# Lab 03 report: QKD

## 3-state 1-decoy efficient BB84 implementation & security analysis

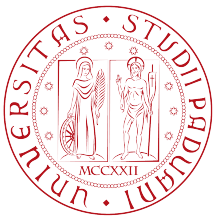
Author: **David Polzoni**

Student ID: **2082157**

Course: **Quantum Cryptography & Security**

Prof. **G. Vallone**, Prof. **N. Laurenti**

Università degli Studi di Padova




A.Y. 2023-2024

Date: February 4, 2024

# Contents

# 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 the two parties. 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 by Bennett and Brassard [1], 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 [1]. We discuss in particular the protocol parameters and the generic steps of post-processing.

## 2.1 BB84 protocol

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

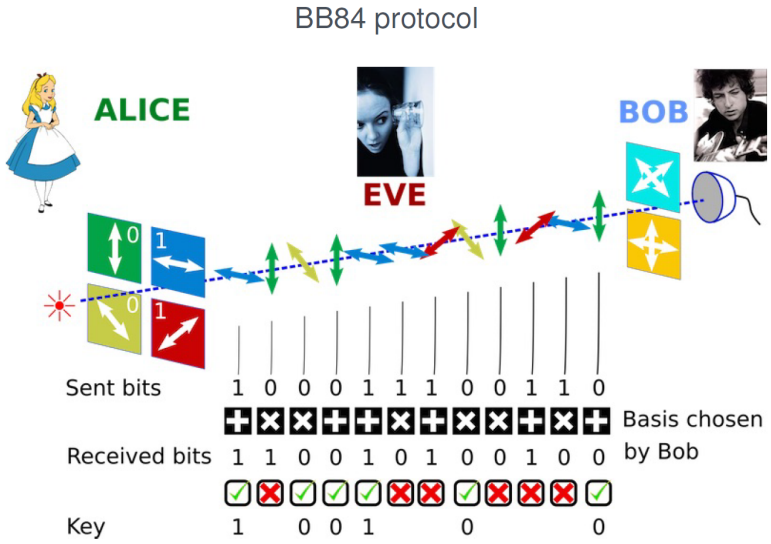



Figure 1: Schematic representation of BB84. Alice sends a photon encoded in a specific basis to Bob, who chooses a basis and performs a measurement. Then, Alice and Bob broadcast their measurement bases on the classical channel: if they chose different bases they discard the bit.

The next step is usually referred as *sifting*: Alice and Bob broadcast their measurement bases on the classical channel. If they chose different bases they just discard the bit, while the remaining bits compose the so called *sifted keys* $k_A^S$ and $k_B^S$ 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 $k_A''^S = k_B''^S$. The main steps of post-processing are:

1. **Information reconciliation**: sharing information over the public channel Alice and Bob produce two equal keys $k_A'^S = k_B'^S$, by leaking the less possible information to Eve;

2. **Error verification**: after information reconciliation we can assume that $k_A'^S = k_B'^S$ 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 which are independent from Eve.

### 2.1.1 Intercept & Resend (I&R) attack and error rate

The advantage of the quantum protocol is particularly evident once we take into account Eve's 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 the Secret Key Rate (SKR), a quantity which shows the amount of secure bits available, thus the level of compression needed to make the key private. The latter can be mathematically expressed as follows:

$$r = \max\{I_{AB} - I_E, 0\} \tag{1}$$

where $I_{AB} \equiv I(A, B)$ represents the mutual information between Alice and Bob, and it can be computed in the subsequent way:

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

where $H$ encodes the classical Shannon's entropy, $h_2$ represents classical binary Shannon's entropy, and $Q$ is referred to the QBER.

### 2.1.2 Generalizations and improvements

BB84 protocol and I&R attack serve as straightforward yet illustrative examples; however, 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 $Z$ and $X$ with biased probabilities $P_Z \approx 1$ and $P_X = 1 - P_Z \approx 0$. In this way, the $X$ basis 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:

$$|\alpha\rangle = \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} e^{-\frac{\mu}{2}} |n\rangle \tag{3}$$

where $\mu$ represents the mean photon number (MPN) associated to the laser intensity, and $|n\rangle$ encodes the eigenstates of the number operator. More generally, we fix $\mu$ and we randomize the phase, thus the emitted state is described by a Poisson distribution:

$$\rho = \sum_{n=0}^{\infty} P_\mu(n) |n\rangle \langle n| \,, \text{ where } P_\mu(n) = \frac{\mu^n}{n!} e^{-\mu} \tag{4}$$

However, it is important 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 used for a highly efficient form of attack known as Photon-Number-Splitting (PNS) (refer to Huttner et al. [2] for more 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 numbers, 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 examine the 3-state 1-decoy efficient BB84 protocol as analyzed in Rusca et al. [3]. The fundamental concept involves the usage of decoy states for attack detection, with standard BB84 states exclusively employed for key generation. Let us consider an efficient BB84, in which two bases $Z$ and $X$ have asymmetric probabilities $P_Z \approx 1$ and $P_X = 1 - P_Z \approx 0$. 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 $\mu_1$ or $\mu_2$. 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 $y_i'$. 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: $Z_k = \{i : a_i = b_i = Z,\ k \in K\}$ and $X_k = \{i : a_i = b_i = X,\ k \in K\}$. Then, Alice and Bob generate the final raw key of fixed post-processing block size $n_Z = \sum_k n_{Z,k}$, where $n_{Z,k}$ is the cardinality of the corresponding set: $|Z_k| = n_{Z,k}$. Note that, we are considering the procedure exposed in Lim et al. [4], assuming that both the intensity levels are employed for the key generation, while typically QKD protocols use only one decoy. After this process, Alice and Bob are both able to compute the number of bit errors in each basis $m_{Z,k}$ which is a fundamental quantity to provide security bounds. Indeed, in the next section, we will 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 $\lambda_{\mathrm{EC}}$ bits of information.

