



HAL
open science

QOSST: A Highly-Modular Open Source Platform for Experimental Continuous-Variable Quantum Key Distribution

Yoann Piétri, Matteo Schiavon, Valentina Marulanda Acosta, Baptiste Gouraud, Luis Trigo Vidarte, Phillippe Grangier, Amine Rhouni, Eleni Diamanti

► **To cite this version:**

Yoann Piétri, Matteo Schiavon, Valentina Marulanda Acosta, Baptiste Gouraud, Luis Trigo Vidarte, et al.. QOSST: A Highly-Modular Open Source Platform for Experimental Continuous-Variable Quantum Key Distribution. 2024. hal-04563985v2

HAL Id: hal-04563985

<https://hal.science/hal-04563985v2>

Preprint submitted on 2 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

QOSST: A Highly-Modular Open Source Platform for Experimental Continuous-Variable Quantum Key Distribution

Yoann Piétri,¹ Matteo Schiavon,¹ Valentina Marulanda Acosta,^{1,2} Baptiste Gouraud,³
Luis Trigo Vidarte,⁴ Philippe Grangier,⁵ Amine Rhouni,¹ and Eleni Diamanti¹

¹*Sorbonne Université, CNRS, LIP6, F-75005 Paris, France*

²*DOTA, ONERA, Université Paris Saclay, F-92322 Châtillon, France*

³*Exail, F-25000, Besançon*

⁴*ICFO - Institut de Ciències Fotòniques, The Barcelona Institute of Science and Technology, Castelldefels (Barcelona) 08860, Spain*

⁵*Université Paris-Saclay, Institut d'Optique Graduate School, CNRS, Laboratoire Charles Fabry, 91127, Palaiseau, France*

(Dated: May 1, 2024)

Quantum Key Distribution (QKD) enables secret key exchange between two remote parties with information-theoretic security rooted in the laws of quantum physics. Encoding key information in continuous variables (CV), such as the values of quadrature components of coherent states of light, brings implementations much closer to standard optical communication systems, but this comes at the price of significant complexity in the digital signal processing techniques required for operation at low signal-to-noise ratios. In this work, we wish to lower the barriers to entry for CV-QKD experiments associated to this difficulty by providing a highly modular, open source software that is in principle hardware agnostic and can be used in multiple configurations. We benchmarked this software, called QOSST, using an experimental setup with a locally generated local oscillator, frequency multiplexed pilots and RF-heterodyne detection, and obtained state-of-the-art secret key rates of the order of Mbit/s over metropolitan distances at the asymptotic limit. We hope that QOSST can be used to stimulate further experimental advances in CV-QKD and be improved and extended by the community to achieve high performance in a wide variety of configurations.

I. INTRODUCTION

Quantum Key Distribution (QKD) aims at the exchange of a random string of bits – a cryptographic key – between two trusted users, commonly referred to as Alice and Bob, who are linked through an untrusted quantum channel and a public authenticated classical channel. This random bit string can then be used in symmetric encryption ciphers. Crucially, the security of the key exchange is not based on computational assumptions but on the laws of quantum physics leading to information-theoretic security [1].

Two major families of QKD protocols can be distinguished by the degrees of freedom in which the key information is encoded. In Discrete-Variable Quantum Key Distribution (DV-QKD), these are properties of single photons, such as polarisation or phase [2, 3], while in Continuous-Variable Quantum Key Distribution (CV-QKD), they are continuous degrees of freedom, such as the quadratures of coherent states [4]. In terms of practical implementation, an important difference between these two families lies in the detection technique. DV-QKD systems require single-photon detectors, which have limited detection rates due to their dead time and need to be operated in sub-kelvin temperatures to achieve a high efficiency [5, 6]. CV-QKD systems, on the other hand, use coherent optical detectors at room temperature. This reduces the size and cost of the detection apparatus and enables higher bandwidth detectors and hence higher measurement rates. Such detectors are also easier to integrate [7].

Several experimental demonstrations of CV-QKD

have been performed in the past few years, including over long distances [8], with high key rates [9, 10] and with integrated transmitters or receivers [11–13]. For such protocols that are inspired by classical communication systems, an important part of the implementation complexity comes down to the Digital Signal Processing (DSP) techniques required to recover the symbols at Bob's side and to correct the impairments coming from the transmission at a low Signal-to-Noise ratio (SNR). Indeed, to limit the amount of information leaked to the eavesdropper, Eve, CV-QKD systems operate in the regime of a few photons per symbol, which is the key difference with classical coherent systems that operate at high SNR. This is common to the several variations of CV-QKD protocols that exist, in terms of symbol shaping, generating a Local Oscillator (LO) for the coherent detection, or sending classical information for frequency and phase compensation.

In this work, our main contribution is to provide an open source, highly modular and highly customizable platform for CV-QKD, that we call QOSST: Quantum Open Software for Secure Transmissions. In practice, our platform is capable of performing CV-QKD exchanges at metropolitan distances, with a locally generated Local Oscillator, frequency multiplexed pilots, using the RF-heterodyne technique for the detection. We report here on asymptotic key rates up to 22 Mbit/s at zero distance and 1.2 Mbit/s at 25 km achieved using our open source platform. The extension of its use to other CV-QKD setups has either been investigated or should be possible without high complexity.

The rest of this paper is structured as follows: in sec-

tion II, we describe the protocol and the chosen setup, while in section III, we present our open source software for CV-QKD, QOSST, along with an in-depth explanation of the DSP algorithm. In section IV, an experimental benchmark of the software using our optical setup is presented, before discussing the results and drawing conclusions in sections V and VI respectively.

II. PROTOCOL AND SETUP

A. Description of the protocol

In general terms, CV-QKD protocols follow the steps shown in Fig. 1. Alice randomly chooses symbols drawn from a specific constellation, encodes them on quantum states and sends them to Bob through a quantum channel, which is characterised by a transmittance T and an excess noise ξ . Bob then measures the quantum states using coherent detection techniques. In some protocols, Bob needs to randomly choose which quadrature he measures at this moment. Then, Bob performs his DSP to recover the symbols and estimate the parameters of the channel (T and ξ) such that he can upper bound Eve's information, assuming that all the excess noise comes from eavesdropping. Finally, Alice and Bob perform error correction and privacy amplification. These last two steps are not implemented in the present work.

We now discuss in more detail the protocol steps. Alice's symbols are defined by complex numbers, whose real and imaginary parts are drawn from a particular probability distribution. CV-QKD systems up until recently have typically used Gaussian modulation, meaning that the real and imaginary parts of the symbols follow a Gaussian distribution. However, more recently discrete modulation was proposed as an attractive alternative to align with standard practices in coherent optical communications and optimise bandwidth use and classical post-processing. In this case, the set of possible symbols (the constellation) is finite. For instance, the M -Phase-Shift Keying (PSK) constellation has M possible symbols with the same amplitude (the symbols are spaced on a circle with an angle difference $2\pi/M$). The M -Quadrature Amplitude Modulation (QAM), with M being a perfect square, is also a discrete modulation where the symbols are placed on a $\sqrt{M} \times \sqrt{M}$ grid. It is also possible to approximate a Gaussian distribution with a QAM, using Probabilistic Constellation Shaping (PCS) or a Binomial Distribution. Examples of such modulations are shown in Fig. 3.

Security proofs for Gaussian modulation are well established, including also finite-size effects. Despite the complexity in proving security for discrete constellations, a security proof is now also available for any such constellation at the asymptotic limit [14], and was used in a recent implementation with PCS-QAM modulation [10].

In our work here, we use as baseline protocol the Gaussian Modulated Coherent States (GMCS) GG02

protocol [4]. After the generation of the symbols on Alice's side with using a random number generator (ideally a true random number generator such as a Quantum Random Number Generator (QRNG)), we prepare them for transmission applying techniques inspired by classical communications. In particular, the quantum symbols are filtered using a Root-Raised Cosine (RRC) filter to occupy a finite bandwidth $B = R_s \cdot (1 + \beta_{\text{RRC}})$ where R_s is the symbol rate and β_{RRC} is the so-called roll-off factor (taking values between 0 and 1). The RRC filter is also applied on Bob's side, giving, at the end, a Raised Cosine (RC) filter, which fulfills the Nyquist criteria to minimise inter-symbol interference [15].

In addition, the DSP algorithm involves the use of classical signals to perform frame synchronisation, clock recovery, carrier frequency estimation and phase compensation. For the synchronisation, we use a Constant Amplitude Zero AutoCorrelation (CAZAC) sequence, namely a Zadoff-Chu sequence [16–18]. This sequence has good correlation properties that allow for an easy synchronisation and is added before the quantum data. For clock recovery, carrier frequency estimation and phase compensation, we use two frequency multiplexed pilot tones, which are complex exponentials at defined frequencies. More details on the RC filter and Zadoff-Chu sequence can be found in appendix E.

