Supplementary Material for "Autonomous Open-source Hardware Apparatus for Quantum Key Distribution"*

Ignacio H. López Grande, Christian T. Schmiegelow, Miguel A. Larotonda

January 19, 2016

1 Optical Hardware Setup

Light pulses are generated by alternatively injecting a 20 ns, 100 mA current pulse on one of four fast IR LEDs (VISHAY TSHG8400) at every clock tick, depending on the random choice of basis and state: each of the LEDs is used to encode one of the four possible polarization states. The LEDs outputs are coupled to FC-type connectorized multimode fibers and later decoupled using aspheric lens packages (THORLABS CFC-11X-B) to define a propagation direction and divergence, and also to equalize the intensities of the four outputs. In order to increase the coupling efficiency of the edge-emitting LEDs into an optical fiber, the plastic dome and the reflector cup were partially removed, exposing one of the four emitting sides of the semiconductor die. The light emitted from the exposed side was directly coupled to the fiber end (see inset of Fig. 1).

The polarization state preparation and spatial multiplexing of the four states was set over a 40 cm × 40 cm optical breadboard, using an optical arrangement that allows for further miniaturization. The outputs from the four pulsed LEDs were polarized in linear states corresoponding to two conjugated bases and combined into a common path. The LEDs 60 nm emission spectrum bandwidth peaked at 830 nm was narrowed down to a 10 nm FWHM, symmetrical spectrum profile centered at 810 nm using a bandpass interference filter. An approximation to a single photon source is realized by attenuating the pulses in such a way that the mean photon number per pulse is approximately 0.1 at the output of the emitter stage; in this way, assuming Poissonian photon statistics, 90% of the clock pulses are empty pulses, while less than 0.5% of the pulses are multi-photon pulses. Valid protocol runs are clock cycles where one (and only one) photon is detected: both empty and multiple detection runs are considered null.

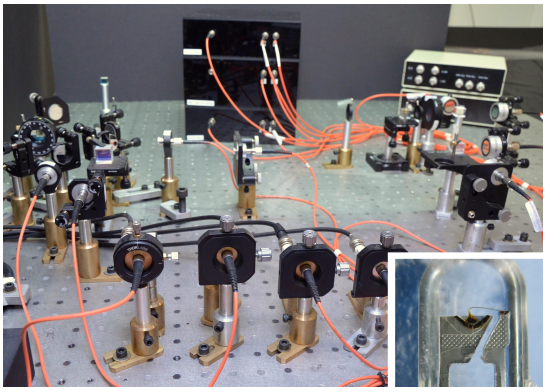

Figure 1: The QKD apparatus: the fiber holders on the front couple the LEDs (hidden by the x-y actuators) to multimode fibers. The optical arrangement on the left is Alice's module, where faint optical pulses of selected polarization are combined in a common-path beam. The detection module is at the right of the picture. On the background, the fiber delays for temporal multiplexing of the detection (center) and the two peripheral controllers for Alice and Bob (far right) can be seen. Inset: a close up of the LED showing the plastic die and part of the parabolic reflector partially removed to maximize the coupling efficiency to the multimode fiber.

At the receiver stage, projected polarization states from the outputs of the polarization beamsplitters are coupled to multimode fibers. For this particular demonstration, Bob's setup was placed one meter away from Alice. The projective measurement is completed with the detection of a single photon on any of the outputs. Temporal multiplexing was achieved by adding fiber patchcords of selected delays: the outputs from each basis were delayed Δt=250 ns from each other using a pair of 50 m fiber spools and then combined using two

*This supplementary material corresponds to the paper I H López Grande, C T Schmiegelow, M A Larotonda, Papers in Physics 8, 080002 (2016), doi: http://dx.doi.org/10.4279/PIP.080002. Corresponding author E-mail: mlarotonda@citedef.gob.ar

multimode fiber couplers. These two paths were further delayed $2\Delta t$=500 ns from each other, and combined afterwards. The four possible outcomes of the experiment are identified by its time of arrival with respect to a clock reference, and photons from the four possible outcomes are detected with a single photon counting module (Perkin Elmer SPCM AQRH-13). Temporal demultiplexing and state determination is obtained measuring coincidences between the single photon detector output and temporal gates with selected delays. Figure 1 shows a photo of the complete setup (see caption for details).