### 2.2.2 Security bounds

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

$$l \leq s^l_{z,0} + s^l_{z,1}(1 - h_2(\phi^u_z)) - \lambda_{\mathrm{EC}} - a \log_2\left(\frac{b}{\epsilon_s}\right) - b \log_2\left(\frac{2}{\epsilon_c}\right) \tag{5}$$

where $s^l_{z,0}$ and $s^l_{z,1}$ are the lower bounds to vacuum and single photon events, and $h_2(x) = x \log_2(x) - (1 - x) \log_2(1 - x)$ is the binary entropy. Moreover, $\phi^u_z$ is the upper bound on the phase error rate, and $\lambda_{\mathrm{EC}}$ 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. [4]). In our specific case, we assume $a = 6$ and $b = 19$. Let us now express explicitly the quantities reported in equation 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 provides the following bound on the number of observations:

$$|n^*_{Z,k} - n_{Z,k}| \leq \sqrt{\frac{n_Z}{2} \log\left(\frac{1}{\epsilon_1}\right)}, \text{ with probability } 1 - 2\epsilon_1 \tag{6}$$

where $n^*_{Z,k}$ and $n_{Z,k}$ are the number of detection in the asymptotic (infinite) scenario and in the finite key one, for each decoy intensity $k$, while $n_Z$ is the total number of observations. Considering a different error $\epsilon_2$, the same inequality can be applied to bound the error rate:

$$|m^*_{Z,k} - m_{Z,k}| \leq \sqrt{\frac{m_Z}{2} \log\left(\frac{1}{\epsilon_2}\right)}, \text{ with probability } 1 - 2\epsilon_2 \tag{7}$$

Thus, based on the earlier definitions, we can define the following quantities, that will be useful for the next computations:

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

Moreover, it is possible to show (Rusca et al. [3], Lim et al. [4]) that by fixing $\epsilon_1 = \epsilon_2 = \epsilon$ we obtain $\epsilon_s = 19\epsilon$. From now on, we will always assume $\epsilon_s = 19\epsilon$. 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:

$$s_{z,0} \geq s^l_{z,0} = \frac{\tau_0}{\mu_1 - \mu_2}\left(\mu_1 \, n^-_{Z,\mu_2} - \mu_2 \, n^+_{Z,\mu_1}\right) \tag{9}$$

$$s_{z,0} \leq s^u_{z,0} = 2\left(\tau_0 \, m^+_{Z,k} + \sqrt{\frac{n_Z}{2} \log\left(\frac{1}{\epsilon_1}\right)}\right) \tag{10}$$

$$s_{z,1} \geq s^l_{z,1} = \frac{\tau_1 \, \mu_1}{\mu_2 \, (\mu_1 - \mu_2)}\left(n^-_{Z,\mu_2} - \frac{\mu_2^2}{\mu_1} \, n^+_{Z,\mu_1} - \frac{\mu_1^2 - \mu_2^2}{\mu_1^2} \, \frac{s^u_{z,0}}{\tau_0}\right) \tag{11}$$

where the term $\tau_n$ represents the probability that Alice sends a $n$-photon state. The latter can be mathematically expressed as follows:

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

Finally, the phase error can be estimated in the subsequent way:

$$\phi_z \leq \phi_x^u = \frac{v_{x,1}^u}{s_{x,1}^l} + \gamma\left(\epsilon_s, \frac{v_{x,1}^u}{s_{x,1}^l}, s_{z,1}^l, s_{x,1}^l\right) \tag{13}$$

where $\gamma(a, b, c, d)$ is defined as follows:

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

Indeed, $v_{x,1}$ represents the number of error detected for a single photon event, which upper bound can be outlined as:

$$v_{x,1} \geq v_{x,1}^u = \frac{\tau_1}{\mu_1 - \mu_2}\left(m_{X,\mu_1}^+ - m_{X,\mu_2}^-\right) \tag{15}$$

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 (equation 5), the secret key rate (SKR) can be computed as follows:

$$\text{SKR} = \frac{l}{N_{tot}}R \tag{16}$$

where $N_{tot}$ is the total number of sent pulses in order to have a key of block size $n_Z$, and $R$ is the repetition rate of the source, that we assume to be 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:

$$\text{QBER}_Z = \frac{m_Z}{n_Z} \tag{17}$$

with trivial analogous on the $X$ basis as:

$$\text{QBER}_X = \frac{m_X}{n_X} \tag{18}$$

### 2.2.3 Errors propagation

Here, we will present how to compute errors on the quantities involved in the previous sections. 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 $\sigma_N = \sqrt{N}$. Let's begin our analysis with the errors on observations and finite key correction:

$$\sigma_{n_Z} = \sqrt{\sigma_{n_{Z,\mu_1}}^2 + \sigma_{n_{Z,\mu_2}}^2} \tag{19}$$

$$\sigma_{n_Z}^{\pm} = \frac{e^k}{p_k}\sqrt{\sigma_{n_{Z,k}^2} + \frac{\log(19/\epsilon_s)}{8\,n_Z}\,\sigma_{n_Z^2}} \tag{20}$$

with trivial extension to the errors computation.