The signal is modulated in one optical sideband (Optical Single SideBand (OSSB) modulation) and measured by Bob using the RF-heterodyne technique (see appendix D for more details). We refer to the whole protocol as OSSB-GMCS-GG02.

Once Bob's DSP is performed, Alice and Bob proceed to the parameter estimation step. Denoting as X and Y the random variables describing the symbols of Alice and Bob, respectively, we have:

$$Y = \sqrt{\eta T} X + N. \quad (1)$$

Here N is the white Gaussian noise with variance $1 + V_{el} + \eta T \xi$ in Shot Noise Units (SNU), where V_{el} and η are the electronic noise and quantum efficiency of the detector, respectively.

Then, if V_A is the modulation variance at Alice's side, the following formulas hold:

$$\begin{aligned} \langle X^2 \rangle &= V_A \\ \langle XY \rangle &= \sqrt{\frac{\eta T}{2}} V_A \\ \langle Y^2 \rangle &= 1 + V_{el} + \frac{\eta T}{2} V_A + \frac{\eta T}{2} \xi \end{aligned} \quad (2)$$

and hence, if η , V_{el} and V_A are known, we have

$$\begin{aligned} T &= 2 \cdot \frac{\langle XY \rangle^2}{V_A^2 \eta} \\ \xi &= 2 \cdot \frac{\langle Y^2 \rangle - 1 - V_{el} - \frac{\eta T}{2} V_A}{\eta T} \end{aligned} \quad (3)$$

We can also define the excess noise at Bob's side as $\xi_B = \eta T \xi / 2$ or $\xi_B = \langle Y^2 \rangle - 1 - V_{el} - \eta T V_A / 2$.

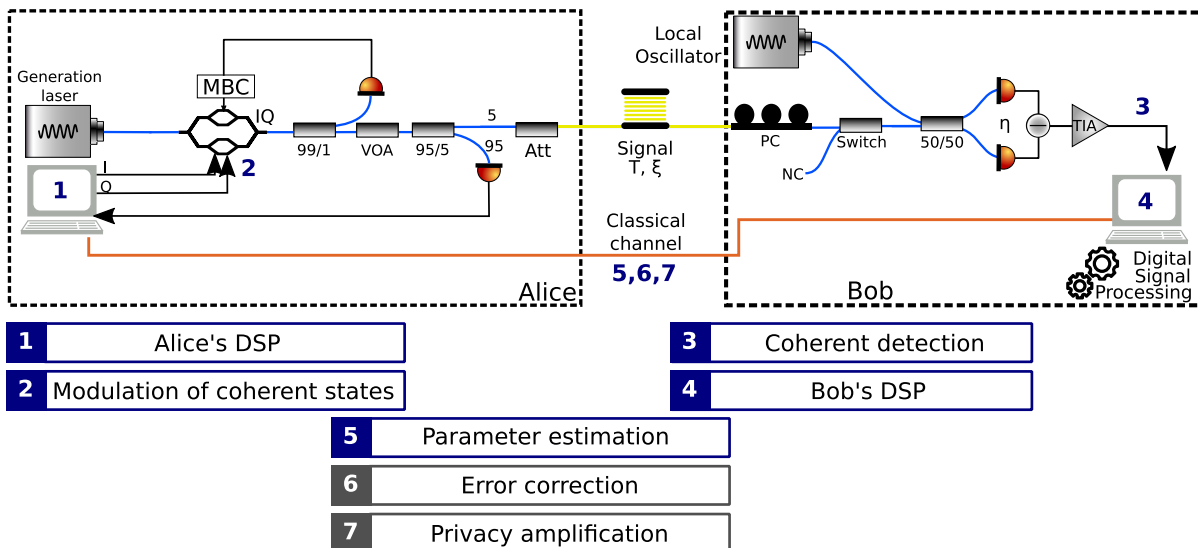


Figure 1. **Scheme of the CV-QKD optical setup and general steps of the protocol.** Here we show a fiber spool but a Variable Optical Attenuator can also be used as a channel. In this case, the polarisation controller at the input of Bob is not present. The attenuator at Alice's side introduces a fixed attenuation of 10 dB. Acronyms: IQ: IQ modulator, MBC: Modulator Bias Controller, VOA: Variable Optical Attenuator, Att: Attenuator (fixed), PC: Polarisation controller, NC: Not Connected, TIA: Trans-Impedance Amplifier, DSP: Digital Signal Processing. The details of the setup are provided in the main text.

After parameter estimation, according to the asymptotic security proof for Gaussian modulation and for reverse reconciliation where Bob's data is used to form the key, it is possible to bound the information that Eve has acquired using the Holevo bound χ_{BE} and compute the Secret Key Rate (SKR) using the Devetak-Winter formula [19]:

$$K_f = \beta I_{AB}(V_A, T, \xi, \eta, V_{el}) - \chi_{BE}(V_A, T, \xi, \eta, V_{el}), \quad (4)$$

where β is the reconciliation efficiency and I_{AB} is the mutual information shared by Alice and Bob. In the security analysis, we make the standard assumption in CV-QKD implementations of a trusted detector, which means that Eve cannot interfere with Bob's receiver. This allows us to separate the losses (η) and the electronic noise (V_{el}) of the receiver and to consider only the remaining losses (T) and noise (ξ) as a signature of the presence of Eve, giving a more favourable bound on the leaked information.

To take into account the finite number of samples used to estimate the values of T and ξ in practice, we use worst-case estimators as follows: assuming that ξ_B and T are sampled from a Gaussian probability distribution, we compute the values x_B^w and T^w such that the probability that the actual value of ξ_B (resp. T) is above ξ_B^w (resp. below T^w) is ε , which in our case is chosen to be $\varepsilon = 10^{-10}$. We also perform finite-size analysis following the analysis of [20] (see section V for more details).

The key rate that we compute in this way gives us the number of secret bits per symbol. To get the secret key rate in bits/s, one can apply the formula $K_s = R_s \cdot K_f$, where R_s is the symbol rate.

Finally, in the complete protocol Alice and Bob perform error correction and privacy amplification. Error correction can be done for instance using techniques described in [21, 22]. Privacy amplification compresses the key so that Eve does not hold any meaningful information on it, and is typically done using Toeplitz matrices [23]. These steps are not implemented in QOSST yet and are beyond the scope of this paper.

B. Experimental setup

The experimental setup is shown in Fig. 1 together with the protocol steps. The setup for generating the quantum states at Alice's side is composed of a continuous wave laser (NKT Koheras Basic), an IQ modulator (Exail MXIQER-LN-30), a Modulator Bias Controller (MBC) (Exail MBC-IQ-LAB), an electronic Variable Optical Attenuator (VOA) (Thorlabs V1550PA), followed by a 95:5 beam splitter, with the 95% output going into the monitoring photodiode (Thorlabs PM101A) and the 5% output going into a 90:10 beam splitter, whose 10% output is sent to the output of Alice. Alice is also composed of a control PC and a Digital-to-Analog Converter (DAC) (Teledyne SDR14Tx). The rate of the DAC is 2 GSa/s with a bandwidth of 1 GHz, and it has a 14-bit resolution. The DAC is connected to the control PC using a PCIe connector.

The role of the MBC is to apply the correct bias DC voltages to the modulator using a feedback loop. Indeed, a single polarization IQ modulator requires 3 bias voltages that can be found by adding low frequency dithers into the DC input and applying an optimiza-

tion algorithm. In this way, it is possible to maximise the extinction ratio and to adjust the relative phase between the two quadratures to 90° . The role of the monitoring photodiode is to measure the number of photons per symbol in the quantum data: if the ratio r_{conv} between the output power of Alice and the power at the monitoring photodiode is known (see appendix A for more details on the calibration of Alice), then the average number of photons per symbol at Alice’s output is given by

$$\langle n \rangle = \frac{r_{\text{conv}} \cdot P_{\text{monitoring}}}{E_{\text{ph}} \cdot R_s}, \quad (5)$$

where R_s is the symbol rate, $P_{\text{monitoring}}$ is the optical power on the monitoring photodiode for the quantum data, and $E_{\text{ph}} = hc/\lambda$ is the energy of the photon. The average number of photons per symbol allows us to compute Alice’s variance V_A using the following formula:

$$V_A = 2 \cdot \langle n \rangle. \quad (6)$$

The final 90:10 beam splitter is used to provide a further 10 dB attenuation. The electronic VOA is used, together with the power of the laser and the state generation parameters of Alice’s DSP, to tune both Alice’s variance and the power of the classical signals (pilot tones). It is controlled using a National Instruments card (USB-6363).

