Fast and Simple Qubit-Based Synchronization for Quantum Key Distribution

Luca Calderaro,[1,2,*] Andrea Stanco,[1,2] Costantino Agnesi,[1,2] Marco Avesani,[1]
Daniele Dequal,[3] Paolo Villoresi,[1,2] and Giuseppe Vallone[1,2,4]

[1] *Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova, via Gradenigo 6B, 35131, Padova, Italy*

[2] *Istituto Nazionale di Fisica Nucleare (INFN)—Sezione di Padova, Via Marzolo 8, 35131, Padova, Italy*

[3] *Matera Laser Ranging Observatory, Agenzia Spaziale Italiana, Matera, Italy*

[4] *Dipartimento di Fisica e Astronomia, Università degli Studi di Padova, via Marzolo 8, 35131 Padova, Italy*

(Received 22 October 2019; revised manuscript received 12 December 2019; accepted 9 April 2020; published 18 May 2020)

We propose Qubit4Sync, a synchronization method for quantum-key-distribution (QKD) setups, based on the same qubits exchanged during the protocol and without requiring additional hardware other than that necessary to prepare and measure the quantum states, in a similar fashion to the clock recovery used in classical communications. Our approach introduces a cross-correlation algorithm with low computational complexity for high channel losses. We test the robustness of our scheme in a real QKD implementation, and we believe it may find application in other quantum communication protocols.

DOI: 10.1103/PhysRevApplied.13.054041

I. INTRODUCTION

Quantum key distribution (QKD) constitutes a promising technology for the security of future communication networks. Introduced in 1984 [1], QKD is a communication protocol for the generation of a secret key shared only by two parties, which afterward can be used to establish a secure communication. The selling point of QKD is that the security of the protocol is guaranteed as long as the laws of quantum mechanics are valid. This is a great leap forward compared to similar classical protocols that are based on the limited computational power of the adversary. The practical implementation of the protocol has developed to the point at which several experiments have been performed, exploiting deployed telecom fibers [2], the daylight free-space channel in urban areas [3–5], and the satellite-to-ground channel [6,7]. Nonetheless, there still remain several challenges to be addressed, such as the communication rate and range, and making QKD systems low cost, compact, and robust [8].

Clock synchronization is crucial for communication networks [9–12], QKD being no exception. Indeed, it is fundamental in QKD protocols not only because it allows the two parties to correctly generate the secret key but also to filter out the noise. Knowledge of the time at which the signal is expected to arrive at the receiver allows us to discard the majority of the detection due to noise, increasing the signal-to-noise ratio (SNR). This is of crucial importance, as the SNR is usually the limiting factor for the performance of QKD systems. The solutions that are usually adopted in current QKD implementations include either sending a decimated copy of the transmitter's clock through a separated single-mode fiber [13] or even the same quantum channel [14–19], or locking the two clocks to an external time reference provided, for instance, by global-navigation-satellite-system (GNSS) receivers [3,20,21]. All these solutions imply the use of additional hardware and hence an increase in the complexity and cost of the setup.

Most of the modern classical communication protocols do not require the sharing of an external clock reference signal, as the clocks' synchronization is recovered by the data stream itself. Among several clock-recovery techniques, self-synchronizing codes are worth mentioning, as they are the standard for Ethernet communication over fiber-optics or copper cable [22]. These methods have found applications in every classical communication system that requires high data throughput, from universal-serial-bus (USB) to fiber-optic communication [23–25]. The clear advantages of these clock-recovery techniques are as follows: (i) data throughput is maximized because any physical channel is used for the data stream; and (ii) communications on different physical channels [e.g., dense-wavelength-division-multiplexing (DWDM) channels in fiber-optic communication] are completely independent from each other.

In this work, we propose Qubit4Sync, a synchronization system that only uses the same qubits exchanged during

[*] luca.calderaro@unipd.it

the QKD protocol, without requiring additional hardware. Our approach is to exploit the information on the measurements that the receiver performs on the qubits, in a similar fashion to the clock recovery used in classical communications. Hence, a preanalysis is performed before the standard QKD postprocessing, extracting the information on the time of arrival of the signal. The synchronization method introduced here has been successfully implemented in a recent complete QKD experiment described in Agnesi *et al.* [26] and we believe it may find application in other quantum communication protocols, inheriting the advantages that self-synchronizing codes offer to classical communications.

II. DESCRIPTION OF THE ALGORITHM

Alice transmits a qubit string (the raw key) encoded in the state of a train of attenuated optical pulses. The time between two consecutive qubits, τ^A, is set by Alice's clock. On the other side, Bob receives some of the qubits (due to losses), analyzes their state, and uses his clock to measure their time of arrival. We consider the case in which Alice's and Bob's clocks may have a time bias as well as a relative drift in time of their frequencies. This implies that Bob may measure a different time τ^B between subsequent qubits.

Bob's goal is to determine the position of the detected qubits in Alice's raw key: this operation is needed to correctly generate the sifted key and to perform the parameter estimation and the subsequent postprocessing. The above problem can be reformulated as follows: Bob needs to determine the expected time of arrival (measured by his clock) of the qubits sent by Alice—namely, he needs to solve two tasks:

(i) *Period recovery*: to recover the period τ^B from the obtained detections.

(ii) *Time-offset recovery*: to determine the time delay between the measured and sent sequences.

Step (i) is needed to correctly reconstruct the separations in the raw key between consecutive detections. Step (ii) is needed to associate each detection with the corresponding bits in Alice's raw string.

This problem can be solved by synchronizing Alice's and Bob's clocks and by knowing the time of flight of the qubits [11]. However, Bob just needs to know at what time Alice's pulses will arrive and not the time at which she sent them. Therefore, their clocks may be synchronous up to a time offset.

We define t_a^m as the measured time of arrival (according to Bob's clock), with $a \geq 1$ enumerating the obtained detections. Since the time separation between the qubits is constant at Alice's site, a model that reproduces the expected time of arrival t_a^e at Bob's site can be expressed as

$$t_a^e = t_0 + n_a\tau^B + \epsilon_a, \quad n_a \in \mathbb{N}. \tag{1}$$

The index n_a identifies the position of the sent qubit in Alice's raw key and t_0 is the expected time of arrival of the first pulse sent by Alice, while ϵ_a is a normal random variable with zero mean and variance σ^2 (due to measurement jitter). If Alice's and Bob's clocks are perfectly synchronized, then $\tau^B = \tau^A$. We note that we are neglecting noise in the model.