Let's now focus on the errors concerning photon detection bounds:

$$\sigma_{s^l_{z,0}} = \frac{\tau_0}{\mu_1 - \mu_2} \sqrt{\mu_1^2 \, \sigma^2_{n^-_{Z,\mu_2}} + \mu_2 \, \sigma^2_{n^+_{Z,\mu_1}}} \tag{21}$$

$$\sigma_{s^u_{z,0}} = 2 \sqrt{\tau_0^2 \, \sigma^2_{m^+_{Z,k}} + \frac{\log(19/\epsilon_s)}{8 \, n_Z} \, \sigma_{n_Z^2}} \tag{22}$$

$$\sigma_{s^l_{z,1}} = \frac{\tau_1 \, \mu_1}{\mu_2 \, (\mu_1 - \mu_2)} \sqrt{\sigma^2_{n^-_{Z,\mu_2}} + \left(\frac{\mu_2^2}{\mu_1^2}\right)^2 \sigma^2_{n^+_{Z,\mu_1}} + \left(\frac{\mu_1^2 - \mu_2^2}{\mu_1^2 \, \tau_0}\right)^2 \sigma^2_{s^u_{z,0}}} \tag{23}$$

Assuming deviations only on $b$, $c$, $d$ parameters, the error on $\gamma$ can be computed as:

$$\begin{aligned} \sigma_\gamma = \Bigg[ & \sigma_b^2 \left[ \frac{(2b-1)(c+d) \, \left( \log\left( \frac{c+d}{cdb(1-b)} \frac{21^2}{a^2} \right) - 1 \right)}{2cd \log(2) \sqrt{\frac{(c+d)(1-b)b}{cd} \log\left( \frac{c+d}{cdb(1-b)} \frac{21^2}{a^2} \right)}} \right]^2 \\ & + \sigma_c^2 \left[ \frac{(b-1)b \, \left( \log\left( \frac{c+d}{cdb(1-b)} \frac{21^2}{a^2} \right) + 1 \right)}{2c^2 \log(2) \sqrt{\frac{(c+d)(1-b)b}{cd} \log\left( \frac{c+d}{cdb(1-b)} \frac{21^2}{a^2} \right)}} \right]^2 \\ & + \sigma_d^2 \left[ \frac{(b-1)b \, \left( \log\left( \frac{c+d}{cdb(1-b)} \frac{21^2}{a^2} \right) + 1 \right)}{2d^2 \log(2) \sqrt{\frac{(c+d)(1-b)b}{cd} \log\left( \frac{c+d}{cdb(1-b)} \frac{21^2}{a^2} \right)}} \right]^2 \Bigg]^{\frac{1}{2}} \end{aligned} \tag{24}$$

Therefore, the statistical uncertainty in the phase error can be calculated as follows:

$$\sigma_{\phi^u_x} = \sqrt{\{\sigma^u_{v,s}\}^2 + \sigma^2_\gamma} \tag{25}$$

where $\sigma^u_{v,s}$ is defined in the subsequent way:

$$\sigma^u_{v,s} = \sqrt{\left(\frac{\sigma_{v^u_{x,1}}}{\sigma_{s^l_{x,1}}}\right)^2 + \left(\frac{v^u_{x,1}}{\{s^l_{x,1}\}^2}\right)^2 \sigma^2_{s^l_{x,1}}} \tag{26}$$

while $\sigma_{v^u_{x,1}}$ can be depicted as:

$$\sigma_{v^u_{x,1}} = \frac{\tau_1}{\mu_1 - \mu_2} \sqrt{\sigma^2_{m^+_{X,\mu_1}} + \sigma^2_{m^-_{X,\mu_2}}} \tag{27}$$

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

$$\sigma_l = \sqrt{\sigma^2_{s^l_{z,0}} + \sigma^2_{s^l_{z,1}} \left(1 - h\left(\phi^u_z\right)\right)^2 + \sigma^2_{h(\phi^u_z)} \left\{s^l_{z,1}\right\}^2} \tag{28}$$

Moreover, errors on QBER can be computed in the subsequent way:

$$\sigma_{\mathrm{QBER}_Z} = \sqrt{\frac{\sigma^2_{n_Z}}{n^2_Z} + \frac{m^2_Z \, \sigma^2_{n_Z}}{n^4_Z}} \tag{29}$$

with a trivial extension on the QBER errors considering the $X$ basis. For what concerns the SKR errors, they can be defined as follows:

$$\sigma_{\mathrm{SKR}} = \sqrt{\left(\frac{\sigma^2_l}{N_{tot}} R\right)^2 + \left(\frac{l}{N^2_{tot}} \, \sigma^2_{N_{tot}} R\right)^2} \tag{30}$$

# 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 fibers, which could change the polarization of the photons. As previously discussed, we implement the 1-decoy 3-state protocol described in Rusca et al. [3], in which the authors compare the performances of 1-decoy and 2-decoy approaches. Following the method used by Lim et al. [4], the authors 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 $Z = \{|H\rangle, |V\rangle\}$, and 10% for the state $|D\rangle$ in the check basis $X = \{|D\rangle, |A\rangle\}$. At the receiver, 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 $\mu_1$, and 30% for $\mu_2$. The decoy intensities are $\mu_1 = 0.6$ and $\mu_2 = 0.1818$ photons per pulse. The schematic representation of the apparatus is shown in figure 2. On Alice side, to practically generate pulses, we employ a laser in the gain-switching regime: such strategy allows to produce light pulses of the order of picoseconds (ps). 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. 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 consistent error.