The entrance of the setup at Bob’s side is controlled by an optical switch (Thorlabs OSW12-1310E), which can be used to disconnect the quantum channel from the detection system during the calibration, in order to prevent an eavesdropper from tampering with this procedure. This also ensures that noise coming from Alice’s setup is not inadvertently included in the calibration. The output of the switch goes into a 50:50 beam splitter, where the signal is mixed with a continuous wave laser (NKT Koheras Basik) acting as a Local Local Oscillator (LLO). The two outputs of the 50:50 beam splitter go into a Balanced Homodyne Detector (BHD) (Thorlabs PDB480C-AC), whose output is filtered using an analog low-pass filter with a cut-off frequency of 700 MHz before being sent to Bob’s control PC through an Analog-to-Digital Converter (ADC) (Teledyne ADQ32) working at 2.5 GSa/s and with a bandwidth of 1.5 GHz. The ADC is connected to the PC using a PCIe connector. Bob’s setup also requires calibration, in particular for the efficiency and the electronic noise (see appendix B for details).

Both the transmitter setup of Alice and the receiver setup of Bob use Polarisation Maintaining (PM) components. The quantum channel between Alice and Bob in our experiments is implemented either with a PM electronic VOA (Thorlabs V1550PA) or with a single mode fiber spool with a length of 25.2 km (linear losses: 4.75 dB, attenuation coefficient: 0.188 dB/km, total losses with connector: 5.22 dB), whose characteristics are similar to the fibers deployed in standard optical fiber networks. When using a single mode fiber spool,

a manual polarisation controller placed at Bob’s input, before the optical switch, is used to recover the polarisation after the transmission.

Both control PCs are connected to the same local network to provide the classical channel. Authentication on the classical channel is done using the Post Quantum Algorithm for Digital Signatures Falcon [24–26], which is expected to be standardised in the NIST Post Quantum Cryptography standardization process [27]. There is no other link between Alice and Bob.

III. DESCRIPTION OF THE OPEN SOURCE PLATFORM

We now present our software, called QOSST (*Quantum Open Software for Secure Transmissions*), that was developed in our team and aims at performing all basic operations that are necessary for CV-QKD implementations.

A. Design of the software

The software is written using the Python language [28], and was conceived to be modular. Importantly, it is in principle not limited to a specific hardware and transmission or detection schemes. It might however require some work to adapt the DSP for operations with time-multiplexed pilots or in the pulsed regime.

QOSST is released as an open source software: the different repositories can be found at <https://github.com/qosst> and a good starting point to understand QOSST would be <https://github.com/qosst/qosst>.

The software is separated in 7 packages, out of which 6 are released and 1 is still under development and has not been released yet. The structure is shown in Fig. 2 along with the interconnections between the packages and the main functionalities. In the following, we provide details on these packages.

B. Hardware Abstraction Layer

The Hardware Abstraction Layer (HAL) package provides abstract interface classes for the hardware. In this sense, it is easy to write a class to control a new hardware and then replace the class in the configuration file. This operation should be totally transparent for the rest of QOSST. This package also provides dummy classes that can be used as default, and special values when the hardware is not used. The classes for the actual hardware that we used are not released with QOSST. A tutorial on how to write new classes is available in the documentation.

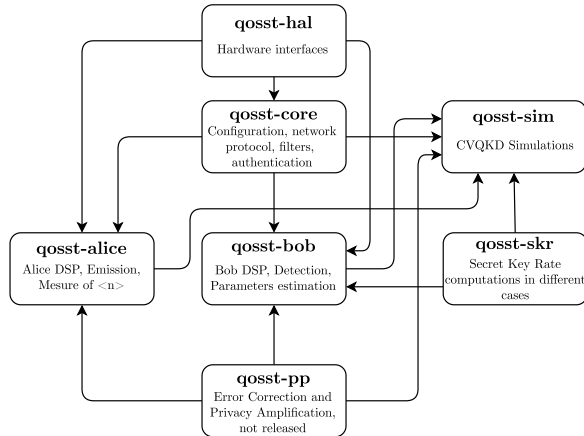


Figure 2. **Interconnections of the QOSST packages along with the main functionalities.** An arrow from block A to block B means that the block A is a dependence of block B. For instance, the arrow from qosst-core to qosst-alice means that qosst-alice uses qosst-core. The details of the packages are provided in the main text.

C. Core

The core package of QOSST is the biggest of the seven and implements the functions that are common to Alice and Bob. In particular, it implements the control protocol, the configuration, the filters, the synchronisation sequence, the modulations, the authentication, the data containers and some common functions.

A specific network protocol for the classical channel has been designed for QOSST and its specifications are described in the documentation. It has a built-in authentication system that can be used to authenticate the classical channel. The package also provides a socket class to easily interact between a server and a client.

The configuration file, which controls the whole protocol, is written in TOML [29] since the Python ecosystem appears to adopt TOML as its default configuration language [30, 31] and this language is also easily readable and modifiable by both humans and machines [30, 32]. The configuration holds more than 100 parameters and part of them needs to be optimized to reach the best values of excess noise and secret key rate. An example configuration file is provided with the software and the explanation of all the parameters can be found in the documentation.

The package also implements the Zadoff-Chu sequence for synchronisation and the following filters: RC, RRC and pulsed (filter with a rectangular temporal response).

The package provides the codes for different modulation schemes, which can be used instead of the default Gaussian modulation: M -PSK, M -QAM, PCS-QAM, Binomial QAM. The modulations are shown in Fig. 3.

The authentication is handled by adding a signed digest in the control protocol frame. In the QOSST implementation, the digest can either be not signed or signed

with the Falcon algorithm.

We also note that the documentation includes the description of some other useful functions present in QOSST but that are not relevant for this paper.

D. Alice

The Alice package implements mainly two functionalities: the first one is the server, with which Bob's client interacts, and the other one is the DSP for signal generation. The server answers to valid Bob's requests. In particular, it generates the data and applies it to the IQ modulator when asked to do so.

In the following, we describe the DSP algorithm for signal generation using Nyquist filters and frequency multiplexed pilots. The first step is the generation of the random symbols from the chosen distribution [33]. In QOSST, at the time of writing of this paper, the entropy source is a pseudo random number generator from the `numpy` library [34]; in practice, entropy should be provided by a QRNG and this can be easily integrated in QOSST. The symbols are then upsampled to match the symbol rate R_s and filtered using a RRC cosine filter with roll-off factor β_{RRC} . The quantum data is then shifted in frequency to the center frequency f_{shift} . Two pilot tones, of frequencies $f_{\text{pilot},1}$ and $f_{\text{pilot},2}$, are frequency multiplexed with the signal. Their frequencies are chosen so that

$$f_{\text{pilot},1}, f_{\text{pilot},2} \notin \left[f_{\text{shift}} - \frac{B_s}{2}, f_{\text{shift}} + \frac{B_s}{2} \right], \quad (7)$$

with

$$B_s = R_s \cdot (1 + \beta_{\text{RRC}}) \quad (8)$$

the bandwidth of the quantum data. Then a Zadoff-Chu sequence with length L_{ZC} and root R_{ZC} is generated and prepended to the signal. The DSP can also prepend and append a sequence of zeros if needed. The frequency form of the signal is shown in Fig. 4.

Alice's server applies this sequence to the IQ modulator when Bob sends the network request indicating he is ready. Alice's server also estimates the average number of photons per symbol $\langle n \rangle$ and sends the data for the parameter estimation to Bob. Finally, this package provides a script to calibrate the conversion factor r_{conv} .

E. Bob

This is the second biggest package of QOSST and provides the DSP and scripts for Bob. First, the package provides ways to characterize the receiver, in particular to measure the detector efficiency η and the electronic noise V_{el} (more precisely, the sequence of electronic noise samples that can be used to compute V_{el} afterwards). Then, Bob provides a client that can either be used with command line scripts or as a Graphical User Interface

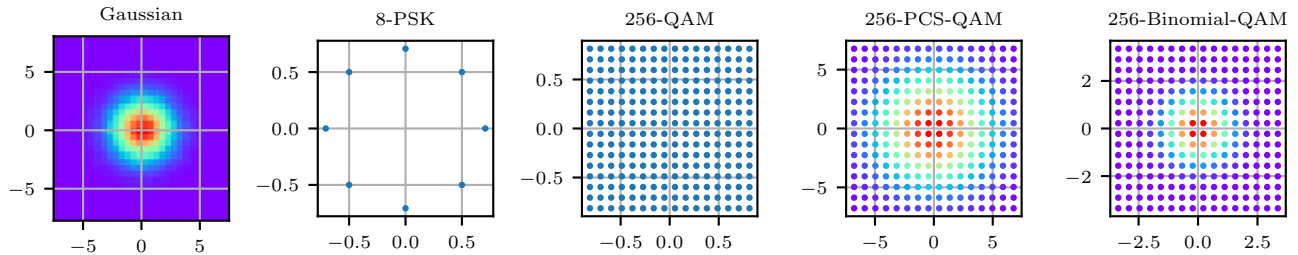


Figure 3. **Possible modulation schemes in the QOSST software.** From left to right: Gaussian (variance: 3, number of points: 100000), 8-PSK (variance: 1), 256-QAM (variance: 1), 256-PCS-QAM (variance: 35, $\nu = 0.01$ [10]), 256-Binomial-QAM (variance: 3). The variance parameter does not always represent the actual variance of the generated sequence, but is a parameter that is proportional to the actual variance.

