

where I_{AB} is the mutual information between Alice and Bob, and it can be computed as:

$$I(A,B) = H(A) - H(A,B) = 1 - h_2(Q) = 1 - Q\log_2 Q - (1-Q)\log_2(1-Q), \tag{2}$$

being h_2 the binary entropy and Q the QBER.

2.1.3 Generalizations and improvements

The BB84 protocol and the intercept-resend attack are simple but yet well-explanatory examples, nevertheless the field of QKD is wide and open to different proposals and strategies. From the point of view of the protocol, an issue of the classical BB84 is that, in average, half of the photons are discarded during sifting. To avoid this problem, a more efficient protocol employs two basis X and Z with biased probabilities $P_X \approx 1$ and $P_Z \approx 1 - P_X$. In this way the basis Z is used only to detect the presence of an eavesdropper, and it is possible to minimize the loss due to sifting. Moreover, QKD can be generalized to d-dimensions and continuous variables, while Alice can exploit different strategies to attack.

2.2 Decoy state protocol

The original proposal for the BB84 protocol is based on the usage of single-photon sources, which is technologically not available yet. Nevertheless, it is possible to build an efficient QKD protocol by means of a weak pulsed laser source. A laser generates a coherent state, to which we will refer as *decoy state*, that can be written as:

$$|\alpha\rangle = \sum_0^\infty \frac{\alpha^n}{\sqrt{n!}} e^{-\mu/2} |n\rangle , \tag{3}$$

where μ is the intensity of the laser and $|n\rangle$ are the eigenstates of the number operator. More in general, we fix μ and we randomize the phase, thus the emitted state is described by a poisson mixture:

$$\rho = \sum_0^\infty P_\mu(n) |n\rangle \langle n| \quad \text{with} \quad P_\mu(n) = \frac{\mu^n}{n!} e^{-\mu}. \tag{4}$$

However, it is worth to highlight that, by employing laser pulses, we cannot control how many photons are generated and, in general, sources generate multiple photons states. This aspect can be exploited for a very effective attack: the so-called photon-number-splitting (PNS) attack, see Huttner et al. [1995] for further details. To avoid such issue, many implementation of the decoy-state method based on minimal changes to the classical BB84 have been proposed. The idea behind such works is straightforward: Alice randomly and independently generates states with different mean photon number, in such a way that Eve cannot adapt her attack to the specific situation. Since PNS depends on the intensity of the pulse, the choice of such strategy prevent the protocol to be attacked by means of PNS.

2.2.1 Protocol description

We discuss the 3-state 1-decoy efficient BB84 analyzed in Rusca et al. [2018], the key idea is that decoy states are used to detect attacks, whereas the standard BB84 states are used for key generation only.

Let us consider an efficient BB84, in which two basis X and Z have asymmetric probabilities $P_Z \approx 1$ and $P_X \approx 1 - P_Z$. Moreover, we employ phase-randomized laser pulses, in a 1-decoy setting: the intensity of each laser, i.e. the number of photon emitted per second, is randomly set to either μ_1 or μ_2.

Therefore, for each laser pulse and for each bit y_i, Alice randomly chooses both the basis a_i and the intensity $k \in \mathcal{K} = \{\mu_1, \mu_2\}$, then she sends the state to Bob via the quantum channel. On the other side, Bob chooses his basis b_i, he performs the measurement and record its outcome y_i'.

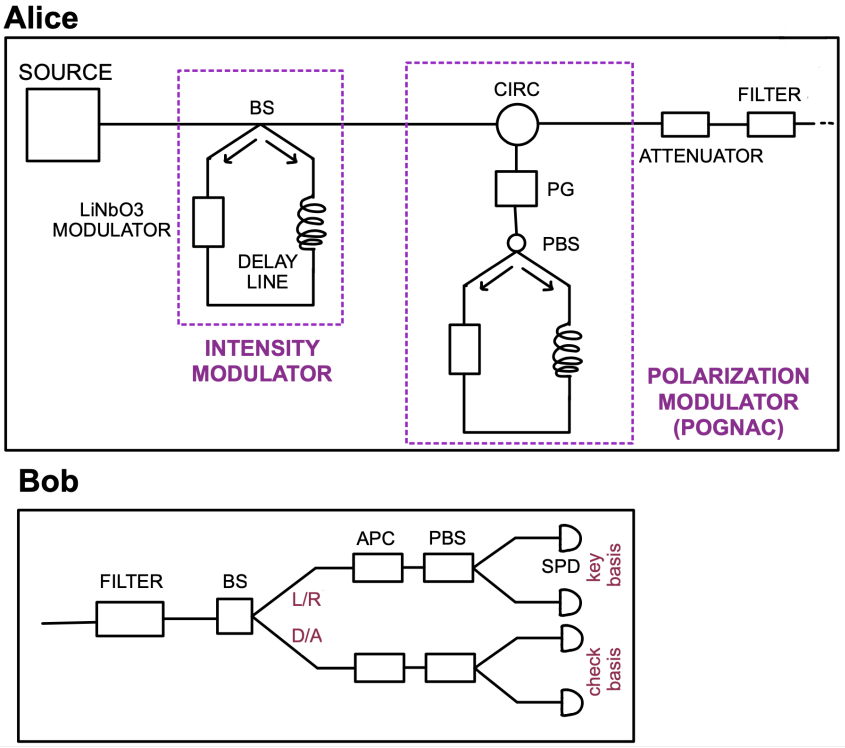


Figure 2: Schematic representation of the apparatus. On Alice side, a source generates pulses, that are then subjected to intensity modulation and polarization modulation, by means of POGNAC. An attenuator and a filter complete Alice's side. On the other hand, Bob's side consists of a filter, a 50% BS and a sequence of APC, PBS and single photon detectors for each branch.