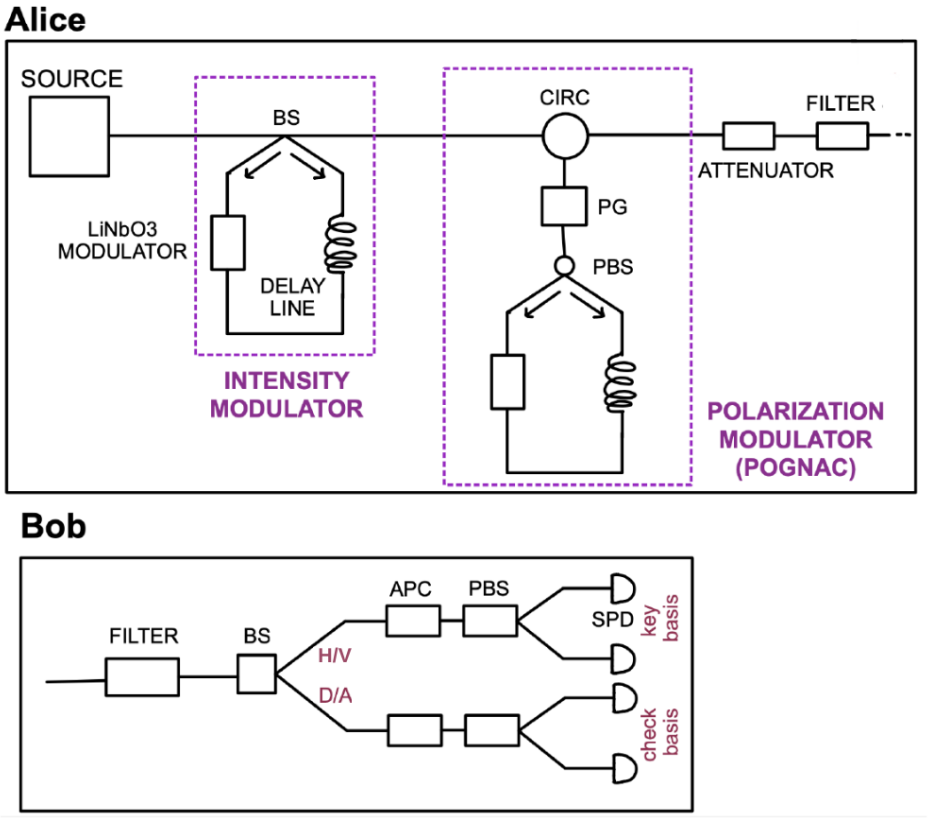



Figure 2: Schematic representation of the apparatus. On Alice side, a source generates pulses, that are then subjected to intensity modulation and polarization modulation, by means of POGNAC. An attenuator and a filter complete Alice's side. On the other hand, Bob's side consists of a filter, a 50% BS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

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. 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 $|\psi\rangle^i = 1/\sqrt{2}\left(|H\rangle + e^{i\phi_0}|V\rangle\right)$. After this process, the pulse encounters a Polarizing Beam Splitter (PBS): the vertically polarized component travels in the clockwise direction while the horizontal one in the counter-clockwise. The first passes through a phase modulator which introduces a phase $\phi_{cw}$ 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 $\phi_{ccw}$. When the two pulses converge again in the PBS, the output is given by the state:

$$|\psi\rangle^o = \frac{1}{\sqrt{2}}\left(|H\rangle + e^{i(\phi_{cw}-\phi_{ccw}-\phi_0)}|V\rangle\right) \tag{31}$$

Thus, by fixing the different phases introduced by the phase modulator we can control the output state. In particular, given $\Delta\phi = \phi_{cw} - \phi_{ccw}$, we have:

$$\begin{aligned} \Delta\phi = 0 &\Rightarrow |\psi\rangle^o = |D\rangle \\ \Delta\phi = \frac{\pi}{2} &\Rightarrow |\psi\rangle^o = |H\rangle \\ \Delta\phi = -\frac{\pi}{2} &\Rightarrow |\psi\rangle^o = |V\rangle \end{aligned} \tag{32}$$

which are the 3 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, then 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.