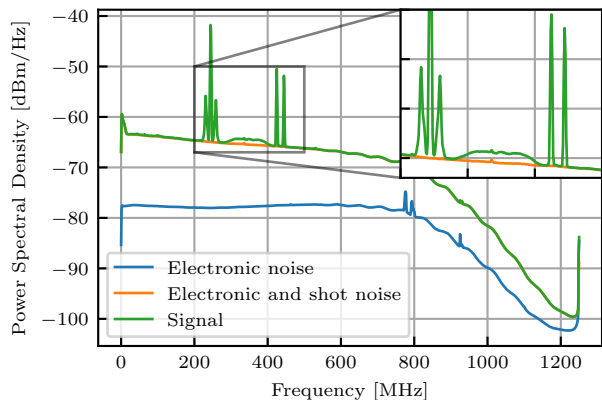


Figure 4. **Power Spectral Densities of the electronic noise, electronic and shot noise and signal data.** In this example, we can see that f_{beat} is around 240 MHz, and that at this frequency, there is some “low-frequency” noise coming from the setup of Alice. This does not impact the performance of the CV-QKD protocol since it is filtered out. The quantum signal has a bandwidth of 140 MHz and is centered at $f_{\text{beat}} + f_{\text{shift}}$ with $f_{\text{shift}} = 100$ MHz. The two pilots are placed at the frequencies $f_{\text{beat}} + f_{\text{pilot},1}$ and $f_{\text{beat}} + f_{\text{pilot},2}$ with $f_{\text{pilot},1} = 180$ MHz and $f_{\text{pilot},2} = 200$ MHz. The first pilot is slightly more powerful than the second one to be consistent when recovering the frequencies of the pilots. The slight signal at $f_{\text{beat}} - f_{\text{shift}}$, $f_{\text{beat}} - f_{\text{pilot},1}$ and $f_{\text{beat}} - f_{\text{pilot},2}$ is due to imperfections on the sideband suppression.

(GUI), which interacts with the server of Alice. In particular, the client starts the acquisition, triggers Alice, and then applies the DSP to recover the symbols. The client also requests Alice’s symbols and performs the parameter estimation.

Again, we give an overview of the DSP algorithm for our protocol. First, if the automatic shot noise calibration is enabled, Bob uses the beginning of the data (before the switch connects his setup with the quantum channel) to estimate the shot noise value, denoted as σ_0^2 , that will be used for normalisation of the symbols in SNU at Bob’s side (more precisely, the sequence of electronic and shot noise samples that can be used to es-

timate σ_0^2 afterwards, as will be explained later). Then, the first step of the actual DSP is to find an estimate of the position of the Zadoff-Chu sequence. Knowing this information, it is possible to isolate the part of the data with the pilots and to search in the frequency domain the two frequencies of the pilots: $\tilde{f}_{\text{pilot},1}^B$ and $\tilde{f}_{\text{pilot},2}^B$. The clock difference is estimated using the formula

$$\Delta f = \frac{\tilde{f}_{\text{pilot},2}^B - \tilde{f}_{\text{pilot},1}^B}{f_{\text{pilot},2} - f_{\text{pilot},1}}, \quad (9)$$

which is then used to correct the clock mismatch. This step is known as clock recovery. Once the clock is recovered, the frequencies of the pilots are estimated again to find $f_{\text{pilot},1}^B$ and $f_{\text{pilot},2}^B$ and the beat frequency is computed as

$$f_{\text{beat}} = f_{\text{pilot},1}^B - f_{\text{pilot},1}. \quad (10)$$

The beat frequency is the frequency difference between the two lasers, and can typically move by a few MHz if the lasers are not very stable. This was not the case in our implementation thanks to the stability of our lasers. The sequence is unshifted by f_{beat} . This step is known as carrier frequency estimation. Then, the Zadoff-Chu sequence is found by cross-correlations, which gives a precise value for the beginning of the sequence. Knowing the number of symbols and the symbol rate, the sequence is reduced to the useful sequence containing only the symbols. The rest of the DSP is performed by subframes because the change in f_{beat} can be relatively quick and we want to do the analysis on a time scale where f_{beat} can be considered constant. In particular, this subframe analysis is critical when the lasers used are not very stable. For each subframe, the frequencies of the two pilots are estimated again, and f_{beat} is also estimated again using Eq. (10). The first pilot is filtered and used as a phase reference, and a matching RRC filter is applied to the quantum data, after it has been shifted to baseband by $f_{\text{beat}} + f_{\text{shift}}$. Following this, the best sampling point is found by the maximal variance (eye diagram) method [35]. The phase is then corrected using the phase reference from the pilot (a uniform filter is applied to the phase reference for

better performance). At this point, Alice's and Bob's symbols only differ by a global phase (and some noise) and the global phase correction is performed by testing all global rotations and keeping the one with the best covariance term $\langle XY \rangle$. More details on the DSP of both Alice and Bob can be found in appendix E.

After these steps, Bob applies the same DSP to the electronic noise and electronic and shot noise samples, whose outputs give the electronic noise and electronic and shot noise symbols. The program also requests $\langle n \rangle$ from Alice. Bob then has the electronic noise symbols, the electronic and shot noise symbols, the average number of photons per symbol $\langle n \rangle$, Alice's symbols (for estimation) and Bob's symbols. The variance of the electronic noise symbols is σ_{el}^2 and the variance of the electronic and shot noise symbols is $\sigma_{el}^2 + \sigma_0^2$, which allows to calculate σ_0^2 . V_{el} is then calculated as $V_{el} = \sigma_{el}^2 / \sigma_0^2$ and T and ξ can be calculated with the formulas in Eq. (3) using the appropriate normalisation (see appendix C for more details).

In this way, Bob ends up with T , ξ , $\langle n \rangle$, η and V_{el} and can proceed to the estimation of the secret key rate using Eq. (4) (using code from the skr package). We typically use the value $\beta = 0.95$ for the reconciliation efficiency [36].

The Bob package also provides automated scripts to allow for performing the experiment over long periods of time.

F. Secret Key Rate

The skr package provides classes with static methods to compute the secret key rate depending on the setup and the security assumptions. Three methods are currently available: untrusted homodyne detector [37], trusted homodyne detector [38] and trusted heterodyne detector [39], all in the asymptotic case. In particular, the code that is released in this package does not take into account discrete modulations and finite-size effects.

G. Simulations and Post-Processing

The last two packages of QOSST are sim (simulations) and pp (Post Processing). The first one, qosst-sim, is also released in the software suite and allows for the simulation of CV-QKD exchanges.

While a partial codebase for the post processing package exists, including error correction with LDPC codes, and privacy amplification with Toeplitz matrices, its content was judged not mature enough for publication.

IV. RESULTS

To assess the performance of CV-QKD driven by QOSST, we tested and benchmarked it using the experimental setup presented in section II. For this, we used

five channel configurations. In four of them, the PM VOA emulates an optical fiber of 0, 5, 10 and 25 km at 0.2 dB/km (the actual attenuation values were in practice slightly higher). The last configuration uses a fiber spool of 25 km (total attenuation: 5.22 dB) and a manual polarisation controller to compensate for the polarisation transformation of the fiber.

For each configuration, we optimized around 10 DSP and physical parameters using automated scripts, including the RRC filter's roll-off factor, the pilot amplitudes, the average number of photons $\langle n \rangle$, the power of Alice's laser, the power of Bob's laser, the size of the subframes, the width of the filters for the tones, the length of the uniform filter of the phase, the baud rate, the frequency shift and the frequency difference between the pilots. The optimization is done automatically using the qosst-bob-optimize script, which exchanges frames and dynamically changes the parameters of Alice and Bob.

Here we give examples of optimization of three parameters in terms of the excess noise at Bob's side, even though it is required to perform the optimization on all the parameters to reach the best performance. For each point on each experiment, five frames were exchanged and the results are given in Fig. 5.