We define the *time-error* function $\mathcal{E}_a$ between the measured and expected times of arrival as

$$\mathcal{E}_a = t_a^m - t_a^e. \tag{2}$$

The time-error variation between two different detections a and $a+b > a$ is the so-called *time-interval error* $\mathcal{E}_a^I(b)$, defined as [10]

$$\mathcal{E}_a^I(b) = \mathcal{E}_{a+b} - \mathcal{E}_a. \tag{3}$$

Below, we describe how the above two tasks (frequency and time-offset recovery) can be realized by using only the qubits exchanged during the QKD protocol, without requiring additional hardware.

A. Period recovery

We first describe how the period τ^B can be obtained by Bob. Let us suppose that Bob acquires data for a time T_{acq} and in this time the relative frequencies of Alice's and Bob's clocks are constant; namely, the periods τ^A and τ^B are constant. Typical values of T_{acq} are of the order of 1 s. The above acquisition corresponds to M pulses sent by Alice with $M = \lfloor T_{\text{acq}}/\tau^B \rfloor$, of which D are the ones detected by Bob. We can label the detected pulses with the index $b = 1, \ldots, D$ such that $t_{a+b}^m - t_a^m < T_{\text{acq}}$, t_a^m being the last detection before the acquisition starts. We define the condition of successful period recovery when the following condition is satisfied:

$$|\mathcal{E}_a^I(b)| < \frac{\tau^B}{2}, \quad \text{for all } b = 1, \ldots, D. \tag{4}$$

The latter condition implies that the M subsequent pulses sent by Alice during the acquisition correspond to exactly M time slots of length τ^B on Bob's clock. We note that Eq. (4) is a sufficient condition to correctly reconstruct the separations, $n_{a+b} - n_a$, in the raw key between consecutive detections, but it is not the optimal one as the signal may arrive at any time inside the time slot τ^B. This would prevent filtering out of the noise. The optimal value for τ^B

is the one such that

$$\frac{1}{D}\sum_{b=1}^{D}|\mathcal{E}_a^I(b)|^2 \simeq \sigma^2. \tag{5}$$

To make a first guess τ_0^B about the value of τ^B, Bob should perform a Fourier analysis of the times-of-arrival signal [27]. The latter is a sequence of N symbols (taking value 0 or 1), with the ones corresponding to the detection times. Assuming that Alice's clock frequency is less than twice that of Bob, we may sample the time of arrival of the photons with a $4/\tau^A$ sampling rate [since the sample is real valued over half of the spectral range for which the discrete Fourier transform (DFT) is meaningful]. For the purpose of real-time analysis (i.e., to speed up computation), we perform the fast Fourier transform (FFT) limiting the number of samples to $N = 10^6$ [27]; namely, we limit the sampling time for the FFT to $T_{\text{samp}} = N(\tau^A/4)$.

The above FFT will provide an estimate τ_0^B of τ^B with an error of approximately $4\tau^A/N$. We note that τ_0^B satisfies Eq. (4) for the first $b = 1, \ldots, D_0$ detections, such that $t^m_{a+b} - t^m_a < T_{\text{samp}}$. However, if the acquisition time T_{acq} is larger than T_{samp} (i.e., $M > N/4$), the estimate τ_0^B may not be sufficiently accurate and the condition given in Eq. (4) may not be satisfied.

Instead of performing a Fourier transform of $4M$ samples (which could increase the computational complexity), we perform a linear regression of $\bmod_{\tau_0^B}(t^m_{a+b})$ as a function of the measured time t^m_{a+b}, for $b = 1, \ldots, D_0$. We use a least-trimmed-squares algorithm as a robust statistical method against background [28]. While the intercept does not provide any useful information, it is easy to prove that the slope of the linear model is equal to $(\tau^B - \tau_0^B)/\tau^B$, with which we have an estimate of τ^B such that Eq. (5) is satisfied.

Once τ^B has been identified, Bob can associate each detection with a different slot of length τ^B, indicated by the indices n_{a+b}, up to a constant (depending on t_0^e). Indeed, Bob can calculate all the index differences by using the relation $n_{a+b} - n_a = [(t^m_{a+b} - t^m_a)/\tau^B]$.

We note that the variation of τ^B during a given acquisition time T_{acq} should be small in order to guarantee that Eq. (5) can be satisfied; namely, $M|\delta\tau^B| = |\partial\tau^B/\partial t|T^2_{\text{acq}}/\tau \lesssim 10\sigma$. If the latter condition is not satisfied, the period recovery should be performed by reducing the acquisition time T_{acq}.

In the next acquisition of D' pulses, the value of τ^B may change due to a relative frequency shift of the clocks. The condition becomes $\sum_{b=1}^{D'}|\mathcal{E}^I_{a+D}(b)|^2 \simeq D'\sigma^2$, which is satisfied by again applying the above analysis.

B. Time-offset recovery

We now describe how the initial delay t_0 can be estimated, which allows us to determine the values of the index n_a. We restate that once period recovery has been performed, Bob has correctly estimated τ^B and the index differences $n_{a+1} - n_a$ for $a \geq 1$. Then, only the first index n_1 is needed to calculate the set of indices $\{n_a\}$. We note that n_1 is related to t_0 by $t_0 = t_1^e - n_1\tau^B$.

Due to losses in the channel, there is a high probability that the first pulse is not detected by Bob. Moreover, the presence of background means that it is not straightforward to distinguish the detection from Alice's qubit.

As a first guess, we may identify Bob's first detection as the first pulse sent by Alice (i.e. $n_1 = 0$). The first detection can be identified by looking for a rising edge of the detection frequency. If the overall transmittance of the system is η and the mean number of photons per pulse sent by Alice is $\mu \sim 1$, Bob expects to make a detection during each of $1/\eta$ pulses. Therefore, the uncertainty on n_1 is of the order of $1/\eta$.