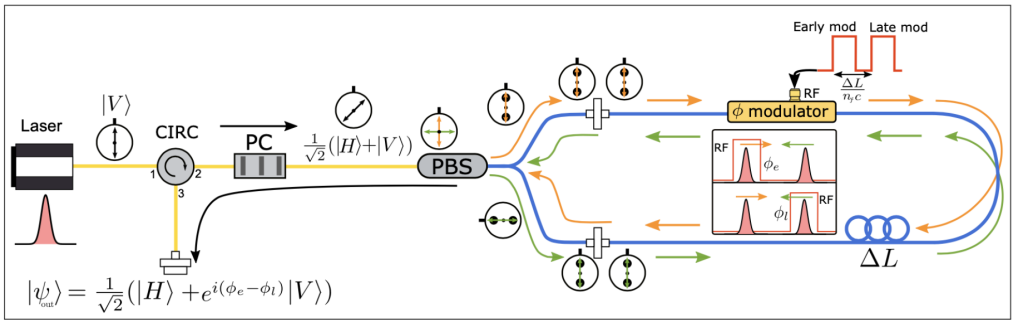



Figure 3: Schematic representation of a POGNAC. A linearly polarized laser passes through an optical circulator and a PC. The pulse encounters a PBS: the vertically polarized component travels in the clockwise direction while the horizontally polarized component in the counter-clockwise. Both pass through a phase modulator and a delay line, but they are subjected to two different phase shifting: $\phi_{cw}$ and $\phi_{ccw}$. This image is extracted from Agnesi et al. [5].

## 3.2 Decoding raw keys

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 $N$ bytes of raw keys. Each block represents one second 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 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.

# 4 Security analysis and results

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 errors, as discussed in section 2.2.3. The parameters' choice is justified by the literature: we set the secrecy and correctness parameters to $\epsilon_s = 10^{-9}$ and $\epsilon_c = 10^{-15}$. While, for what concerns $\lambda_{\mathrm{EC}}$, it should be set to the size of the information exchanged during error correction procedure. In practice, it can be related to the error correction efficiency $f_{\mathrm{EC}}$ as follows:

$$\lambda_{\mathrm{EC}} = f_{\mathrm{EC}}\, h(e_z) \tag{33}$$

where $e_z$ is the average of the observed error rates in the key basis. Also, as proposed in Rusca et al. [3], we fix $f_{\mathrm{EC}} = 1.16$. For what concerns the evaluation of the $s^l_{z,0}$, we have to observe that the results are often negative. That is due to the finite key effect, and to the specific form of the bounds proposed in Rusca et al. [3]: in fact, even considering block sizes of orders $10^6$ and $10^7$, the bounds provided are not tight enough in the case of zero photon events. When such situations occur, we just fix $s^l_{z,0} = 0$.

## 4.1 QBER

As we discussed, the QBER can be computed with equation 17 and its errors are given by equation 29. Let us first consider the distributions of QBER both for $Z$ and $X$ basis in the cases where the block length $n_Z$ is $10^5$ and $10^6$. The results are reported in figure 4: the main observation is that, for both the considered block sizes, the distribution in the check basis is gaussian 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.

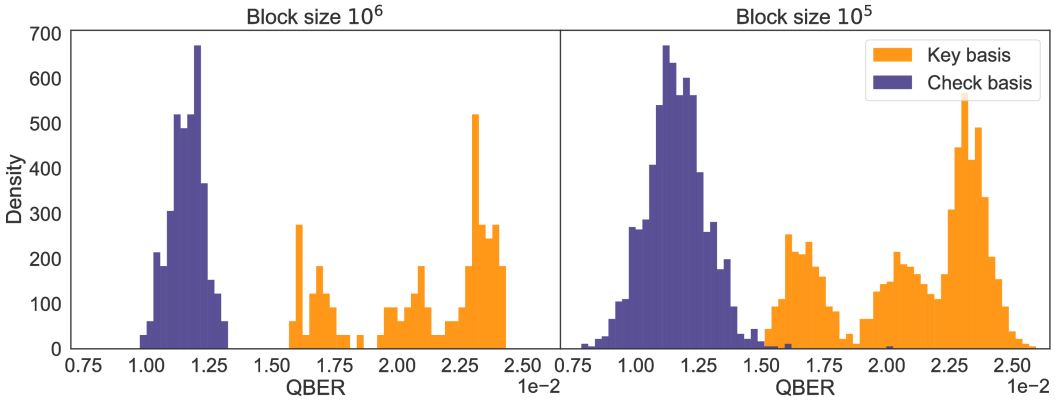



Figure 4: Distributions of QBER in the key and check basis for block sizes $10^5$ and $10^6$. The distribution in the check basis is gaussian and follows a regular path, while the distribution in the key basis exhibits different maxima.

In tables 1 and 2, 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: $a : \mathrm{QBER}_Z < 0.019$, while $b : 0.019 \leq \mathrm{QBER}_Z < 0.022$ and $c : \mathrm{QBER}_Z \geq 0.022$. 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.

| Block size | $< \mathrm{QBER}_X >$ | $< \mathrm{QBER}_Z >$ |
|---|---|---|
| $10^6$ | $0.0116 \pm$ 3e-04 | $0.02094 \pm$ 2e-05 |
| $10^5$ | $0.0116 \pm$ 3e-04 | $0.02096 \pm$ 2e-05 |

Table 1: Mean values observed for the QBER in both basis and for two different block sizes. We conclude that the QBER in the check basis is lower on average.

| Block size | $< \mathrm{QBER}_Z^a >$ | $< \mathrm{QBER}_Z^b >$ | $< \mathrm{QBER}_Z^c >$ |
|---|---|---|---|
| $10^6$ | $0.0167 \pm$ 2e-04 | $0.0206 \pm$ 2e-04 | $0.0234 \pm$ 3e-04 |
| $10^5$ | $0.0168 \pm$ 2e-04 | $0.0206 \pm$ 2e-04 | $0.0233 \pm$ 3e-04 |

Table 2: Division in the key basis according to the criteria: $a : \mathrm{QBER}_Z < 0.019$, $b : 0.019 \leq \mathrm{QBER}_Z < 0.022$, and $c : \mathrm{QBER}_Z \geq 0.022$.