2 Drivers and Control Tasks

2.1 Electronic Drivers

A simplified schematic for the peripheral driver at Alice's side is depicted in Fig. 2. Based on a random 2-bit sequence, the Arduino microcontroller sets a logic high on one of the four possible outputs. A monostable multivibrator uses this logic transition to generate a 20 ns pulse which in turn is used as the input for a high speed LED driver. Each LED drive circuit is constructed using 74ACT00 NAND gates, which switch the LED on and off. Gates are used in parallel to increase the current capacity of the driver, while a clock signal for each triggered channel is generated. The network between the gate output and the LED provide a prebias current and current peaking, [4]. This circuit is copied for each of the four outputs; a common clock signal is generated using XOR gates. An additional variable delay circuit is placed at the input of the driver circuit in order to equalize any temporal difference that may appear at the Arduino digital outputs: in this way any information of the transmitted states that may be transferred to the temporal delays between light pulses can be erased. The choice of standard TTL or CMOS integrated circuits was made to show that the peripheral controllers for this QKD implementation can be built using popular and easily available discrete electronic components. The pulse width is limited by the LED response, although more sophisticated techniques allow for producing light pulses of a few nanoseconds [5].

At Bob's side, single photon pulses are routed through different delay paths according to their polarization, and the delayed photon clicks are identified as polarization state projections by temporal demultiplexing the digital detections. Bob receives a clock pulse and a 16ns TTL pulse from the single photon counter, which is allocated in the corresponding state channel by comparison with a pulse pattern that repeats the temporal delays added by the optical fibers, using fast AND gates. The pattern was generated using a pulse generator, triggered by the clock pulse. Additional monostable circuits equalize the delay and stretch the pulses for an efficient detection at the digital inputs of the Arduino microcontroller (Fig. 3).

Synchronization tasks, as well as communication between Alice and Bob, sieving operations and system diagnosis were implemented on Arduino Mega 2560 programmable microcontrollers. These devices have a clock speed of 16 MHz, several digital input/output pins and a hardware Universal Asynchronous Reception and Transmission (UART) for serial communication, which is channeled over USB for programming and data transfer to and from a personal computer. Additional interfacing of the microcontrollers with other peripherals such as the LED pulse driver, and between the sender and receiver stages were built on expansion printed circuit boards that plug onto the microcontroller boards. Microprocessor programming scripts, as well as the schematics and PCB printed layouts for peripheral circuits are also available as Supplementary Material [1].

2.2 Initial delay calibration

For increased flexibility, the system is able to run a routine that performs a measurement of the temporal delay between pulses before starting the key generation protocol. During this stage, the arrival time of the pulses at the detection stage with respect to the interrupt signal is measured. The system can therefore adapt to different conditions on the distance between the sender and the receiver. The routine consists on successive interrupts sent from Bob to Alice. After sending each interrupt, Bob scans the input channels for 100μs at a sampling rate of 4 MHz. When Alice receives an interrupt signal she sends eight pulses with randomly chosen polarization. The arrival times of the detections are accumulated, and from the resulting histogram the temporal bins of the different polarization states can be measured within the precision of the timing system.

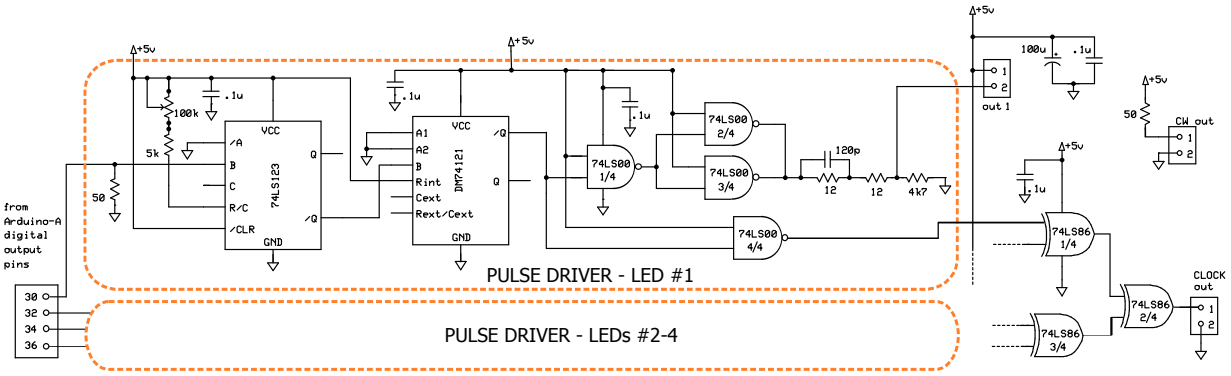

Figure 2: Short pulse led driver, delay compensation and clock signal generation at the emitter side. Logic signals at the input generate a 20 ns current pulse on the specific LED. The delay on the four channels is equalized at the output so that no information of the selected state can be obtained from time-of-arrival measurements. An auxiliary continuous wave output is also provided for alignment of the optical setup. Resistor values in ohms and capacitances in farads