To precisely determine t_0^A, our approach is to calculate the correlation between the signal received by Bob with a synchronization string s^A that has length $L \gg 1/\eta$. The string s^A, which is also known to Bob, is placed at the beginning of Alice's raw string. We encode s^A in the base that is more frequently measured by Bob (say, the Z basis) and we assign the values $+1$ or -1 to the two orthogonal states of such a basis. In the event of no detection, we assign the value 0. Then, once Bob has determined the period τ^B and has a first guess about t_0 (hence n_1), he can produce a string s^B with values 0, -1, or $+1$. In order to precisely determine t_0, we may exploit the cross-correlation between the signal received by Bob and s^A: indeed, the value that maximizes the cross-correlation corresponds to the required offset.

Here, we recall that the cross-correlation function between s^A and s^B is defined as $(m = 0, \ldots, L-1)$

$$x_m^{AB} = \frac{1}{L}\sum_{n=0}^{L-1} s^{*A}_{n+m}s^B_n, \tag{6}$$

with the convention that $s^A_{n'} = s^A_{n'-L}$ for $n' \geq L$. The offset between Alice's and Bob's strings is

$$\mathcal{E}_0/\tau^B = n_1 = m_{\text{opt}}, \tag{7}$$

where m_{opt} is the value of m that maximizes the cross-correlation x_m^{AB}. By exploiting the convolution theorem, the maximum of the cross-correlation x_m^{AB} can be evaluated by Bob with $\mathcal{O}(L\log_2 L)$ operations (we are assuming that the Fourier transform of s^A is already known by Bob before the measurements). Below, we propose an algorithm that reduces computational the computational complexity.

Our method is based on the properties of particular synchronization strings, which allow us to calculate the cross-correlation more efficiently. More precisely, we need to use a synchronization string such that its autocorrelation

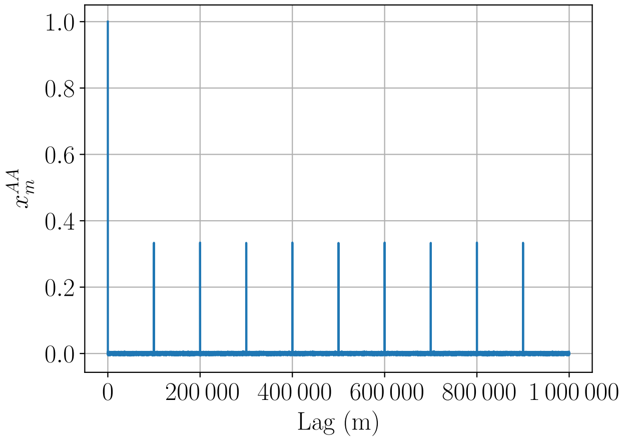

FIG. 1. An example of autocorrelation x_m^{AA} for a synchronization string with $L = 10^6$, $N_1 = 10$, and $c_0 = \frac{1}{3}$.

function x_m^{AA} has N_1 periodic peaks; namely, it satisfies

$$\begin{aligned} x_0^{AA} &= 1,\\ x_{jL_1}^{AA} &\simeq c_0, \quad \text{for } j > 0,\\ x_{u+jL_1}^{AA} &\simeq 0, \quad \text{for } u > 0, \end{aligned} \tag{8}$$

where $0 < c_0 < 1$, $L = N_1 L_1$, N_1 and L_1 being integer numbers, $u = 0, \ldots, L_1 - 1$, and $j = 0, \ldots, N_1 - 1$. The method to generate a string s^A that satisfies Eq. (6) is described in Appendix A. Figure 1 shows the autocorrelation x of such a string with $L = 10^6$ and $N_1 = 10$. We leave the study of the optimal value of c_0 in the function of losses and errors for future investigation.

To simplify the computational complexity, we may exploit the periodicity of the autocorrelation. We need to first calculate the interleaved sum of x_m^{AB}, defined as $(1/N_1)\sum_{j=0}^{N_1-1} x_{u+jL_1}^{AB}$. To do so, we need to evaluate S^A (S^B), the interleaved DFT of s^A (s^B):

$$S_{r,j}^{A} = \sum_{k=0}^{N_1-1} s_{r+kL_1}^{A} e^{-(2\pi i/N_1)jk}, \tag{9}$$

where $r = 0, 1, \ldots, L_1 - 1$ and $j = 0, 1, \ldots, N_1 - 1$. The index j run through the frequency domain, but the time domain is still present due to the index r. We note that the above operation corresponds to reshaping the sequence s into an $L_1 \times N_1$ matrix and calculating the FFT for each row (see Fig. 2). Therefore, we can define a cross-correlation in the time domain of S^A and S^B for $u = 0, 1, \ldots, L_1 - 1$:

$$X_{u,j}^{AB} = \frac{1}{L_1}\sum_{r=0}^{L_1-1} (S_{r+u,j}^{A})^{*} S_{r,j}^{B}. \tag{10}$$

We note that the Fourier coefficients $S_{r,j}$ are defined for $r = 0, \ldots, L_1 - 1$. However, by extending the original definition (9), it is possible to define them for larger values of r, by means of the recursive relation $S_{r+L_1,j} = S_{r,j}\, e^{(2\pi i/N_1)j}$.

In Appendix B, we prove the following lemma.

Lemma 1. *The cross-correlation* $X_{u,j}^{AB}$ *is related to the cross-correlation* $x_{u+jL_1}^{AB}$ *by a* DFT :

$$x_{u+jL_1}^{AB} = \sum_{k=0}^{N_1-1} e^{-(2\pi i/N_1)jk} X_{u,k}. \tag{11}$$

The interleaved sum of x_m^{AB} can be easily evaluated by using Eq. (11):

$$\frac{1}{N_1}\sum_{j=0}^{N_1-1} x_{u+jL_1}^{AB} = X_{u,0} = \frac{1}{L_1}\sum_{r=0}^{L_1-1} (S_{r+u,0}^{A})^{*} S_{r,0}^{B}. \tag{12}$$

By defining $m_{\text{opt}} = u_{\text{opt}} + j_{\text{opt}} L_1$, we may first determine u_{opt} by maximizing $X_{u,0}$. Indeed, due to the periodicity of the autocorrelation, the correlation $X_{u,0}$ presents a single peak for $u = u_{\text{opt}}$; namely, we will have $X_{u_{\text{opt}},0} \simeq c_0 + (1 - c_0)/N_1$ while $X_{u \neq u_{\text{opt}},0} \simeq 0$. Thus, u_{opt} is the index that maximizes the averaged cross-correlation $X_{u,0}$.