For the roll-off factor, we see that a value below 0.2 induces a slightly higher excess noise, and for high values a steep increase of the excess noise is observed. We can explain this effect by recalling that the bandwidth of the signal after the RRC filter is given by Eq. (8). Since in this experiment the frequencies of the pilots are fixed, Eq. (7) is violated when β_{RRC} becomes too big. In practice, the quantum data and the pilots start to be superposed in frequency, increasing the noise on the quantum data and reducing the precision of the phase compensation algorithm. This effect starts when $\beta_{\text{RRC}} = 0.65$, which corresponds to having $f_{\text{shift}} + B_s/2 = 182.5$ MHz and is the first roll-off value where $f_{\text{shift}} + B_s/2 > f_{\text{pilot},1} = 180$ MHz.

For the subframe size, we see that the excess noise tends to increase when the size increases. This is reasonable as the bigger the subframe, the more the f_{beat} frequency is moving and the less precise the DSP algorithm is. However, for very small values of subframe size, the excess noise also seems to be slightly higher. This was attributed to the fact that, in this regime, there are not enough samples to properly estimate the frequencies of the pilots.

Finally for the pilot amplitudes, a too low value results in a very high excess noise. This is due to the fact that either the frequency of the pilots cannot be properly recovered, or the SNR of the pilots is too low to get a good phase compensation. Also, the excess noise increases with the pilot amplitudes, which can be explained by frequency cross-talk between the quantum data and the pilots. The optimal value depends mainly on the SNR at the receiver, which means that this value should be optimized for every distance. This optimization was done for each of the 5 configurations. This first

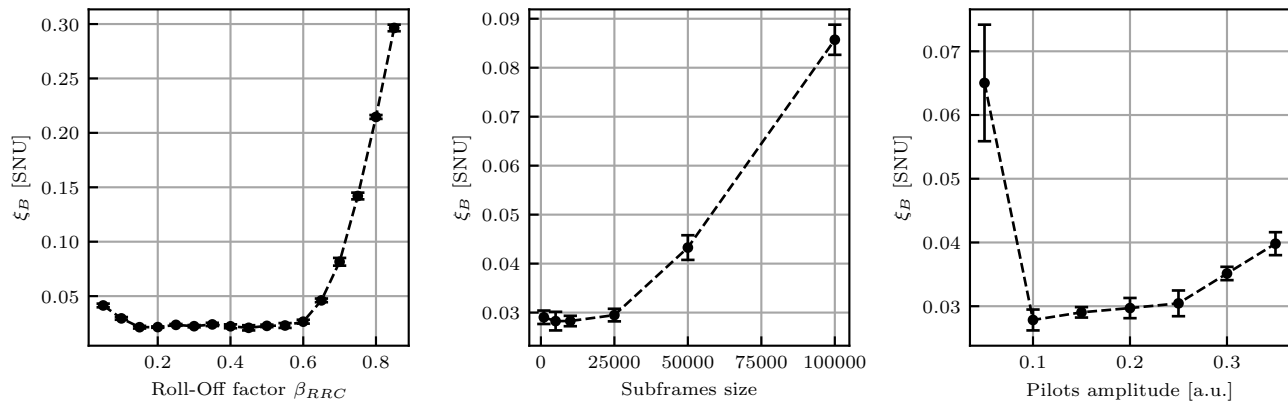


Figure 5. **Results of three examples of optimisation.** For the three experiments, the number of frames per point is 5. For the pilot amplitude a fine search was conducted after a first coarse one to find the optimal value in the 0.05-0.15 region.

Parameter	Value
β_{RRC}	0.5
R_s	100 MBaud
f_{shift}	100 MHz
$f_{pilot,1}$	180 MHz
$f_{pilot,2}$	200 MHz
L_{ZC}	3989
R_{ZC}	5
Acquisition time	50 ms
Shot noise time	10 ms
DAC rate	2 GSa/s
ADC rate	2.5 GSa/s
Modulation	Gaussian

Table I. Values of the main parameters that are independent of the distance and other fixed parameters in our setup.

coarse optimisation was also followed by a fine optimisation (in this example in the region 0.05-0.15). The amplitude of the pilots is given in arbitrary units, and the actual amplitude of the pilot is proportional to this value. The optimal value was found to correspond to ~ 12 dB.

Our tests showed that the optimal value of some parameters is independent of the attenuation. The values of these parameters are given in table I, along with some other important fixed parameters in our experiments. Once all the parameters were optimized, an experiment of 200 CV-QKD frames was launched for each configuration, which corresponds to around 10 to 12 hours. In our setup, the instability, due for instance to mechanical vibrations, was not compensated. This does not necessarily lead to negative key rates for some frames but it may affect the worst-case estimators in the finite-size analysis because of an increased variance, leading to pessimistic values. Especially for the fiber experiment, the effect of instability is important as, after some time, the polarisation mismatch becomes too big and the efficiency of the detection drops, giving a null key rate.

In Fig. 6, we plot the excess noise for the experiments using the VOA with attenuation corresponding to 25 km

Experiment	ξ_B (SNU)	K_∞ (MBit/s)	K_{FSE} (MBit/s)
VOA 0 km	0.0095	22.4	17.7
VOA 5 km	0.0091	11.9	5.82
VOA 10 km	0.0076	6.35	2.55
VOA 25 km	0.0062	1.43	0
Fiber 25 km	0.0072	1.17	0

Table II. **Average results of the experiments.** K_∞ is the asymptotic secret key rate and K_{FSE} is the secret key rate considering finite-size effects.

and the fiber spool. The estimation of the excess noise is rather stable. The stability of the estimation of $\langle n \rangle$, T , shot noise and V_{el} were also checked. The results in terms of the secret key rate are given in Fig. 7.

The average results are summarized in table II. The secret key rates are given for both the asymptotic and finite-size case. For the experiments with the fiber spool and the VOA at 25 km, the finite-size key rate is zero using the worst-case estimators. However, it is also possible to measure the finite-size key rate for each frame using the framework from [20], similarly to how the system would operate in an operational setting, but once again it is not possible to extract a secret key rate for the number of symbols used here. We obtain a positive average when considering at least 10^7 symbols (average finite-size key rate in order of hundreds of kbit/s), but it would be challenging to reach this block size over the same number of frames, since it would roughly multiply the DSP, and hence the experiment time, by 10. Increasing the block size will require to perform optimisation on the overall software in order to reduce the computation time (and memory usage).

V. DISCUSSION

Using as a starting point our obtained results, there are several ways these can be extended and improved. We discuss some directions below.

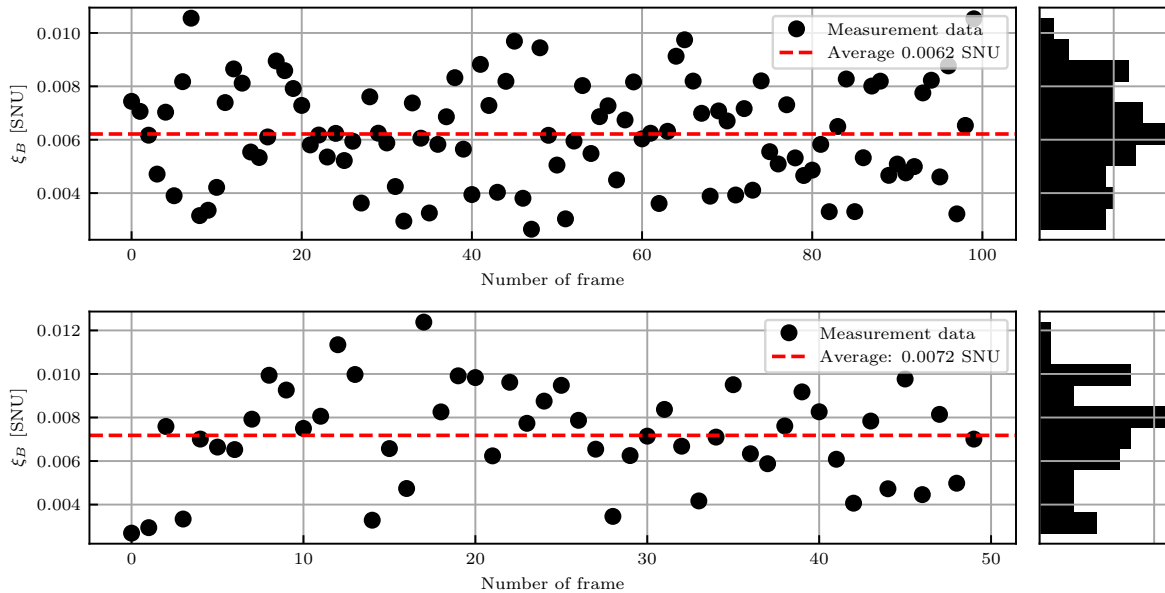


Figure 6. **Excess noise at Bob's side for the experiments with the VOA (equivalent to 25 km) and the 25 km fiber spool.** Top: experiment with the VOA. One frame where the DSP failed was removed from the 100 frames. The average excess noise at Bob's side is 0.0062 SNU. Bottom: experiment with the fiber. Only the first 50 frames where the experiment was stable are shown. During this time, the average excess noise at Bob's side is 0.0072 SNU.