2.3 Classical communication routines

The protocol starts with Bob on a waiting loop. In this condition, the input buffer from the serial port is periodically scanned for a start byte from Alice. Once this start byte is detected, Bob sends a confirmation byte and Alice responds with the protocol parameters: the target key length and the number of data packages sent on each execution of the Q COM routine. Each data package consists of 8 attenuated light pulses with randomly selected polarization. In the Q COM routine, data packages are prepared and sent by Alice. Pairs of random numbers are used to select the basis and the bit value for each pulse. Random numbers can be obtained from a quantum random number generator based on the work of Jennewein *et al.* [2], built for this purpose over a similar Arduino microcontroller with an add-on extension board. At the development stage of the apparatus, the pseudo random number generator available at the basic functions library from the controller software environment was also used.

Once the quantum communication routine has concluded, the system begins with the classical sifting procedure: Alice sends the list of bases used to code the previously sent photons through the serial port. Bob compares this list with the list of detection bases and builds a list of coincidences between emission and detection bases, which is sent back to Alice. They both use this information to locally and independently sift the key. The code has consistency check routines to avoid errors due to data loss. The protocol stops and restarts whenever part of the information is lost at any stage. Also, different counters along the

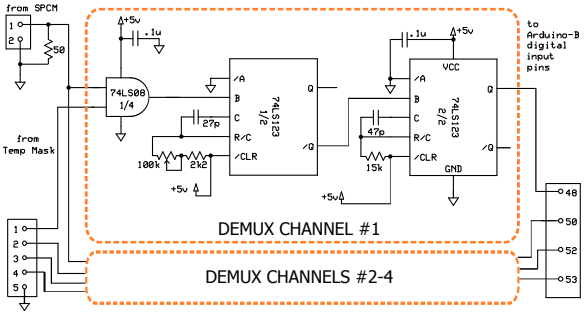

Figure 3: Demultiplexer at Bob's side. A temporal mask, with delayed pulses that correspond with the temporal delay imposed by the fiber stages is compared with the synchronized signal from the detector, to identify the detection. Demultiplexed pulses are reshaped to simplify the detection at the digital input pins of Bob's Arduino.

2.4 Quantum communication routine

This block of the protocol deals with the emission and detection of the faint light pulses, which is the main requirement to generate the raw key. The complete Q COM routine starts with a bidirectional synchronization sequence , which ensures that faint polarized pulses are only emitted when the system is running error-free.

In order to detect the pulses, the digital inputs are scanned several times to detect a sequence of eight events, each valid event having at most one 500 ns pulse, which is generated at the tempo-

ral demultiplexing stage. Using the previously acquired calibration information, Bob registers only the detections that fall within the arrival window of Alice's pulses. If a valid detection takes place, information on basis and state is saved on the raw key list. The code scripts used to program these processes on both microcontrollers is written in Wiring language, [1]. This sequence is repeated for the previously agreed number of times. Once completed, the sifting routine is applied to the lists, and the complete protocol runs until the desired key length is reached.

The temporal delays (256 ns) imposed at the fibers are much longer than the detector temporal jitter (below 1 ns for our specific detector), and the duration of the light pulses (25 ns FWHM, limited by the LED rise and fall times). This helps to minimize the crosstalk between temporal channels at the detection stage. The measured temporal dispersion of the detected signals on the SPCM is of 30 ns FWHM, which is mainly due to the duration of the light pulse. Since the system is based on single photon detections, random jitter dominates over deterministic jitter. The combined jitter of the current pulse driver, the clock signal and the detector is around the nanosecond. Also, the delay between these temporal states is larger than the mean time of the detector afterpulsing probability (typically 200 ns). Such conditions lead to a detection probability of two events on different channels due to detector artifacts below 0.4% [3]. Nevertheless, runs with multiple state detection are discarded. The penalty of such arrangement is a maximum allowed raw pulse rate given by the total delay time of the sequence of pulses: in the present setup this time is conservatively set to $\approx$1 μs, although it can be shortened down to 400 ns (100 ns delay between consecutive pulses).

References

[1] Arduino sketches, pcb schematics and boards also available as Supplementary Material .

[2] T Jennewein, *et al. A fast and compact quantum random number generator.* Rev Sci Instrum, **71**, 1675 (2000).

[3] Perkin Elmer SPCM-AQRH-13, Datasheet (2005).

[4] Agilent Application Bulletin 78. *Low Cost Fiber-Optic Links for Digital Applications up to 155 MBd.*

[5] E Ronchi, *et al. A bipolar led drive technique for high performance, stability and power in the nanosecond time scale.* Nucl Instrum Meth A, **599**, 243 (2009).