The above relation provides a method to find the position of the N_1 peaks of x_m^{AB}, which are located at positions $m = u_{\text{opt}} + jL_1$. To find j_{opt}, we can now use Eq. (11) to calculate (and maximize) the cross-correlation only in such N_1 equally separated points, $x_{u_{\text{opt}}+jL_1}^{AB}$ for $j = 0, \ldots, N_1 - 1$.

1. Computational complexity

The algorithm to calculate the offset can be visualized in Fig. 2. Alice's and Bob's strings, s^A and s^B, are rearranged into two matrices, with L_1 rows and N_1 columns. For each row, the FFT is calculated to find the matrices S^A and S^B. S^A can be calculated in advance, hence we consider just the computational costs for Bob's string, which amount to $\mathcal{O}(L_1 N_1 \log_2 N_1)$. At this point, we apply Eq. (12) and calculate the cross-correlation $X_{u,0}$ between the first columns of S^A and S^B. This operation can be carried out with the FFT, for a computational cost of $\mathcal{O}(L_1 \log_2 L_1)$. We then find the position u_{opt} that maximizes $X_{u,0}$.

We then evaluate $X_{u_{\text{opt}},j}$ by means of Eq. (10) for $j = 1, \ldots, N_1 - 1$ (the $X_{u_{\text{opt}},0}$ have been already calculated) with a computational cost of $\mathcal{O}(L)$. Finally, by using Lemma 1, we use a FFT to calculate $x_{u_{\text{opt}}+jL_1}^{AB}$ and its maximum with $\mathcal{O}(N_1 \log_2 N_1)$ operations. To summarize, the overall computational cost is $\mathcal{O}[(L + N_1)\log_2 N_1 + (L/N_1)\log_2(L/$

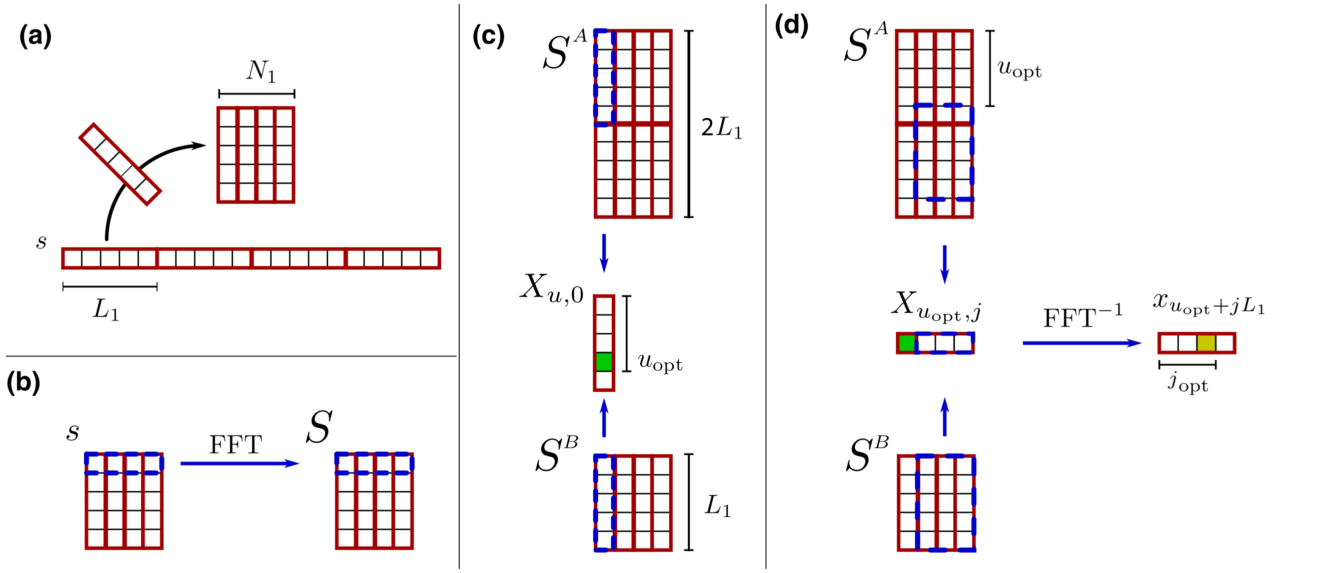

FIG. 2. (a) Alice's string, s^A, and Bob's string, s^B, are divided into N_1 blocks of length L_1 and reshaped into an $L_1 \times N_1$ matrix. (b) For each row of the matrix, the FFT is calculated, thus obtaining the matrices S^A and S^B. Note that S^A can be calculated in advance. (c) The cross-correlation $X_{u,0}$ of the first columns of S^A and S^B is calculated. The position $u_{\rm opt}$ that maximizes $\hat{X}_{u,0}$ corresponds to the position of the first peak of the cross-correlation X. (d) Consider the block of S^A shifted by $u_{\rm opt}$ rows and calculate the cross-correlation between the remaining columns of S^A and S^B. The resulting vector $X^{AB}_{u_{\rm opt},j}$ is antitransformed so as to obtain $x^{AB}_{u_{\rm opt}+jL_1}$. The $j_{\rm opt}$ that maximizes $x^{AB}_{u_{\rm opt}+jL_1}$ provides the position of the major peak among the smaller peaks.

$N_1)]$, which can be optimized by choosing $N_1 = \log_2(L)$, resulting in

$$\mathcal{O}[L \log_2(\log_2 L)]. \tag{13}$$

Compared to other algorithms [29–31], the better efficiency comes with the disadvantage of a synchronization string satisfying Eq. (6). Therefore, this approach cannot be applied to pseudorandom strings.

We note that our protocol shares some steps of the QuickSynch algorithm proposed in Ref. [29]. In particular, a similar method to obtain $u_{\rm opt}$ is used in Ref. [29]. However, since in Ref. [29] a pseudorandom string s^A is used, the autocorrelation has a single peak and $X_{u_{\rm opt}} \simeq 1/N_1$: this is why in Ref. [29] it is suggested that N_1 repetitions of the s^B string should be collected to be able to determine the peak of $X_{u,0}$. Moreover, in Ref. [29], to estimate $j_{\rm opt}$, the correlation $x^{AB}_{u_{\rm opt}+jL_1}$ is approximated by summing only L_1 points [namely, the authors calculate, for all j's, the quantity $\tilde{x}^{AB}_{u_{\rm opt}+jL_1} = (1/L_1)\sum_{r=0}^{L_1-1} s^A_{r+u_{\rm opt}+jL_1} s^B_r$], while our method exploits relation (11) to calculate it efficiently and exactly.