The next step is basis reconciliation: Alice and Bob communicate their choices over the public channel and they individuate the two sets of pairs state generation/measurement which they performed in the same basis for a given intensity. In practice they compute: $X_k = \{i : a_i = b_i = X,\ k \in \mathcal{K}\}$ and $Z_k = \{i : a_i = b_i = Z,\ k \in \mathcal{K}\}$. Then, Alice and Bob generate the final raw key of fixed postprocessing block size $n_Z = \sum n_{Z,k}$, where $n_{Z,k}$ is the cardinality of the corresponding set: $|Z_k| = n_{Z,k}$. Note that here we are following the procedure exposed in Lim et al. [2014], assuming that both the intensity levels are employed for the key generation, while typically QKD protocols use only one decoy. After this procedure, they are able to compute the number of bit errors in each basis $m_{Z,k}$ which is fundamental quantity to provide security bounds, indeed in the next section we discuss formally how to compute a bound to security by measuring errors and coincidences.

Finally, during the post-processing Alice and Bob perform an error-correction step that reveals at most λ_{EC} bits of information.

2.2.2 Security bounds

First of all, let us introduce the basic parameters of our security analysis: given $\epsilon_{sec}, \epsilon_{cor} > 0$ we assume that the protocol is $\epsilon_{sec} + \epsilon_{cor}$ secure if it is ϵ_{sec} secret and ϵ_{cor} correct. Coherently with the literature we fix $\epsilon_{sec} = 10^{-9}$ and $\epsilon_{cor} = 10^{-15}$. Under such assumptions, a ϵ_{sec}-secret key length of the protocol

can be bounded to the quantity:

$$\ell \leq s_{z,0}^{low} + s_{z,1}^{low}(1 - h(\phi_Z^{up})) - \lambda_{EC} - a\log_2(b/\epsilon_{sec}) - \log_2(2/\epsilon_{cor}), \tag{5}$$

where $s_{z,0}^{low}$ and $s_{z,1}^{low}$ are the lower bounds to vacuum and 1-photon events and $h(x) = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy. Moreover, ϕ_Z^{up} is the upper bound on the phase error rate, and λ_{EC} is the number of disclosed bits in the error correction. The two parameters a and b depend on the analysis that is taken into account, details of the discussion for the 2-decoy protocol are reported in the appendix of Lim et al. [2014], in our specific case we assume $a = 6$, $b = 19$.

Let us now express explicitly the quantities reported in Eq. (5), first of all we have to compute the bounds that appear in such formula. Let us consider a QKD protocol in which the key is encoded in the Z basis, in the finite key scenario Hoeffding's inequality Hoeffding [1963] provides the following bound on the number of observations:

$$|n_{Z,k}^* - n_{Z,k}| \leq \sqrt{\frac{n_Z}{2}\log(1/\epsilon_1)}, \quad \text{with probability } \; 1 - 2\epsilon_1, \tag{6}$$

where $n_{Z,k}^*$ and $n_{Z,k}$ are the number of detection in the asymptotic scenario and in the finite key, for each decoy intensity k, while n_Z is the total number of observations. Considering a different error ϵ_2, the same inequality can be applied to bound the error rate:

$$|m_{Z,k}^* - m_{Z,k}| \leq \sqrt{\frac{m_Z}{2}\log(1/\epsilon_2)}, \quad \text{with probability } \; 1 - 2\epsilon_2. \tag{7}$$

Thus, from the previous definitions we can define the following quantities, that will be useful for the next computations:

$$n_{Z,k}^{\pm} = \frac{e^k}{p_k}\left(n_{Z,k} \pm \sqrt{\frac{n_Z}{2}\log(1/\epsilon_1)}\right), \quad m_{Z,k}^{\pm} = \frac{e^k}{p_k}\left(m_{Z,k} \pm \sqrt{\frac{m_Z}{2}\log(1/\epsilon_2)}\right). \tag{8}$$

Moreover, it is possible to show (Rusca et al. [2018] , Lim et al. [2014]) that by fixing $\epsilon_1 = \epsilon_2 = \epsilon$ we obtain $\epsilon_{sec} = 19\epsilon$. From now on we will always assume $\epsilon_{sec} = 19\epsilon$.

For what concerns the lower bounds on the vacuum and single photon events, assuming $\mu_1 > \mu_2$ and applying the finite-key corrections, the following inequalities hold:

$$s_{Z,0} \geq s_{Z,0}^{low} = \frac{\tau_0}{\mu_1 - \mu_2}\left(\mu_1 n_{Z,\mu_2}^- - \mu_2 n_{Z,\mu_1}^+\right), \tag{9}$$

$$s_{Z,1} \geq s_{Z,1}^{low} = \frac{\tau_1\mu_1}{\mu_2(\mu_1 - \mu_2)}\left(n_{Z,\mu_2}^- - \frac{\mu_2^2}{\mu_1^1}n_{Z,\mu_1}^+ - \frac{\mu_1^2 - \mu_2^2}{\mu_1^2}\frac{s_{Z,0}^{up}}{\tau_0}\right), \tag{10}$$

$$s_{Z,0} \leq s_{Z,0}^{up} = 2\left(\tau_0 m_{Z,k}^+ + \sqrt{\frac{n_Z}{2}\log(1/\epsilon_1)}\right). \tag{11}$$