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 fact, considering figure 6, the distribution of pulses per second at which every key was generated is reported: it is clear that data is 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.

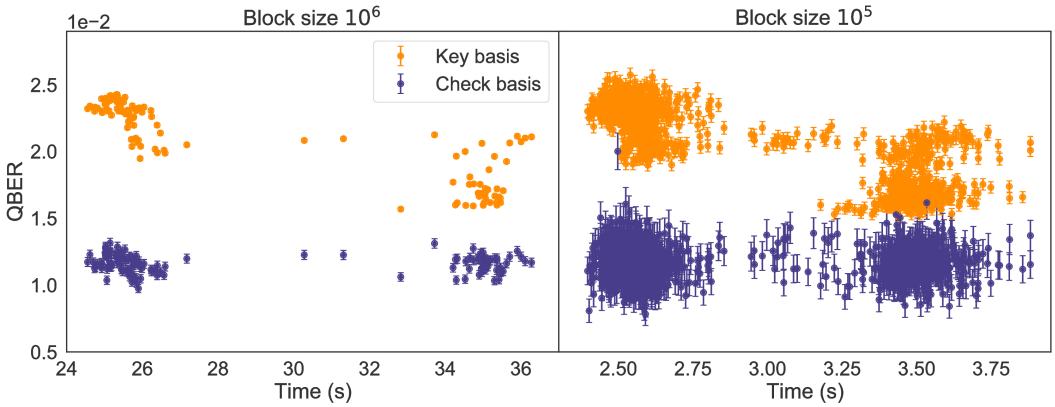



Figure 5: Relation between $\text{QBER}_X$ and $\text{QBER}_Z$ on the time necessary to generate a raw key of length $n_Z$, both for $n_Z = 10^5$ and $n_Z = 10^6$. We observe two clusters, each of which refers to a specific status of the apparatus, indeed it is possible that the number of pulses per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

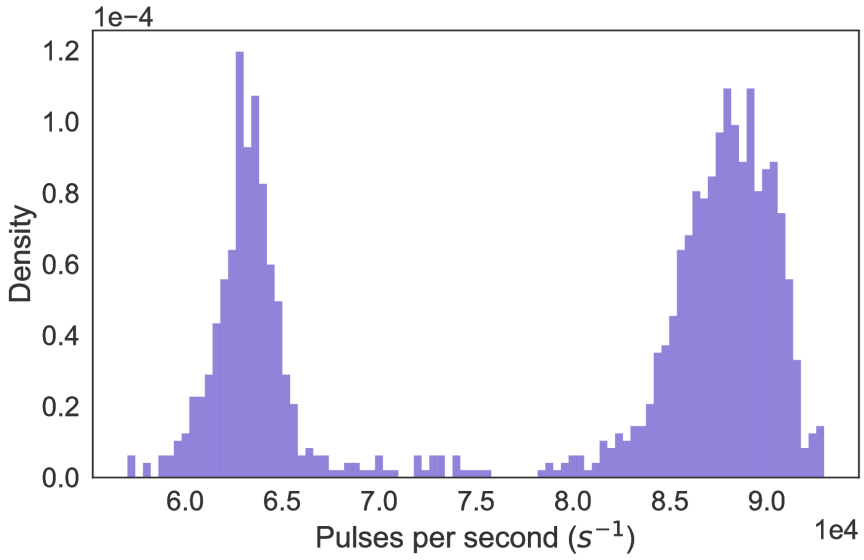



Figure 6: Distribution of pulses per second generated by the apparatus for each key in the case of $n_Z = 10^5$. Data is divided into two main clusters, thus time needed to generate a key depends on the status of the apparatus at the time in which the key was generated.

## 4.2 SKR

In this section, we will discuss the SKR, its computation is carried out by means of equations discussed in section 2.2.2 with error propagation as described in section 2.2.3. To begin our analysis, in figure 7, we report the distribution of the SKR. The latter is noteworthy for its regularity, with the density plot showcasing a consistent pattern. The regularity of the distribution indicates a stable and predictable behaviour in the generation of secret keys under the given conditions. This uniformity in the SKR distribution is a positive characteristic, suggesting reliability and consistency in the QKD process employed. In figure 8a 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 8b. Instead, in figure 8c, we provide the same results for what concerns keys of block size $n_Z = 10^7$. 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, our conclusion is that by maintaining a constant 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.