III. EXPERIMENT AND RESULTS

We test the Qubit4Sync algorithm in a QKD setup, illustrated in Fig. 3. The quantum states are encoded in the polarization of the attenuated laser pulses. Their polarization is modulated by a source exploiting a scheme for polarization encoding based on a Sagnac loop (POGNAC) [32], controlled by a Zynq-7000 system on a chip (SoC) which comprehends both a CPU and a field programmable gate array (FPGA) (SoC, manufactured by Xilinx). Alice's time reference is given by a 10-MHz reference signal to which the FPGA is locked. The repetition rate of the train of pulses is 50 MHz, with a period of $\tau^A = 20$ ns in Alice's time. At the receiver side, a passive state analyzer performs the measurement on the polarization and four superconducting-nanowire single-photon detectors (SNSPDs) generate an electrical signal by the arrival of the optical pulse. A time-to-digital converter (TDC) measures the time of arrival, with an 81-ps time resolution. We do not provide any external time reference to the TDC but just its own internal clock. Then, the software processes the times of arrival every $T_{\rm acq} = 1$ s of acquisition time, analyzing the frequency of the qubits. The offset

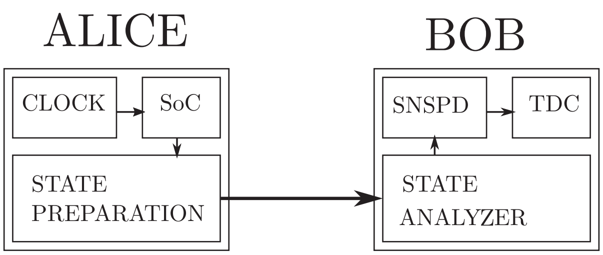

FIG. 3. The experimental setup.

analysis is performed just once with the data of the first second of acquisition.

Figure 4 shows the $\mathcal{E}^I$, the time error between Alice's and Bob's clocks, after an interval of T_{acq}, in the case in which Bob is not correcting his clock [i.e., using $\tau^B = 20$ ns in Eq. (1)]. The graph shows that Alice's and Bob's clocks accumulate a mean time error of about 0.5 ms in every interval of 1 s. This violates Eq. (4) as we have $\mathcal{E}^I_a \gg \tau^B$ and hence, if a period analysis is not performed, a correct separation in the raw key between consecutive detections cannot be achieved. We note that, with such a $\mathcal{E}^I$, correct separation can be achieved only if Alice sends her pulses with $\tau^A > 0.5$ ms. Despite the large mean $\mathcal{E}^I$, the fluctuation around the mean is limited by 2 ns over 400 s.

We implement the three-polarization-states version of the efficient Bennett and Brassard protocol (BB84) [33], in which the receiver measures the polarization on the Z and X bases with 90% and 10% probability, respectively. The synchronization string, s^A, sent by Alice is entirely encoded in the Z basis, so 90% of it is decoded in the right basis (sifted). For the purpose of the synchronization algorithm, just the number of sifted bits at Bob's side matters. Hence, we will talk about the overall transmittance η as the ratio between the number of sifted bits at Bob's side and the number of pulses sent by Alice. The string sent by Alice is composed of a synchronization string, followed by random bits obtained from the quantum random-number generator described in Ref. [34]. We choose a number of states in the synchronization string s^A of $L = 10^6$ (corresponding to the first 20 ms of data acquisition), divided into $N_1 = 10$ blocks. If η is the overall transmittance, the number of synchronization states received by Bob is $L\eta$. Therefore, assuming a zero quantum bit error rate (QBER) and background noise, the maximum correlation value is $\simeq \eta$, while the standard deviation of the correlation for other lags is $\simeq \sqrt{\eta/L}$. The distinguishability, Δ, of the maximum correlation peak among the others is given by the ratio of the former and the latter, $\Delta \simeq \sqrt{L\eta}$. We set a threshold on the distinguishability of $\Delta \geq 10$ as successful detection of the maximum correlation. Hence, for our choice of L, the algorithm can cope with overall losses up to 40 dB. On the other hand, for other values of the overall transmittance η, the minimum string length necessary to achieve synchronization is $L \simeq 100/\eta$. In practice, the presence of background and misalignment between the transmitter and the receiver lowers the maximum losses that the algorithm can handle.

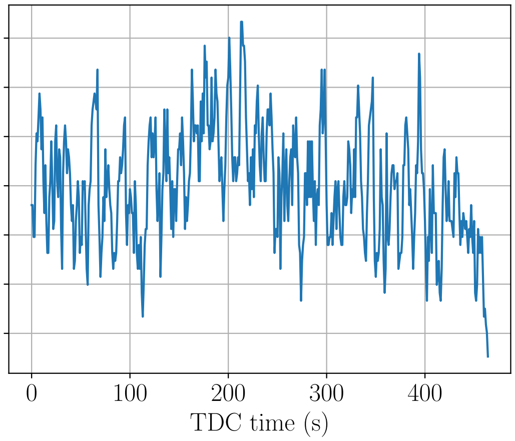

FIG. 4. The $\mathcal{E}^I$ between Alice's and Bob's clocks, after an interval of 1 s, without Bob changing his clock pace [i.e., using $\tau^B = 20$ ns in Eq. (1)]. The mean value of the $\mathcal{E}^I$ is $\overline{\mathcal{E}^I} \simeq -0.503$ ms.