The term τ_n represents the probability that Alice sends a n-photons state, in formula:

$$\tau_n = \sum_{k\in\mathcal{K}} \frac{e^{-k}k^n}{n!}p_k. \tag{12}$$

Finally, the phase error can be estimated as follows:

$$\phi_Z \leq \phi_x^{up} = \frac{v_{X,1}^{up}}{s_{X,1}^{low}} + \gamma\left(\epsilon_{sec}, \frac{v_{X,1}^{up}}{s_{X,1}^{low}}, s_{Z,1}^{low}, s_{X,1}^{low}\right), \tag{13}$$

where:

$$\gamma(a,b,c,d) = \sqrt{\frac{(c+d)(1-b)b}{cd\log 2}\log_2\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)}, \tag{14}$$

being $v_{Z,1}$ the number of error detected for a single photon event, whose upper bound reads:

$$v_{X,1} \geq v_{X,1}^{up} = \frac{\tau_1}{\mu_1 - \mu_2}\left(m^+_{X,\mu_1} - m^-_{X,\mu_2}\right). \tag{15}$$

Note that to compute the phase error we need to measure the number of observed errors in the basis which is not used to encode the key.

Finally, once we have a bound on the secret key length, Eq. 5, the secret key rate (SKR) can be computed as follows:

$$SKR = \frac{\ell}{N_{tot}}R, \tag{16}$$

Where N_{tot} is the total number of pulse sent in order to have a key of block size n_Z, and R is the repetition rate of the source, that we assume unitary.

Moreover, in order to calculate the QBER on the Z basis, we can just evaluate the ratio between the total probability of error and the total probability of detection:

$$QBER_Z = \frac{m_Z}{n_Z}, \tag{17}$$

with trivial analogous in the X basis.

3 Experimental realization

Now that we have discussed all the theoretical details, we still need to describe the actual implementation of the protocol.

3.1 Apparatus and measurements

In order to represent the quantum states we choose light polarization as degree of freedom for the system, thus the apparatus must be set up in order to control possible changes in the polarization of the qubits. Indeed, actual systems usually employ optic fiber, which could change the polarization of the photons.

As previously discussed, we implement the 1-decoy 3-state protocol described in Rusca et al. [2018], in which the authors compare the performances of 1-decoy and 2-decoy approaches. Following the method used by Lim et al. [2014], they conclude that for most experimental settings, the use of only 1-decoy level is advantageous. In our implementation we fix the selection probabilities at the transmitter as: 90% for the basis which encodes the key $Z = \{|L\rangle, |R\rangle\}$, and 10% for the state $|D\rangle$ in the check basis $X = \{|D\rangle, |A\rangle\}$. At the receiver, thus from Bob's point of view, the basis selection probabilities are: 50 % for Z and 50% for X. Finally, assuming $\mu_1 > \mu_2$, the decoy probabilities are: 70% for μ_1, and 30% for μ_2. The decoy intensities are $\mu_1 = 0.4699$ and $\mu_2 = 0.1093$ photons per pulse.

The schematic representation of the apparatus is shown in Figure 3 and Figure 2. On Alice side, to practically generate pulses we employ a laser in the gain-switching regime: such strategy allows to produce pulses of light of the order of picoseconds, in our case we generate pulses of 30ps.

Moreover, we need to perform phase randomization, thus our apparatus must be designed in such a way that the cavity of the the laser is empty every time we generate a pulse, otherwise correlations between phases may occur. In practice, a FPGA keeps the laser close to the limit of its threshold, but still not in stimulated emission, in order to provide both a fast response and phase randomization.

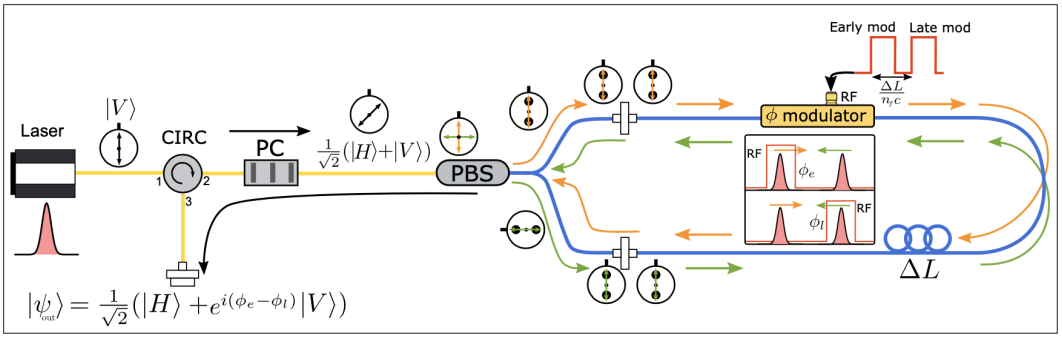


Figure 3: Schematic representation of a POGNAC. A linearly polarized laser passes through an optical circulator and a PC. The pulse encounters a PBS: the vertically polarized component travels in the clockwise direction while the horizontally polarized component in the counter-clockwise. Both pass through a phase modulator and a delay line, but they are subjected to two different phase shifting: ϕ_{cw} and ϕ_{ccw}. Source Agnesi et al. [2019].