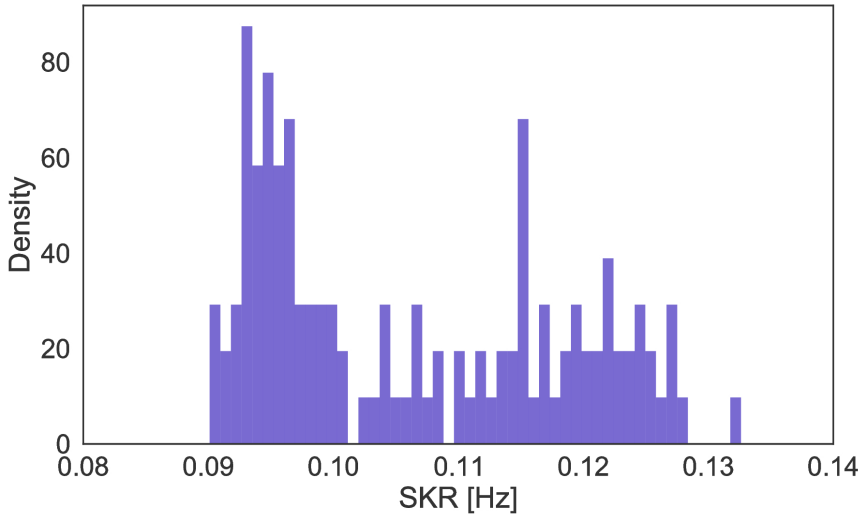
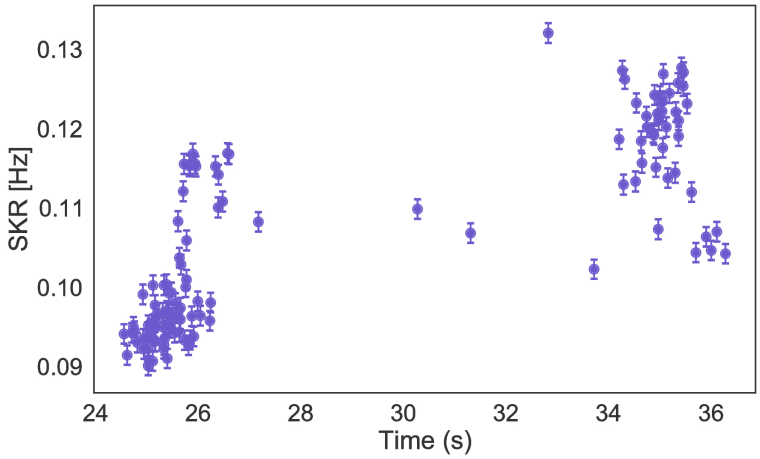
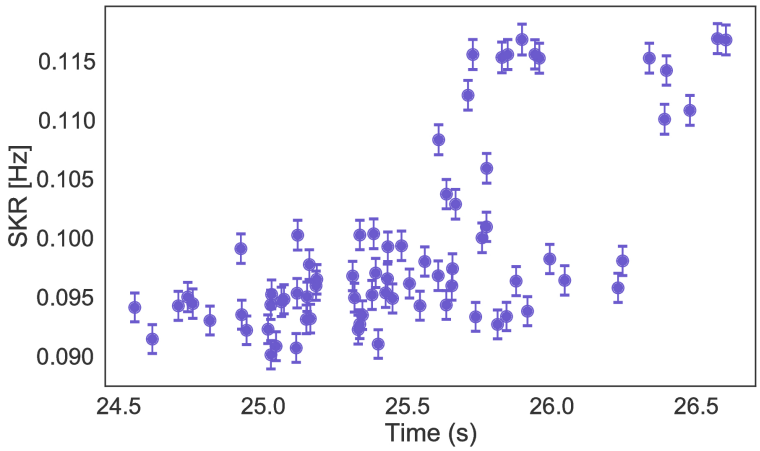


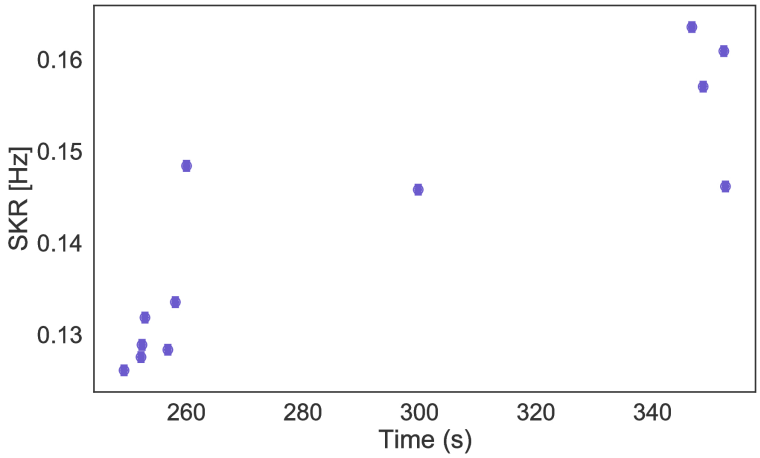

Figure 7: Distribution of SKR for block size $10^5$. The density plot provides a visual representation of the SKR values' distribution, allowing us to discern patterns and characteristics associated with this specific block size.




(a) SKR: all the keys generated with $n_Z = 10^6$.