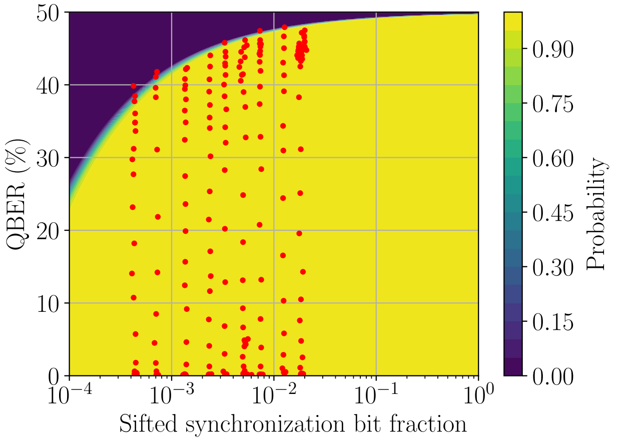

FIG. 5. The robustness of the time-offset analysis for different values of the QBER and a sifted synchronization bit fraction. The plot shows the probability of correctly detecting the time offset, which is estimated through simulation of the algorithm. The red dots correspond to successful synchronization with data generated by our setup. The synchronization string used for this figure has length $L = 10^6$.

We test the robustness of the offset analysis by tuning the QBER and the number of bits of s^B. We use strings generated from several QKD runs as well as simulations of the experiment. In particular, the simulation takes into account the losses and misalignment of the setup but not the presence of the background and dark counts. Figure 5 shows the results of the simulation: the probability of correctly detecting the time offset. As regards the strings generated by the QKD setup, the red dots show when the analysis is successful, satisfying Eq. (5) and correctly finding the time offset that maximizes the correlation.

As expected, the simulation shows a good outcome of the analysis up to 10^{-4} sifted synchronization bit fraction, which is the fraction of the synchronization string received in the correct base (Z). This is no longer true for high values of the QBER. Over 30% of QBER, the algorithm needs more bits in s^B to contrast the reduction of the maximum correlation due to the multiple bit flips. The background detection comes into play in the experimental runs, reducing the amount of losses that the algorithm can tolerate. In our case, the analysis fails below a sifted synchronization bit fraction of $3 \cdot 10^{-4}$, with 200 Hz of free-running background detection rate. The robustness to the QBER is comparable to that obtained with the simulated strings. The comparison is limited to a ratio of about $3 \cdot 10^{-2}$ due to the maximum event rate that our TDC can process. It is interesting to note the very high robustness to the QBER, well above the threshold required to establish a secure channel. In fact, a very rough alignment between the transmitter and the receiver is sufficient for the synchronization to take place. This implies that the precise alignment of the receiver and transmitter may be realized after the synchronization phase, maybe using the same states sent by Alice and without the use of external references that require additional lasers and detectors.

IV. CONCLUSIONS

We introduce Qubit4Sync, a synchronization procedure only requiring the same photons encoding the quantum states that are exchanged in a QKD protocol. Moreover, we develop a fast cross-correlation algorithm. The common solution to synchronizing two terminals in a QKD setup includes either an additional pulsed laser or two GPS receivers. This work simplifies the practical implementation of a QKD setup because it avoids the use of the additional hardware required for a synchronization subsystem, meaning cheaper apparatus and a lower failure probability due to hardware.

We note that while QKD requires classical communication for the postprocessing, the latter cannot be straightforwardly used for QKD synchronization for several reasons. (1) Classical communication systems do not use external synchronization systems, since they implement self-synchronizing codes. Hence, an external synchronization system is not readily available. (2) The use of classical systems for QKD synchronization requires a physical connection between “classical” and “quantum” hardware, as well as proper modification of classical transceivers. (3) Classical and quantum communication can be realized on different physical channels (i.e., fiber and radio link). (4) For QKD implementation, it is not required that the classical postprocessing is synchronous with respect to the quantum communication. Thus, while classical communication is required for QKD, extra features are needed in order to use it for the synchronization of quantum communication.

Qubit4Sync allows a “software”-based synchronization, in the same spirit as self-synchronizing codes. With Qubit4Sync, the faithful exchange of qubits does not require any additional clock-distribution systems, making quantum communication completely independent of any classical communication service. Indeed, with Qubit4Sync, no physical connection between the classical and quantum communication systems is required and they become completely independent from the hardware point of view. We believe this is a step forward toward the sustainable integration of QKD systems with the classical communication network. In a point-to-point fiber-optics link, the communication channel, used for the synchronization service of quantum communication in the previous solutions, can now be used either to enhance the data throughput for classical communication or to add a parallel quantum channel for an independent QKD system, so as to enhance the qubit stream throughput.

Even though our procedure uses the qubits exchanged in the QKD protocol, the security is not undermined or weakened. The shared synchronization string is not used as part of the secure key, whereas the frequency analysis just uses the information on the time of arrival and not that on the qubit state. The synchronization algorithm is also robust against eavesdroppers’ denial-of-service attacks, since the QKD fails before the synchronization. Indeed, if an adversary tries to intercept the qubits, the QKD protocol will stop when the QBER is above 11% [1,33].

In our experiment, we are limited to the use of a passive receiver, since the algorithm is implemented at the software level and is running on a CPU. Therefore, the output of the analysis cannot be used to provide proper feedback to directly control the modulator of an active receiver. However, we note that a hardware implementation on a FPGA could allow the correct timing of the receiver modulation.

Our cross-correlation algorithm may be applied to GPS receivers, the task of which is to correlate the signal sent by the satellite so as to lock to its clock.

Finally, we believe that Qubit4Sync may be applied in other quantum communication protocols in which an exchange of qubits is involved, such as quantum direct communication [35] or quantum bit commitment [36].

ACKNOWLEDGMENTS

Part of this work was supported by the Italian Ministry of Education, University and Research (Ministero dell’Istruzione, Università e della Ricerca) under the initiative “Departments of Excellence” (Law No. 232/2016); the Agenzia Spaziale Italiana (ASI) via the “Q-SecGroundSpace” project (Grant No. E16J1600149000), and the Istituto Nazionale di Fisica Nucleare (INFN) via the “MoonLIGHT-2” project.

APPENDIX A: METHOD FOR THE GENERATION OF THE SYNCHRONIZATION STRING

We use the following method to generate a string s. From a uniform distribution in the $[-1, 1)$ interval, we extract L_1 real numbers x_u, with $u = 0, \ldots, L_1 - 1$, and

L real numbers $y_{u,j}$, with $j = 0, \dots, N_1 - 1$. The synchronization string will take values as follows:

$$s_{u+jL_1} = 2\Theta(y_{u,j} - \lambda x_u) - 1, \tag{A1}$$

where Θ is the Heaviside function and λ is a positive real value. The parameter λ can be used to tune the value of c_0. Indeed, if $\lambda \leq 1$, we have $c_0 = \lambda^2/3$; while if $\lambda > 1$, then $c_0 = 1 - 2/3\lambda$.