Intensity modulation is realized by means of LiNbO3 based phase modulators, since such material is characterized by wide transmission window, and low optical loss at telecom wavelengths. The refraction index depends on the voltage applied between the electrodes on the modulator: for weak voltages the relation is linear both in the ordinary and in the extraordinary axis, in which two different linear relations are respected with two coefficients $\alpha \neq \beta$. In practice, a traditional Mach-Zehnder modulator is not stable if subjected to temperature variations or mechanic stress, thus we choose a Sagnac interferometer amplitude modulator. In Figure 2 we represent schematically how such mechanism is made: the pulses are splitted by a Beam Splitter (BS), then one component travels in counter-clockwise direction while the other in clockwise. The first encounters a modulator which introduces a phase, while the second passes through a fiber delay line, and we program the modulator in such a way that when the clockwise signal arrives it is turned off. In this way, the two signals follow the same path, but the counter-clockwise has a phase shift due to the modulator, thus we can control how such pulses interfere once they encounter each other again in the BS. We should point out that in order to reduce errors, high time precision in the modulator switching is required and that it is useful to choose a 30-70% BS, since this choice allows to work with high intensities without introducing a too high error.

Moreover, we need to modulate the polarization of the pulse in order to actually generate the states. To do so we employ the POGNAC, a polarization modulator based on a LiNbO3 phase modulator inside a Sagnac interferometer Agnesi et al. [2019], in Figure 3 we provide a schematic description of the optical apparatus. A linearly polarized laser enters the optical circulator, then it passes through a Polarization Controller (PC) which generates the state $|\psi\rangle^{in} = 1/\sqrt{2}(|H\rangle + e^{i\phi_0}|V\rangle)$. After this process, the pulse encounters a Polarization Beam Splitter (PBS): the vertically polarized component travels in the clockwise direction while the horizontally polarized component in the counter-clockwise. The first passes through a phase modulator which introduces a phase ϕ_{cw} and then in a delay line, while the second follows the opposite path and it is set in such a way that the modulator introduces a different phase ϕ_{ccw}. When the two pulses converge again in the PBS, the output is given by the state:

$$\psi^{out} = \frac{1}{\sqrt{2}}\left[|H\rangle + e^{i(\phi_{cw}-\phi_{ccw}-\phi_0)}|V\rangle\right]. \tag{18}$$

Thus, by fixing the different phase introduced by the phase modulator we can control the final state. In

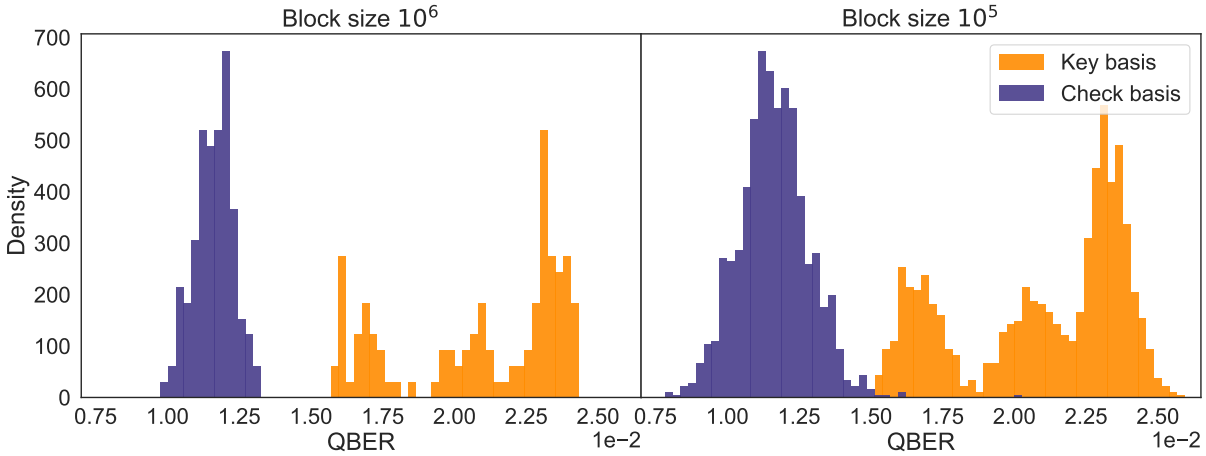


Figure 4: Distributions of QBER in the key and check basis for block sizes 10^5 and 10^6. The distribution in the check basis is gaussian and follows a regular path, while the distribution in the key basis exhibits different maxima. This behaviour can be explained if we remind that the key basis was the only one subjected to a modification in the transmission channel.

Block size	$< QBER_X >$	$< QBER_Z >$	$< QBER_Z^a >$	$< QBER_Z^b >$	$< QBER_Z^c >$
10^6	0.0116	0.02096	0.0167	0.0206	0.0234
10^5	0.0116	0.02098	0.0168	0.0206	0.0233

Table 1: Mean values observed for the QBER in both basis and for two different block sizes. The division in the key basis is carried out according to the following criteria: $a : QBER_Z < 0.019$, while $b : 0.019 \leq QBER_Z < 0.022$ and $b : 0.022 \leq QBER_Z$. We conclude that the QBER in the check basis is lower in average.

particular, given $\Delta\phi = \phi_{cw} - \phi_{ccw}$ we have:

$$\begin{aligned} &\Delta\phi = 0 \Rightarrow \psi^{out} = |D\rangle\,, \\ &\Delta\phi = \pi/2 \Rightarrow \psi^{out} = |L\rangle\,, \\ &\Delta\phi = -\pi/2 \Rightarrow \psi^{out} = |R\rangle\,, \end{aligned} \tag{19}$$

which are the three states employed in our protocol. After the POGNAC, Alice's side is completed by employing an attenuator, that allows to transmit less than one photon per pulse, and a filter.