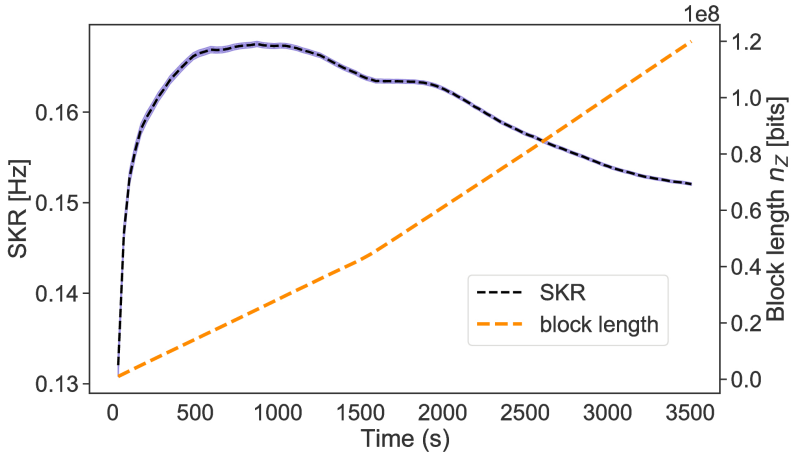

(b) SKR: second half of the keys generated with $n_Z = 10^6$.



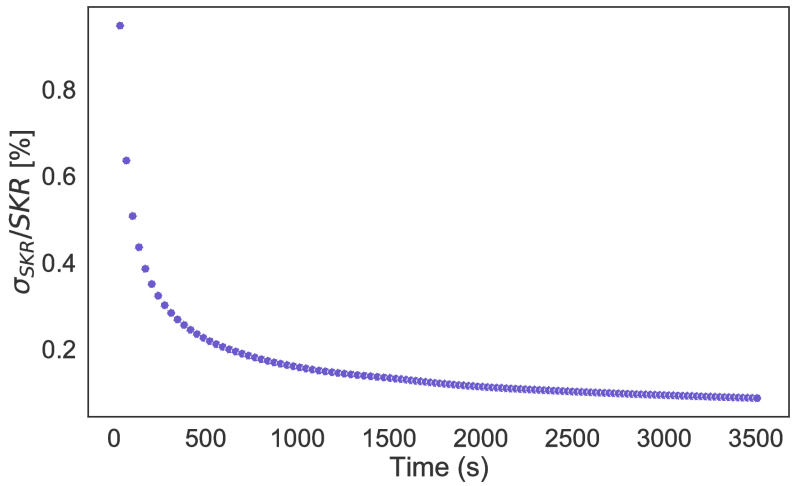

(c) SKR: keys generated with $n_Z = 10^7$.

Figure 8: 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.

A further analysis of the relation between time and SKR can be given if we avoid to fix the block length of the key. In figure 9a we show the SKR for different block sizes $n_Z$ (for different timings). In particular, we observe that it increases quickly until 874 s, i.e. with $n_Z = 24769022 \approx 10^8$ bits, then the trend is inverted and the curve exhibits a slow decrease. Moreover, in figure 9b, 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 important to emphasize, as previously noted, that the finite key effect can impact the calculation of bounds for $s_{z,0}$. Our observations indicate that, even with block lengths on the orders of $10^6$ and $10^7$, the bounds suggested in Rusca et al. [3], result in negative outcomes. Therefore, it is possible that the behaviour of the curve for low block lengths can be also connected with such approximation.




(a) Left axis: SKR over the time to generate the key, right axis: corresponding block size.




(b) Relative error on SKR computation decreases as the size of the key increases.

Figure 9: SKR increases with time until 874 s, i.e. $n_Z \approx 10^8$ 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.

# 5 Conclusions

To conclude, we discussed and tested the implementation of the 3-state 1-decoy QKD protocol proposed in Rusca et al. [3]. In particular, we had to deal with the finite key effects 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 at 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 analyzed the relation between time and 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. If there is an interest in delving into the developed code, it can be found at the following link: `QC&S-Lab-03-QKD-Report`.

## References

[1] C. H. Bennet, and G. Brassard. Quantum cryptography: Public-key distribution and coin tossing. Proceedings IEEE Int. Conf. on Computers, Systems and Signal Processing, Bangalore, India, pp. 175–179, December 1984.
DOI: 10.1016/j.tcs.2014.05.025.
URL: http://dx.doi.org/10.1016/j.tcs.2014.05.025;

[2] B. Huttner, N. Imoto, N. Gisin, and T. Mor. Quantum cryptography with coherent states. Phys. Rev. A, 51:1863-1869, March 1995.
DOI: 10.1103/PhysRevA.51.1863.
URL: https://link.aps.org/doi/10.1103/PhysRevA.51.1863;

[3] D. Rusca, A. Boaron, F. Grünenfelder, A. Martin, and H. Zbinden. Finite-key analysis for the 1-decoy state QKD protocol. Applied Physics Letters, 112(17):171104, April 2018. DOI: 10.1063/1.5023340.
URL: https://doi.org/10.1063/1.5023340;

[4] C. C. W. Lim, M. Curty, N. Walenta, F. Xu, and H. Zbinden. Concise security bounds for practical decoy-state quantum key distribution. Phys. Rev. A, 89:022307, February 2014. DOI: 10.1103/PhysRevA.89.022307.
URL: https://link.aps.org/doi/10.1103/PhysRevA.89.022307;

[5] C. Agnesi, M. Avesani, A. Stanco, P. Villoresi, and G. 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.