APPENDIX B: PROOF OF LEMMA 1

The Fourier coefficients $S_{r,j}$ are defined for $r = 0, \dots, L_1 - 1$. However, from the original definition,

$$S^A_{r,j} = \sum_{k=0}^{N_1-1} s^A_{r+kL_1} e^{-(2\pi i/N_1)jk},$$

it is possible to extend their evaluation for larger values of r. Indeed, we may define

$$S_{r+L_1,j} = S_{r,j}\, e^{(2\pi i/N_1)j}. \tag{B1}$$

The correlation can now be written as

$$\begin{aligned} x^{AB}_{u+jL_1} &= \frac{1}{L} \sum_{k=0}^{N_1-1} \sum_{r=0}^{L_1-1} s^{A*}_{r+u+(k+j)L_1} s^B_{r+kL_1} \\ &= \frac{1}{L} \sum_{k=0}^{N_1-1} \left[\sum_{r=0}^{L_1-u-1} s^{A*}_{r+u+(k+j)L_1} s^B_{r+kL_1} \right. \\ &\quad \left. + \sum_{r=L_1-u}^{L_1-1} s^{A*}_{r+u-L_1+(k+j+1)L_1} s^B_{r+kL_1} \right]. \end{aligned}$$

By using the definition of S, we obtain

$$\begin{aligned} x^{AB}_{u+jL_1} = \frac{1}{L} \sum_{k,\ell_1,\ell_2=0}^{N_1-1} & \left[\sum_{r=0}^{L_1-u-1} S^{A*}_{r+u,\ell_1} S^B_{r,\ell_2} e^{[(-2\pi i)/N_1][(k+j)\ell_1 - k\ell_2]} \right. \\ & \left. + \sum_{r=L_1-u}^{L_1-1} S^{A*}_{r+u-L_1,\ell_1} S^B_{r,\ell_2} e^{[(-2\pi i)/N_1][(k+j+1)\ell_1 - k\ell_2]} \right]. \end{aligned}$$

By using the definition (B1), for which we have $S^A_{r+u,\ell_1} = S^A_{r+u-L_1,\ell_1} e^{(2\pi i/N_1)\ell_1}$, if $r + u \geq L_1$ we obtain

$$\begin{aligned} x^{AB}_{u+jL_1} &= \frac{1}{L} \sum_{k,\ell_1,\ell_2=0}^{N_1-1} \sum_{r=0}^{L_1-1} S^{A*}_{r+u,\ell_1} S^B_{r,\ell_2} e^{[(-2\pi i)/N_1][(k+j)\ell_1 - k\ell_2]} \\ &= \sum_{\ell_1,\ell_2=0}^{N_1-1} \frac{1}{L_1} \sum_{r=0}^{L_1-1} (S^A_{r+u,\ell_1})^* S^B_{r,\ell_2} e^{-(2\pi i/N_1)j\ell_1} \delta_{\ell_1,\ell_2} \\ &= \sum_{k=0}^{N_1-1} e^{-(2\pi i/N_1)jk} \left[\frac{1}{L_1} \sum_{r=0}^{L_1-1} (S^A_{r+u,k})^* S^B_{r,k} \right]. \end{aligned}$$

Finally, from the definition of $X^{AB}_{u,j}$,

$$X^{AB}_{u,j} = \frac{1}{L_1} \sum_{r=0}^{L_1-1} (S^A_{r+u,j})^* S^B_{r,j},$$

we have the lemma:

$$x^{AB}_{u+jL_1} = \sum_{k=0}^{N_1-1} e^{-(2\pi i/N_1)jk} X^{AB}_{u,k}.$$

The inverse relation is

$$X^{AB}_{u,k} = \frac{1}{N_1} \sum_{j=0}^{N_1-1} e^{(2\pi i/N_1)jk} x^{AB}_{u+jL_1},$$

from which we can derive

$$\frac{1}{N_1} \sum_{j=0}^{N_1-1} x^{AB}_{u+jL_1} = X_{u,0} = \frac{1}{L_1} \sum_{r=0}^{L_1-1} (S^A_{r+u,0})^* S^B_{r,0}.$$

[1] C. H. Bennett and G. Brassard, Quantum cryptography: Public key distribution and coin tossing, Theor. Comput. Sci. **560**, 7 (2014).

[2] K.-I. Yoshino, T. Ochi, M. Fujiwara, M. Sasaki, and A. Tajima, Maintenance-free operation of WDM quantum key distribution system through a field fiber over 30 days, Opt. Express **21**, 31395 (2013).

[3] M. Avesani, L. Calderaro, M. Schiavon, A. Stanco, C. Agnesi, A. Santamato, M. Zahidy, A. Scriminich, G. Foletto, G. Contestabile, M. Chiesa, D. Rotta, M. Artiglia, A. Montanaro, M. Romagnoli, V. Sorianello, F. Vedovato, G. Vallone, and P. Villoresi, Full daylight quantum-key-distribution at 1550 nm enabled by integrated silicon photonics, arXiv:1907.10039 (2019).

[4] Y.-H. Gong, K.-X. Yang, H.-L. Yong, J.-Y. Guan, G.-L. Shentu, C. Liu, F.-Z. Li, Y. Cao, J. Yin, S.-K. Liao, J.-G. Ren, Q. Zhang, C.-Z. Peng, and J.-W. Pan, Free-space quantum key distribution in urban daylight with the SPGD algorithm control of a deformable mirror, Opt. Express **26**, 18897 (2018).

[5] S.-K. Liao *et al.*, Long-distance free-space quantum key distribution in daylight towards inter-satellite communication, Nat. Photonics **11**, 509 (2017).

[6] R. Bedington, J. M. Arrazola, and A. Ling, Progress in satellite quantum key distribution, npj Quantum Inf. **3**, 30 (2017).

[7] S. K. Liao *et al.*, Satellite-to-ground quantum key distribution, Nature **549**, 43 (2017).

[8] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, Practical challenges in quantum key distribution, npj Quantum Inf. **2**, 16025 (2016).