For what concerns Bob's side, we first employ a filter, a BS with 50% probability and finally a sequence with APC and PBS for each branch, in order to measure the single state by means of single photon detectors.

3.2 Decoding

In practice the procedure generates a set of binary files, each of which contains key blocks of different lengths. A file is made up by a sequence of 8 bytes that code for a uint64 big-endian, which is the length N of the following block, and the N bytes of raw keys. Each block represents 1s of acquisition for the QKD protocol, such approach allows to measure the time necessary to generate each key once the block size of the raw key is fixed. Indeed, since each byte corresponds to an element of the key, i.e. to a quantum

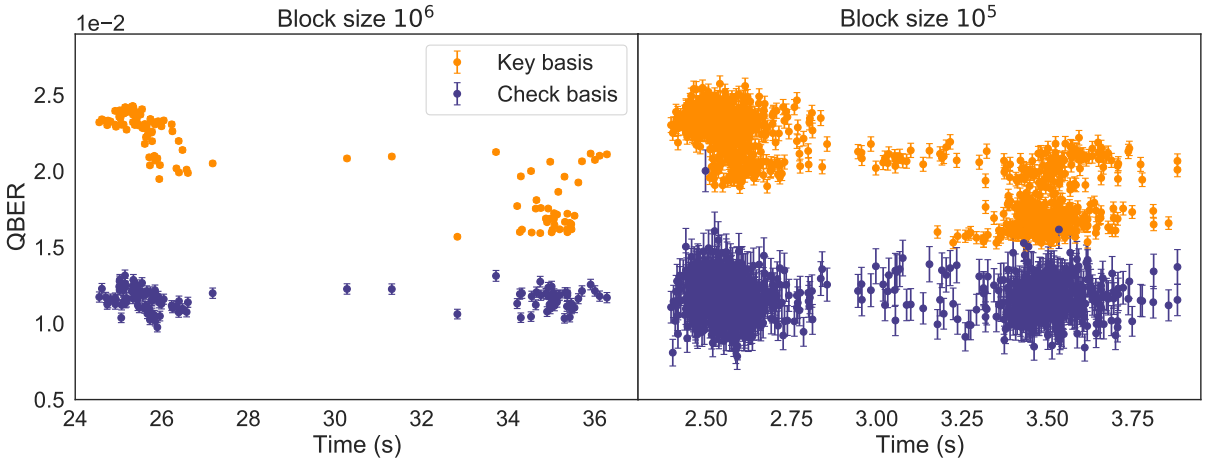


Figure 5: Relation between $QBER_X$ and $QBER_Z$ on the time necessary to generate a raw key of length n_Z, both for $n_Z = 10^5$ and $n_Z = 10^6$. We observe two clusters, each of which refers to a specific status of the apparatus, indeed it is possible that the number of pulses per second emitted by the source changed during data collection.

state shared on the channel, we assume that the time necessary to operate with each state as $t_i = 1/N$, being N the number of bytes per seconds transmitted in that block.

3.3 Analysis

Let us consider the protocol described in Section 2.2.1, our analysis is carried out as described in Section 2.2.2. We compute all the quantities involved with the corresponding error, as discussed in Appendix A. The choice of parameters is justified by the literature: we set the secrecy and correctness parameters to $\epsilon_{sec} = 10^{-9}$ and $\epsilon_{cor} = 10^{-15}$. While, for what concerns λ_{EC}, it should be set to the size of the information exchanged during error-correction. In practice it can be related to the error-correction efficiency f_{EC} as follows:

$$\lambda_{EC} = f_{EC}\ h(e_{obs}), \tag{20}$$

where e_{obs} is the average of the observed error rates in the key basis, and we fix $f_{EC} = 1.16$ as it is proposed in Rusca et al. [2018].

For what concerns the evalutation of the $s^{low}_{Z,0}$ we must observe that the results are often negative. This is due to the finite key effect, and to the specific form of the bounds proposed in Rusca et al. [2018]: even considering block size is of orders $10^6, 10^7$ the bounds provided are not tight enough in the case of zero photon events. When such situations occur we just fix $s^{low}_{Z,0} = 0$. Before entering into the discussion of the results it is also worth to highlight that the apparatus was subjected to a worsening of the quality of the channel for what concerns the key basis. The hypothesis that a situation like this could subsist arouse during the analysis, and it has been later confirmed, we will discuss the consequences of such change in the quality of the signal by studying the results.

3.3.1 QBER

As we discussed, the QBER can be computed with Eq. (17) and error given by Eq. (31). Let us first consider the distributions of QBER both for X and Z basis in the cases of $n_Z = 10^5$ and $n_Z = 10^6$ reported in Figure 4. The main observation is that, for both the considered block sizes, the distribution in the check