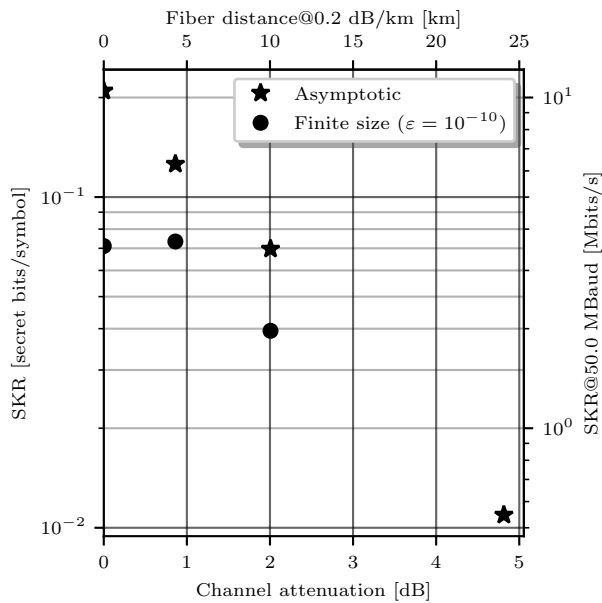


Figure 7. **Achievable secret key rate in the asymptotic and finite-size regime.** The security parameter is $\varepsilon = 10^{-10}$. For each experiment, the actual attenuation performed by the VOA was always slightly above the targeted one. For the experiments with the VOA, the extra attenuation corresponds to a slightly higher attenuation coefficient on the actual fiber, and a small mismatch in polarisation that is not included in the efficiency. For the experiment with the fiber spool and the VOA at 25 km the finite-size secret key rate was zero.

a. Finite-size effects. Unfortunately the experiments at 25 km do not give any key rate in the finite-size case with the worst-case estimator method. As was briefly mentioned above, another method is to follow the analysis in [20] for each frame. For $N = 10^6$, which was the number of symbols in our frame, we unfortunately also got a key rate of zero for each frame. However, increasing N for the analysis, we start to get some frames with a positive finite-size key rate. For the experiment with 25 km, 40 frames out of 100 at $N = 10^7$ and then all the frames at $N = 10^8, 10^9, 10^{10}$. For the experiment with the fiber, there were respectively 14, 77, 93 and 94 frames with a positive finite-size key rate at $N = 10^7, 10^8, 10^9, 10^{10}$. We now give the average key rate over all the frames (and in parentheses the average over the positive frames) for the VOA experiment at 25 km: 96 kbit/s (240 kbit/s), 817 kbit/s, 1.1 Mbit/s, 1.2 Mbit/s for $N = 10^7, 10^8, 10^9, 10^{10}$ and for the experiment with the fiber: 39 kbit/s (280 kbit/s), 475 kbit/s (617 kbit/s), 718 kbit/s (772 kbit/s), 803 kbit/s (854 kbit/s) with the same N . This makes us confident that one way to get better results would be to increase the number of points that are used for the estimation of T and ξ . We indeed checked that the variance of those two variables has a dependence over the number of samples used to estimate them. This would reduce the confidence interval and give better worst-case estimators and subsequently better key rates.

b. Increasing the distance. One of the current limitations of the QOSST software is the achievable distance. In practice the DSP can work at least until 10 dB but at that value the estimation of the distance is im-

precise and the excess noise too high to get a secret key. However, the distance could be improved either by improving the DSP or by changing the generation of the pilots. Indeed, we believe that one of the limitations comes from the finite amplitude difference that can be applied between the quantum symbols and the pilots. A solution could be to generate the pilots and the quantum symbols on different paths, or to send the pilots on the other polarisation, which would allow for more powerful pilots and less crosstalk, as suggested in [40].

c. Stability for polarization. Another important issue is the manual polarisation controller, since it allows only for a few hours of stability of the setup before another calibration becomes necessary. Two solutions may be used to handle this issue: the first is to use an automatic polarisation controller with an error signal coming from the detection to actively correct for polarisation drifts in the fiber, and the second one is to perform a polarisation-diverse detection by adding a Polarisation Beam Splitter (PBS) on the signal path and detecting both polarisations by using two balanced detectors. The first one has the advantage of requiring an optical scheme very similar to the one already implemented, and of using the same DSP. However, the algorithm to adjust the polarisation controller should be added. The second solution requires more hardware but can be easily adapted for using a double polarisation in the transmitter setup.

Since the experiments presented in this paper were done, automatic polarisation compensation with a motorised polarisation controller was implemented and tested, and is released with QOSST.

d. Increasing the symbol rate. The choice of the optimised parameters reported in this work was performed using a slightly older system with a DAC that was limited to 500 MSa/s (Keysight M330A) and with a bandwidth of 200 MHz. This justifies the relatively low symbol rate used in the present work. The increase of the symbol rate (and hence the occupation bandwidth), allowed by the new DAC will be the subject of future investigation. We believe that there are no particular limitations coming from our software other than longer time and larger memory occupation for the DSP.

e. Duration of the Digital Signal Processing. Currently, the DSP (and the parameter estimation) is run for each frame before requesting the next frame, which means that two frames are temporally separated by the DSP time. We conducted an experiment on 20 frames to evaluate the DSP time and we measured two durations: the first is the DSP itself, which outputs the symbols for each subframe with a global phase difference, and the second is recovery of the global phase, which also requires Alice sending her data to Bob. The experiment is conducted with the same parameters as the experiment for 0 km and requests 50% of the symbols to Alice (500000 symbols per frame). The average time for the DSP is 70 s/frame and the average time for the symbol exchange and global phase correction is 164 s/frame for a total time between frames of 234 s/frame. The

second time can be reduced by exchanging less data for the parameter estimation. For example, taking 10% of the data (*i.e.* 100000 symbols for the parameter estimation), the DSP time is similar (72 s/frame on average) and the data exchange and global phase recovery has an average time of 101 s/frame, yielding a total of 173 s between the frames. This has the advantage of reducing the time and increasing the key rate, but the estimators for the finite-size analysis are worse. Both times could be reduced by optimizing the code, or by using a different programming language.

VI. CONCLUSION

We have presented a highly modular open source platform for experimental continuous-variable quantum key distribution that is able to exchange a secret key rate at metropolitan distances using the OSSB-GCMS-GG02 protocol with frequency multiplexed pilots and RF-heterodyne coherent detection. We obtain key rates of the order of the Mbit/s for distances up to 25 km at the asymptotic limit. Our platform can be easily extended to other CV-QKD protocols and hardware configurations, including with double polarisation and phase-diverse heterodyne detection. Our platform can be improved and extended by integrating techniques accessible in the near term.

This software is released to the community in the hope that it will help to stimulate further experimental research in CV-QKD by providing the initial means to set up a working CV-QKD setup. We also hope that the community will help this software suite grow, for instance by adding other CV-QKD configurations, increasing the execution speed, complementing the package for computing secret key rates under different assumptions and implementing the post processing operations.

AUTHOR CONTRIBUTIONS

VMA implemented the global phase recovery algorithm. VMA and MS contributed to increasing the achievable distance and to tests of the software. YP implemented the rest of the software. MS, BG, LTV, PG, AR and ED participated in crucial discussions that directly contributed to the platform design and its improvements.

ACKNOWLEDGMENTS

Figure 1 was created by partly using the gwopics component library distributed under CC-BY-NC license, attribution to Alexander Franzen [41]. The authors acknowledge financial support from the European Union's Horizon Europe research and innovation programme under the project QSNP (Grant

No. 101114043) and the PEPR integrated project

QCommTestbed (ANR-22-PETQ-0011), which is part of Plan France 2030.