[9] J. Bellamy, Digital network synchronization, IEEE Commun. Mag. **33**, 70 (1995).

[10] S. Bregni, Clock stability characterization and measurement in telecommunications, IEEE Trans. Instrum. Meas. **46**, 1284 (1997).

[11] L. Narula, S. Member, and T. E. Humphreys, Requirements for secure clock synchronization, IEEE J. Sel. Topics Signal Process. **12**, 749 (2018).

[12] V. D'Auria, B. Fedrici, L. A. Ngah, F. Kaiser, L. Labonté, O. Alibart, and S. Tanzilli, A universal, plug-and-play synchronisation scheme for practical quantum networks, npj Quantum Inf. **6**, 1 (2020).

[13] B. Korzh, C. C. W. Lim, R. Houlmann, N. Gisin, M. J. Li, D. Nolan, B. Sanguinetti, R. Thew, and H. Zbinden, Provably secure and practical quantum key distribution over 307 km of optical fibre, Nat. Photonics **9**, 163 (2015).

[14] Y. Liu, T.-Y. Chen, J. Wang, W.-Q. Cai, X. Wan, L.-K. Chen, J.-H. Wang, S.-B. Liu, H. Liang, L. Yang, C.-Z. Peng, K. Chen, Z.-B. Chen, and J.-W. Pan, Decoy-state quantum key distribution with polarized photons over 200 km, Opt. Express **18**, 8587 (2010).

[15] P. Liu and H.-L. Yin, Secure and efficient synchronization scheme for quantum key distribution, OSA Continuum **2**, 2883 (2019).

[16] N. Walenta *et al.*, A fast and versatile quantum key distribution system with hardware key distillation and wavelength multiplexing, New J. Phys. **16**, 013047 (2014).

[17] J. F. Dynes, W. W. Tam, A. Plews, B. Fröhlich, A. W. Sharpe, M. Lucamarini, Z. Yuan, C. Radig, A. Straw, T. Edwards, and A. J. Shields, Ultra-high bandwidth quantum secured data transmission, Sci. Rep. **6**, 1 (2016).

[18] M. Sasaki *et al.*, Field test of quantum key distribution in the Tokyo QKD network, Opt. Express **19**, 10387 (2011).

[19] S. Wang *et al.*, Field and long-term demonstration of a wide area quantum key distribution network, Opt. Express **22**, 21739 (2014).

[20] G. Vallone, D. G. Marangon, M. Canale, I. Savorgnan, D. Bacco, M. Barbieri, S. Calimani, C. Barbieri, N. Laurenti, and P. Villoresi, Adaptive real time selection for quantum key distribution in lossy and turbulent free-space channels, Phys. Rev. A **91**, 042320 (2015).

[21] J.-P. Bourgoin, N. Gigov, B. L. Higgins, Z. Yan, E. Meyer-Scott, A. K. Khandani, N. Lütkenhaus, and T. Jennewein, Experimental quantum key distribution with simulated ground-to-satellite photon losses and processing limitations, Phys. Rev. A **92**, 052339 (2015).

[22] International Telecommunication Union, *G.8262: Timing characteristics of a synchronous Ethernet equipment slave clock*, Technical report, International Telecommunication Union (2010).

[23] H. G. Kim and H. J. Lee, A new burst-mode clock recovery technique for optical passive networks, AEÜ Int. J. Electron. Commun. **64**, 339 (2010).

[24] S. Salem and M. Saneei, All-digital clock and data recovery circuit for USB applications in 65 nm CMOS technology, AEÜ Int. J. Electron. Commun. **103**, 1 (2019).

[25] C. A. Eldering, F. Herrerias-Martin, R. Martin-Gomez, and P. J. Garcia-Arribas, Digital burst mode clock recovery technique for fiber-optic systems, J. Light. Technol. **12**, 271 (1994).

[26] C. Agnesi, M. Avesani, L. Calderaro, A. Stanco, G. Foletto, M. Zahidy, A. Scriminich, F. Vedovato, G. Vallone, and P. Villoresi, Simple quantum key distribution with qubit-based synchronization and a self-compensating polarization encoder, Optica **7**, 284 (2020).

[27] M. Frigo and S. Johnson, The design and implementation of FFTW3, Proc. IEEE **93**, 216 (2005).

[28] L. M. Li, An algorithm for computing exact least-trimmed squares estimate of simple linear regression with constraints, Comput Stat. Data Anal. **48**, 717 (2005).

[29] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk, in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12 (ACM, Istanbul, Turkey, 2012), ISBN 978-1-4503-1159-5, p. 353.

[30] S. Soliman, F. Newagy, and I. Hafez, in *2017 34th National Radio Science Conference (NRSC)* (IEEE, Alexandria, Egypt, 2017), ISBN 978-1-5090-4611-9, p. 371.

[31] B. Zhao, C. Cheng, Z. Ma, and F. Yu, Time delay estimation via co-prime aliased sparse FFT, EICE Trans. Fundam. Electron., Commun. Comput. Sci. **99**, 2566 (2016).

[32] C. Agnesi, M. Avesani, A. Stanco, P. Villoresi, and G. Vallone, All-fiber self-compensating polarization encoder for quantum key distribution, Opt. Lett. **44**, 2398 (2019).

[33] F. Grünenfelder, A. Boaron, D. Rusca, A. Martin, and H. Zbinden, Simple and high-speed polarization-based QKD, Appl. Phys. Lett. **112**, 051108 (2018).

[34] M. Avesani, D. G. Marangon, G. Vallone, and P. Villoresi, Source-device-independent heterodyne-based quantum random number generator at 17 Gbps, Nat. Commun. **9**, 5365 (2018).

[35] R. Qi, Z. Sun, Z. Lin, P. Niu, W. Hao, L. Song, Q. Huang, J. Gao, L. Yin, and G.-L. Long, Implementation and security analysis of practical quantum secure direct communication, Light Sci. Appl. **8**, 1 (2019).

[36] T. Lunghi, J. Kaniewski, F. Bussières, R. Houlmann, M. Tomamichel, A. Kent, N. Gisin, S. Wehner, and H. Zbinden, Experimental Bit Commitment Based on Quantum Communication and Special Relativity, Phys. Rev. Lett. **111**, 180504 (2013).