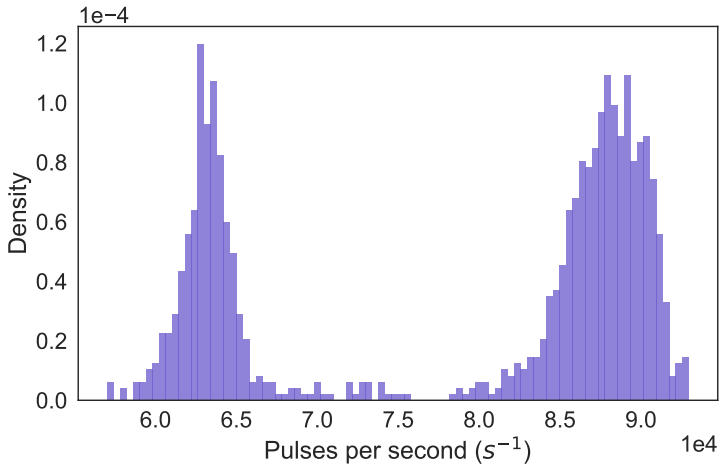


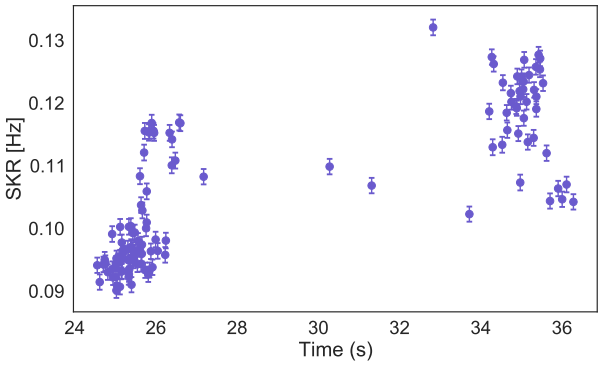
Figure 6: Distribution of pulses per second generated by the apparatus for each key in the case of $n_Z = 10^5$. Data are divided into two main clusters, thus time needed to generate a key depends on the status of the apparatus at the time the key was generated.

basis is normal and follows a regular path, while the distribution for what concerns the key basis exhibits different maxima in correspondence of various values. This behaviour can be explained if we remind that the key basis was the only one subjected to a modification in the transmission channel. In Table 3.1 we report the mean values observed for the QBER, to compute such quantity for the key basis we divided the dataset into three parts in order to observe the drift due to the channel modification. The division was carried out according to the following criteria: $a : QBER_Z < 0.019$, while $b : 0.019 \leq QBER_Z < 0.022$ and $b : 0.022 \leq QBER_Z$. We also provide the overall average in the key basis, even though such quantity must be considered in reference to the shape of the distributions. In general, we can conclude that the QBER in the check basis is lower in average if compared with the same quantity in the key basis.

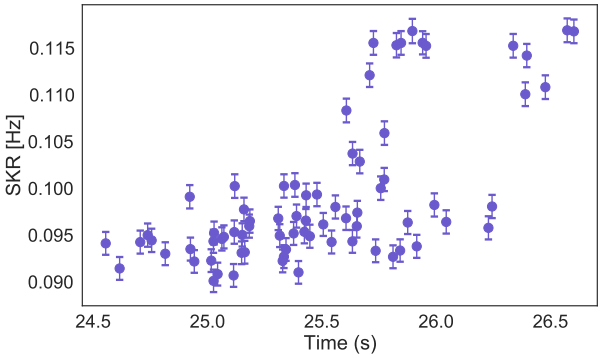
The same behaviour can be observed if we consider Figure 5, in which we provide the relation between QBER and the time necessary to generate a raw key of length n_Z. The two clusters that it is possible to observe are related to two different parts of the dataset: the last part of the data collected is characterized by a higher number of pulses per seconds for each key, thus by a lower time at the same QBER. In Figure 6 we report the distribution of pulses per second at which every key was generated: it is clear that data are divided into two main clusters. As a consequence the time needed to generate a key with a given n_Z depends on the status of the apparatus at that point. Such observation is coherent with what we can retrieve from Figure 5: the QBER of the key basis is in average larger than the QBER on the check basis, while for each of the two basis we observe two clusters according to a different behaviour of the apparatus.

3.3.2 Secret Key Rate (SKR)

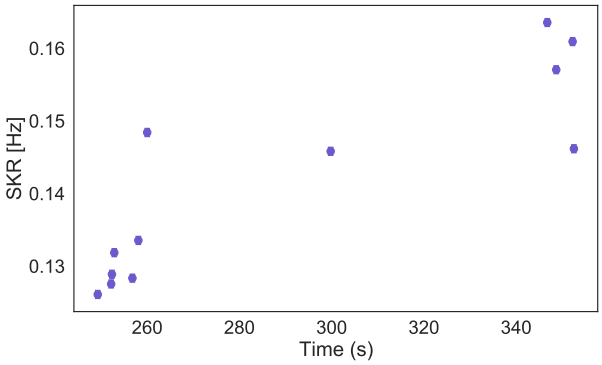
In this section we discuss the SKR, computation is carried out my means of formulas discussed in Section 2.2.2 with error propagation as discussed in Appendix A. In Figure 7(a) we provide the relation between SKR and the time needed to compute the corresponding key for a fixed block length $n_Z = 10^6$. As we already observed in the previous section, the dataset can be sliced into two main parts, each of which is characterized by a different rate of pulses per seconds, thus by a different interval of time needed to generate the key. In general we observe that SKR exhibits an increase in time, as it is also possible to deduce from Figure 7(b). In Figure 7(c) we provide the same results for what concerns keys of block size



((a)) SKR vs time for all the keys generated with $n_Z = 10^6$



((b)) SKR vs time for the second half of keys generated with $n_Z = 10^6$



((c)) SKR vs time for keys generated with $n_Z = 10^7$

Figure 7: SKR as a function of time in the case of a block length $n_Z = 10^6$ and $n_Z = 10^7$. In particular, points are clustered in two main groups, each of which corresponds to a different part of the dataset. In general, SKR exhibits an increase in time.