-
- [1] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. L. Pereira, M. Razavi, J. Shamsul Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, *Advances in Optics and Photonics* **12**, 1012 (2020).
- [2] C. H. Bennett and G. Brassard, in *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, edited by I. Press (1984).
- [3] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, *Nature* **557**, 400 (2018).
- [4] F. Grosshans and P. Grangier, *Physical Review Letters* **88** (2002), 10.1103/physrevlett.88.057902.
- [5] E. Ip, A. P. T. Lau, D. J. F. Barros, and J. M. Kahn, *Opt. Express* **16**, 753 (2008).
- [6] M. D. Eisaman, J. Fan, A. Migdall, and S. V. Polyakov, *Review of Scientific Instruments* **82** (2011), 10.1063/1.3610677.
- [7] Q. Liu, Y. Huang, Y. Du, Z. Zhao, M. Geng, Z. Zhang, and K. Wei, *Entropy* **24**, 1334 (2022).
- [8] Y. Zhang, Z. Chen, S. Pirandola, X. Wang, C. Zhou, B. Chu, Y. Zhao, B. Xu, S. Yu, and H. Guo, *Physical Review Letters* **125** (2020), 10.1103/physrevlett.125.010502.
- [9] Y. Pan, H. Wang, Y. Shao, Y. Pi, Y. Li, B. Liu, W. Huang, and B. Xu, *Optics Letters* **47**, 3307 (2022).
- [10] F. Roumestan, A. Ghazisaeidi, J. Renaudier, L. T. Vidarte, A. Leverrier, E. Diamanti, and P. Grangier, *Journal of Lightwave Technology*, 1 (2024).
- [11] J. Aldama, S. Sarmiento, S. Etcheverry, I. L. Grande, L. T. Vidarte, L. Castolvero, A. Hinojosa, T. Beckerwerth, Y. Piétri, A. Rhouni, E. Diamanti, and V. Pruneri, in *Optical Fiber Communication Conference (OFC) 2023* (Optica Publishing Group, 2023) p. M11.3.
- [12] Y. Piétri, L. T. Vidarte, M. Schiavon, P. Grangier, A. Rhouni, and E. Diamanti, in *Optical Fiber Communication Conference (OFC) 2023* (Optica Publishing Group, 2023) p. M11.2.
- [13] A. A. E. Hajomer, C. Bruynsteen, I. Derkach, N. Jain, A. Bomhals, S. Bastiaens, U. L. Andersen, X. Yin, and T. Gehring, (2023), arXiv:2305.19642v1 [quant-ph].
- [14] A. Denys, P. Brown, and A. Leverrier, *Quantum* **5**, 540 (2021).
- [15] J. G. Proakis, *Digital Communications 5th Edition* (McGraw Hill, 2007).
- [16] R. Heimiller, *IEEE Transactions on Information Theory* **7**, 254 (1961).
- [17] D. Chu, *IEEE Transactions on Information Theory* **18**, 531 (1972).
- [18] H.-J. Zepernick and A. Finger, *Pseudo random signal processing* (Wiley-Blackwell, Hoboken, NJ, 2005).
- [19] I. Devetak and A. Winter, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **461**, 207 (2005).
- [20] A. Leverrier, F. Grosshans, and P. Grangier, *Physical Review A* **81** (2010), 10.1103/physreva.81.062343.
- [21] X. Wang, Y. Zhang, S. Yu, and H. Guo, *Scientific Reports* **8** (2018), 10.1038/s41598-018-28703-4.
- [22] K. Gümüş, T. A. Eriksson, M. Takeoka, M. Fujiwara, M. Sasaki, L. Schmalen, and A. Alvarado, *Scientific Reports* **11** (2021), 10.1038/s41598-021-90055-3.
- [23] E. Bai, X.-q. Jiang, and Y. Wu, *Electronics* **11**, 377 (2022).
- [24] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, “Falcon,” (2020).
- [25] Thomas Prest, “Falcon, Python Implementation,” (2020).
- [26] The Python implementation was slightly modified to separate the public and private keys.
- [27] NIST, “Pqc standardization process: Announcing four candidates to be standardized, plus fourth round candidates,” .
- [28] Python Software Foundation, “Python,” .
- [29] Tom Preston-Werner, “Tom’s obvious minimal language (toml),” .
- [30] Brett Cannon, Nathaniel Smith, Donald Stufft, “PEP 518 - Specifying Minimum Build System Requirements for Python Projects,” (2016).
- [31] Taneli Hukkinen, Shantanu Jain, “PEP 680 - tomllib: Support for Parsing TOML in the Standard Library,” (2022).
- [32] Nathaniel J. Smith, “Comparison of configuration file languages,” (2016).
- [33] The symbols can also be loaded from a file.
- [34] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, *Nature* **585**, 357 (2020).
- [35] K. Kikuchi, *Journal of Lightwave Technology* **34**, 157 (2016).
- [36] P. Jouguet, S. Kunz-Jacques, and A. Leverrier, *Physical Review A* **84** (2011), 10.1103/physreva.84.062317.
- [37] A. Leverrier, *Theoretical study of continuous-variable quantum key distribution*, Theses, Télécom ParisTech (2009).
- [38] J. Lodewyck, M. Bloch, R. García-Patrón, S. Fossier, E. Karpov, E. Diamanti, T. Debuisschert, N. J. Cerf, R. Tualle-Brouri, S. W. McLaughlin, and P. Grangier, *Physical Review A* **76** (2007), 10.1103/physreva.76.042305.
- [39] S. Fossier, E. Diamanti, T. Debuisschert, R. Tualle-Brouri, and P. Grangier, (2008), 10.1088/0953-4075/42/11/114014, *J. Phys. B* **42**, 114014 (2009), arXiv:0812.4314v1 [quant-ph].
- [40] Y. Pi, H. Wang, Y. Pan, Y. Shao, Y. Li, J. Yang, Y. Zhang, W. Huang, and B. Xu, *Optics Letters* **48**, 1766 (2023).
- [41] Alexander Franzen, “gwoptics components library,” (2012).

Appendix A: Calibration of Alice

The calibration of the optical setup of Alice consists mainly in the calibration of the conversion factor r_{conv} . This is done by directly connecting the input of Alice's VOA to the laser, and adding a powermeter at the output of Alice. While the power is kept constant, the attenuation of the VOA is changed and, for each attenuation, the power on the monitoring photodiode and the one on the powermeter at Alice's output are recorded and r_{conv} is measured by a linear fit of the output power to the monitoring power. This method has several advantages: first r_{conv} doesn't depend on the VOA (this is due to the fact that the monitoring photodiode is after the VOA) and r_{conv} is estimated without changing the optical connection (the only change in the optical connections happens before the region of r_{conv}). In our setup the measured value of r_{conv} was around $4.4 \cdot 10^{-3}$, which is compatible with $r_{\text{conv}}^{\text{ideal}} = \frac{0.05}{0.95} \cdot 0.1 \simeq 5.3 \cdot 10^{-3}$.

A characterization of Alice's VOA is necessary to know the relation between the applied voltage and the attenuation. In the experiment, a voltage of 2.9 V is applied, corresponding to an attenuation of 3.78 dB. Both characterizations can be done using a script in the qosst-alice package.

Appendix B: Calibration of Bob

The calibration of Bob is mainly composed of the measurement of the detector efficiency η , which takes into account the global efficiency of the receiver on the quantum signal path (from the input of the quantum signal to the photodiodes). For a single photodiode, the efficiency is defined as the ratio between the number of emitted electrons and the number of received photons, which reaches $\eta_{PD} = 1$ for the ideal photodiode that emits one electron for each incoming photon. One can relate the efficiency of a photodiode to its responsivity $\mathcal{R} = I/P$ (*i.e.* the ratio between the photocurrent and the optical power) as

$$\eta_{PD} = \frac{n_e}{n_p} = \frac{\frac{I}{e}}{\frac{P}{\frac{hc}{\lambda}}} = \frac{hc}{\lambda e} \mathcal{R} \stackrel{\lambda=1550 \text{ nm}}{\simeq} \frac{\mathcal{R}}{1.25 \text{ A/W}}. \quad (\text{B1})$$

For a balanced detector, the responsivity is usually defined as the sum of the two photocurrents divided by the input power at the beginning of the interferometer. For the whole detector, we can compute the responsivity as the sum of all photocurrents divided by the input optical power, and then divide this value by 1.25 A/W to get the efficiency. In our setup, the Thorlabs balanced detector has two voltage monitoring outputs V_+ , V_- , such that

$$\begin{aligned} V_+ &= G_m \cdot I_+, \\ V_- &= G_m \cdot I_-, \end{aligned} \quad (\text{B2})$$

where I_+ and I_- are the photocurrents, $G_m = 10 \text{ kV/A}$ is the monitoring gain and V_+ and V_- are the voltages of the monitoring outputs. Hence

$$\eta = \frac{hc}{\lambda e} \cdot \frac{V_+ + V_-}{G_m \cdot P_{sig}}, \quad (\text{B3})$$

where P_{sig} is the optical power at the signal input of Bob. In practice, this calibration is done by putting a laser at a power of a few mW, followed by a VOA, and a 50:50 beam splitter with one output on a powermeter and the other on the signal input of Bob. The attenuation of the VOA is changed and for each attenuation the value of V_+ , V_- and P_{sig} are recorded. The detector efficiency η is measured by linear fit using the formula in Eq. (B3).