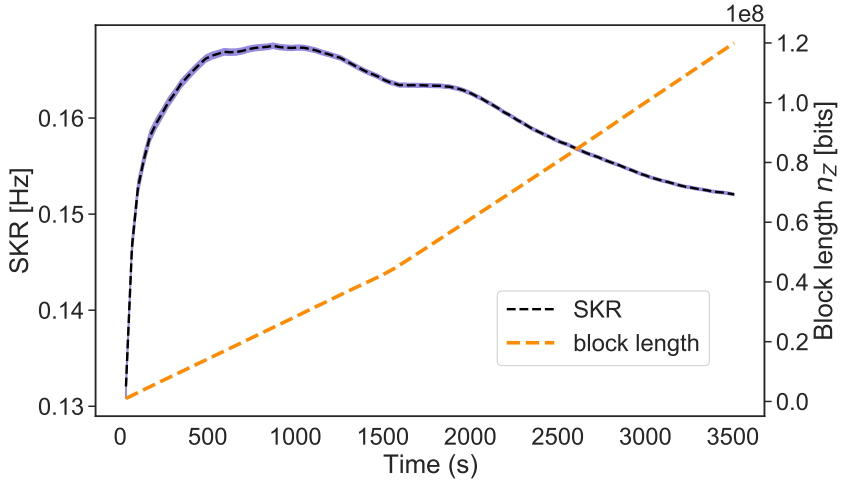
$n_Z = 10^7$. We observe the same trend as in the case of a 10^6 key, even though the order of magnitude of both SKR and time is a factor 10 higher than in the previous case. In general, we conclude that fixing the block length we are dealing with the effects of the evolution of the apparatus. Indeed, each key is generated considering a different slice of the dataset.

A further analysis of the relation between time and the SKR can be given if we avoid to fix the block length of the key. In Figure 8(a) we show, for different block sizes n_Z, thus for different timings, the SKR. In particular, we observe that SKR increases quickly until 874s, i.e. with $n_Z = 24769022 \sim 10^8$ bits, then the trend is inverted and the curve exhibits a slow decrease. Moreover, in Figure 8(b) we show that the relative error on the computation of SKR decreases as the size of the key increases, even though it still remains beyond 1% for all the collected data. It is worth to highlight, as we already observed, that the finite key effect can affect the computation of the $s_{Z,0}$ bounds, since we observed that even for block length of orders $10^6 - 10^7$ the bounds proposed in Rusca et al. [2018] lead to negative results. To take into account such effects and avoid non-physical observations we floored at 0 all the negative results of this kind. Therefore, it is possible that the behaviour of the curve for low block lengths can be also connected with such approximation.

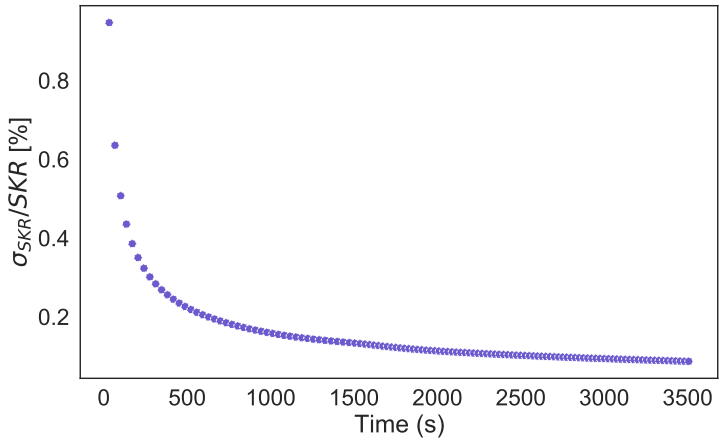
4 Conclusions

To conclude, we discussed and tested the implementation of the 1-decoy 3-states QKD protocol proposed in Rusca et al. [2018]. In particular, we had to deal with the finite key effect in order to compute the security bounds for SKR and provide an estimation of the QBER.

For what concerns the computation of the QBER we also observed that the channel transmitting the key basis states was subjected to a modification during data collection, leading to a drift in the QBER. Moreover, by plotting the QBER computed in correspondence of different times and the number of pulses per seconds, we deduced that the dataset is divided into two main parts, each of which was characterized by a different number of pulses per second in average. Finally we studied the relation between time and the SKR both for fixed block length, with reference to different part of the dataset, and for different times and block length without fixing n_Z. In the first case we still observed the effect of changes in the apparatus, while the second approach allows to get a more detailed explanation of the relation between time and SKR.



((a)) On the left axis we show SKR vs the time needed to generate the key, on the right axis the corresponding block size if plotted.



((b)) The relative error on the computation of SKR decreases as the size of the key increases

Figure 8: SKR increases with time until 874s, i.e. $n_Z = 24769022$ bits, then the trend is inverted and the curve exhibits a slow decrease. Moreover, the relative error on the computation of SKR decreases as the size of the key increases.