The value of η changes between the setup with the emulated channel using the VOA and the one with the fiber spool, since in the latter a manual polarisation controller is added to Bob's setup. For both cases, η is estimated using an automated script available in the qosst-bob package, giving $\eta_{voa} = 55.4\%$ and $\eta_{fiber} = 41.3\%$. We can compare those values with the expected efficiency of the receiver. The responsivity of the photodiodes in the balanced detector is typically around 0.9 A/W at 1550 nm (verified by measurement of the balanced detector only), giving an efficiency of $\eta_{PD} = 72\%$. The excess losses are typically 7% (beam splitter), 15% (switch), 10% (polarisation maintaining mating sleeves, 2 in total), giving $\eta_{typ} = 46\%$. Monitoring the photocurrents is crucial in order to get a good estimation of the detector efficiency η , which can lead to security issues if not properly estimated, in particular in the trusted detector model.

The receiver is also characterized by another important parameter, which is the electronic noise V_{el} (or clearance). The precise value of V_{el} needs to be calibrated for each frame for two reasons: first, the shot noise is calibrated at each step, and as V_{el} is normalised relative to the shot noise, the value changes with the shot noise, and, second, V_{el} is dependent on the value of f_{beat} (and the other frequency parameters). Indeed, the same DSP is applied to the electronic noise samples (and the electronic and shot noise samples) before measuring the noises, and the exact behaviour of the DSP depends on the frequency parameters.

Appendix C: Shot noise units and normalisation

In natural units, the following relations hold in terms of the quadrature operators:

$$\begin{aligned} [\hat{q}, \hat{p}] &= i\hbar, \\ \Delta\hat{q}\Delta\hat{p} &\geq \frac{\hbar}{2}. \end{aligned} \quad (\text{C1})$$

The Shot Noise Units (SNU) are defined by choosing $\hbar = 2$ so that

$$\begin{aligned} [\hat{q}, \hat{p}] &= 2i, \\ \Delta\hat{q}\Delta\hat{p} &\geq 1. \end{aligned} \quad (\text{C2})$$

Using this convention, the shot noise is equal to 1. In practice, we normalize the received quantum symbols by the variance of the shot noise symbols. The electronic and electronic and shot noise symbols are obtained by applying the same DSP on the quantum, the electronic noise and the electronic and shot noise, giving the quantum symbols, the electronic symbols and the electronic and shot noise symbols. The variance of the shot noise is obtained and used for normalization to ensure that the variance of the shot noise is 1.

Appendix D: RF-Heterodyne

RF-heterodyne is a detection scheme that only uses one balanced detector to measure both quadratures. The basic idea is that the data is encoded on a single sideband, by displacing it with a frequency $f_{\text{shift}} > \frac{B}{2} = \frac{(1+\beta) \cdot R_s}{2}$. In this way, the data can be recovered by detecting the signal after a physical homodyne detection and recovering the two encoded quadratures by demodulating the signal by multiplying by the complex exponential at frequency $-f_{\text{shift}}$. In practice, the demodulation frequency is not exactly $-f_{\text{shift}}$ but $-f_{\text{shift}} - f_{\text{beat}}$, where f_{beat} is the frequency difference between the two lasers.

This method has some limitations: one is limited to modulating only one sideband, and needs to ensure that no information is leaking to the other, unmonitored sideband. However, the setup is much simpler, using only one balanced detector, which also simplifies the analysis on the shot and electronic noise. It adds however some complexity in the DSP with the demodulation.

Appendix E: Digital Signal Processing

1. Zadoff-Chu sequence

Here, we extend the explanation of the DSP algorithm, and give a scheme with some examples. The

Zadoff-Chu sequence is defined by the following formula:

$$ZC(n) = \exp\left(-j \frac{\pi R_{ZC} n(n + c_f + 2q)}{L_{ZC}}\right), \quad (\text{E1})$$

with $0 \leq n < N_{ZC}$, R_{ZC} the root of the sequence, L_{ZC} the length of the sequence, q the cyclic shift and $c_f = N_{ZC} \bmod 2$. The Zadoff-Chu sequence requires that $0 \leq R_{ZC} \leq L_{ZC}$ and that R_{ZC} and L_{ZC} are coprimes. In practice, we choose L_{ZC} and N_{ZC} to be prime numbers and $q = 0$, which simplifies the generation formula of the Zadoff-Chu sequence,

$$ZC(n) = \exp\left(-j \frac{\pi R_{ZC} n(n + 1)}{L_{ZC}}\right). \quad (\text{E2})$$

This sequence is called a Constant Amplitude Zero AutoCorrelation (CAZAC) sequence and its good periodic autocorrelation properties make it extremely fit for time synchronisation.

2. Raised Cosine and Root-Raised Cosine

Raised Cosine (RC) filters are a family of filters that minimize Inter-Symbol Interference (ISI), defined by the following frequency domain description

$$H_{rc}(f) = \begin{cases} 1, & |f| \leq \frac{1-\beta}{2} \cdot R_S \\ \frac{1}{2} \left[1 + \cos\left(\frac{\pi}{\beta \cdot R_S} \left[|f| - \frac{1-\beta}{2} \cdot R_S \right] \right) \right], & \frac{1-\beta}{2} \cdot R_S < |f| \leq \frac{1+\beta}{2} \cdot R_S \\ 0, & \text{otherwise,} \end{cases} \quad (\text{E3})$$

where β is the roll-off of the RC filter and R_S is the symbol rate. The bandwidth of this filter is $2 \cdot \frac{1+\beta}{2} \cdot R_S = (1+\beta) \cdot R_S$. Using the time description of the filter, it is possible to see that a perfect sampling will induce to recover one symbol with no interference from any other

symbol. In practice, we want to apply a filter at the transmitter side and a matched filter at the receiver side to optimise the SNR, as the matched filter is the filter that maximizes the SNR in the presence of white noise. For this we use the Root-Raised Cosine filter defined as

$$H_{rc}(f) = H_{rrc}(f) \cdot H_{rrc}(f). \quad (\text{E4})$$

3. Scheme of the Digital Signal Processing

In Fig. 8 we provide a scheme of the signal processing applied in the QOSST software. The Quadrature Phase Shift Keying (QPSK) modulation is used only for display purposes.

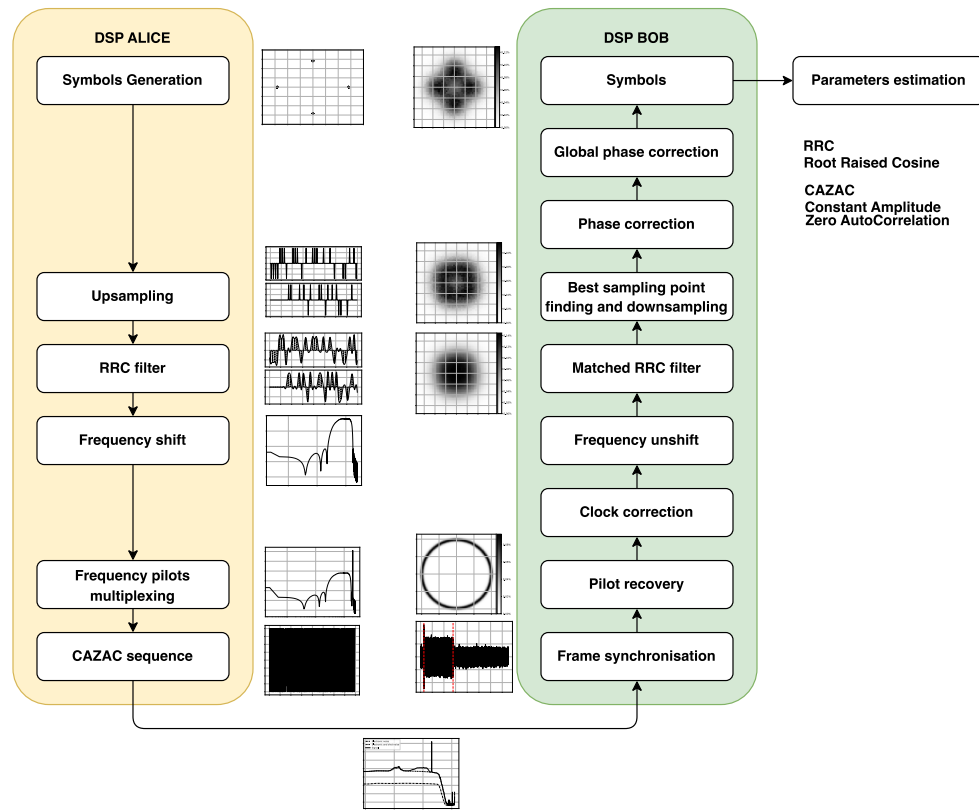


Figure 8. Scheme of the Digital Signal Processing in QOSST.