A Propagation of errors

Here we present how to compute errors on the quantities involved. Notice that the error in all the following formulas is the propagation from the only measurement error on counting, which is assumed to be poissonian: for a count N we fix $\sigma_N = \sqrt{N}$. Errors on observations and on the finite key correction read:

$$\sigma_{n_Z} = \sqrt{\sigma^2_{n_{Z,\mu_1}} + \sigma^2_{n_{Z,\mu_2}}}, \tag{21}$$

$$\sigma_{n^{\pm}_{Z,k}} = \frac{e^k}{p_k}\sqrt{\sigma^2_{n_{Z,k}} + \frac{\log(19/\epsilon_{sec})}{8n_Z}\sigma^2_{n_Z}}, \tag{22}$$

with trivial extension to the computation of errors. Errors on the photon detection bounds follows:

$$\sigma_{s^{low}_{Z,0}} = \frac{\tau_0}{\mu_1 - \mu_2}\sqrt{\mu_1^2\sigma^2_{n^-_{Z,\mu_2}} + \mu_2^2\sigma^2_{n^+_{Z,\mu_1}}}, \tag{23}$$

$$\sigma_{s^{low}_{Z,1}} = \frac{\tau_1\mu_1}{\mu_2(\mu_1-\mu_2)}\sqrt{\sigma^2_{n^-_{Z,\mu_2}} + \left(\frac{\mu_2^2}{\mu_1^2}\right)^2\sigma^2_{n^+_{Z,\mu_1}} + \left(\frac{\mu_1^2-\mu_2^2}{\mu_1^2\tau_0}\right)^2\sigma^2_{s^{up}_{Z,0}}}, \tag{24}$$

$$\sigma_{s^{up}_{Z,0}} = 2\sqrt{\tau_0^2\sigma^2_{m^+_{Z,k}} + \frac{\log(19/\epsilon_{sec})}{8n_Z}\sigma^2_{n_Z}}. \tag{25}$$

Assuming error only on the parameters b, c, d, error on γ reads:

$$\begin{aligned}\sigma_\gamma = \Bigg[&\sigma_b^2\left[\frac{(2b-1)(c+d)\left(\log\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)-1\right)}{2cd\log 2\sqrt{\frac{(c+d)(1-b)b}{cd}\log\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)}}\right]^2 + \\ &+\sigma_c^2\left[\frac{(b-1)b\left(\log\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)+1\right)}{2c^2\log 2\sqrt{\frac{(c+d)(1-b)b}{cd}\log\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)}}\right]^2 + \\ &+\sigma_d^2\left[\frac{(b-1)b\left(\log\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)+1\right)}{2d^2\log 2\sqrt{\frac{(c+d)(1-b)b}{cd}\log\left(\frac{c+d}{cd(1-b)b}\frac{21^2}{a^2}\right)}}\right]^2\Bigg]^{1/2}\end{aligned} \tag{26}$$

Error on the phase error follows.

$$\sigma_{\phi^{up}_x} = \sqrt{\sigma^2_{\frac{v^{up}_{X,1}}{s^{low}_{X,1}}} + \sigma^2_\gamma} \tag{27}$$

where,

$$\sigma_{\frac{v^{up}_{X,1}}{s^{low}_{X,1}}} = \sqrt{\left(\frac{\sigma_{v^{up}_{X,1}}}{s^{low}_{X,1}}\right)^2 + \left(\frac{v^{up}_{X,1}}{{s^{low}_{X,1}}^2}\sigma_{s^{low}_{X,1}}\right)^2}, \tag{28}$$

and:

$$\sigma_{v^{up}_{X,1}} = \frac{\tau_1}{\mu_1-\mu_2}\sqrt{\sigma^2_{m^+_{X,\mu_1}} + \sigma^2_{m^-_{X,\mu_2}}}. \tag{29}$$

Finally, we can now compute the error on the secret key length as follows:

$$\sigma_\ell = \sqrt{\sigma^2_{s^{low}_{z,0}} + \sigma^2_{s^{low}_{z,1}}(1 - h(\phi^{up}_Z))^2 + \sigma^2_{h(\phi^{up}_Z)} {s^{low}_{z,1}}^2}. \tag{30}$$

Moreover, errors on QBER and SKR can be computed as follows:

$$\sigma_{QBER_z} = \sqrt{\frac{\sigma^2_{m_Z}}{n^2_Z} + \frac{{m_Z}^2 \sigma^2_{n_Z}}{n^4_Z}}, \tag{31}$$

$$\sigma_{SKR} = \sqrt{\left(\frac{\sigma_\ell}{N_{tot}} R\right)^2 + \left(\frac{\ell}{N^2_{tot}} \sigma_{N_{tot}} R\right)^2}. \tag{32}$$

References

C. H. Bennett and G. Brassard. *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore*, page 175–179, 1984.

Erik Stock. Self-organized quantum dots for single photon sources. 01 2011. doi: 10.14279/depositonce-2721.

B. Huttner, N. Imoto, N. Gisin, and T. Mor. Quantum cryptography with coherent states. *Phys. Rev. A*, 51:1863–1869, Mar 1995. doi: 10.1103/PhysRevA.51.1863. URL `https://link.aps.org/doi/10.1103/PhysRevA.51.1863`.

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. doi: 10.1063/1.5023340. URL `https://doi.org/10.1063/1.5023340`.

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. doi: 10.1103/PhysRevA.89.022307. URL `https://link.aps.org/doi/10.1103/PhysRevA.89.022307`.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL `https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830`.

Costantino Agnesi, Marco Avesani, Andrea Stanco, Paolo Villoresi, and Giuseppe Vallone. All-fiber self-compensating polarization encoder for quantum key distribution. *Optics Letters*, 44(10):2398, May 2019. ISSN 1539-4794. doi: 10.1364/ol.44.002398. URL `http://dx.doi.org/10.1364/OL.44.002398`.