Emergent Scientist **1**, 7 (2017)
DOI: 10.1051/emsci/2017009

Emergent Scientist

Available online at:
emergent-scientist.edp-open.org

International Physicists' Tournament 2016

RESEARCH ARTICLE OPEN ACCESS

A simple quantum generator of random numbers

Hugo Roussille[1,*], Lionel Djadaojee[1], and Frédéric Chevy[2]

[1] Département de Physique, ENS-PSL Research University, 24 rue Lhomond, 75005 Paris, France
[2] Laboratoire Kastler Brossel, ENS-PSL Research University, CNRS, UPMC, Collège de France, 24 rue Lhomond, 75005 Paris, France

Received: 10 September 2017 / Accepted: 11 September 2017

Abstract. Cryptography techniques rely on chains of random numbers used to generate safe encryption keys. Since random number generator algorithms are in fact pseudo-random their behavior can be predicted if the generation method is known and as such they cannot be used for perfectly safe communications. In this article, we present a perfectly random generator based on quantum measurement processes. The main advantage of such a generator is that using quantum mechanics, its behavior cannot be predicted in any way. We verify the randomness of our generator and compare it to commonly used pseudo-random generators.

1 Introduction

With the increasing need of secure online communications, cryptography has become one of the cornerstones of computer science. The security of the communications relies on the strength of the algorithm used: for instance the widely used RSA algorithm is only partially secure since its security is based on the difficulty of factorizing large numbers [1]. On the other hand, encryption algorithms based on random keys provide perfectly safe communication protocols [2]. According to Kerckhoffs's principle [3], the strength of this class of cryptographic algorithms rely entirely the randomness of the key. If the behavior of the generator can be in any way predicted, so does the key, and as a result the privacy of the communication is compromised. This problem has been studied thoroughly, for example in the case of Windows operating system [4].

Computer generated random sequences are only pseudo-random. They are based on deterministic algorithms and they use the entropy generated by the user's actions and the computer components to generate randomness. An example of such a pseudo-random number generator (PRNG) is Linux's/dev/random [5]. The main drawback of this kind of generator is that they can easily loop, or create low-entropy numbers. As a consequence, one must be extra careful when using a PRNG [6]. A possible solution would be to use quantum random numbers generators (QRNGs) [7]. Being based on the intrinsic probabilistic nature of quantum measurements, their main advantage is that they can produce a string which is fundamentally unpredictable, even if the device's behavior is fully known. QRNGs have been implemented on different experimental platforms, for example using the quantum fluctuations of vacuum [8], parametric oscillators [9], Raman scattering [10, 11] or photon shot-noise [12].

Here, we present a simple quantum randomizer based onthe statistics of arrival times of single photons on an avalanche photodiode. We show that the randomness of the generator is directly related to its bit-rate and that for a single detector, an almost perfectly random chain of 0 and 1 can be obtained for a bit rate of $\simeq$20 bit/s.

2 Principle of the generator

Every number can be written in base 2, and if this number has been chosen randomly, then every bit of its decomposition in base 2 follows a Bernoulli law of parameter 0.5; the reciprocal is also true. Then, to get a random number, we only have to generate random bits.

The principle of our generator relies on the fact that photon emission is an intrinsically quantum process [13]. As a consequence, their emission time is a random quantity which can be used to generate a string of random bits. Let $(t_n)_{n\in\mathbf{N}}$ be the times at which photons are detected and let us define

$$(d_n)_{n\in\mathbf{N}} = t_{n+1} - t_n, \tag{1}$$

the duration between two subsequent detection events (see Fig. 1). If the photon pairs corresponding to the time intervals d_m and d_n are independent, we expect to have $\mathbf{P}(d_n > d_m) = \mathbf{P}(d_m > d_n)$. But we have $\mathbf{P}(d_n > d_m) + \mathbf{P}(d_m > d_n) + \mathbf{P}(d_n = d_m) = 1$ and $\mathbf{P}(d_n = d_m) = 0$ for we consider continuous probabilities. Thus, if two durations

* e-mail: hugo.roussille@ens.fr

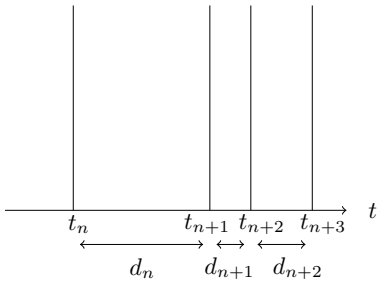

Fig. 1. Scheme of the durations received (each vertical bar represents a photon). t_n is the arrival time of the nth photon and $d_n = t_{n+1} - t_n$. We define a random bit from the sign of $d_n - d_m$ for sufficiently different values of n and m to avoid correlations between arrival times.

are independent, there is a 1/2 probability that the first is greater than the second and 1/2 probability that the second is greater than the first:

$$\mathbf{P}(d_n > d_m) = \mathbf{P}(d_m > d_n) = 0.5. \quad (2)$$

From this 1/2 probability, we can extract easily a bit by defining $b_n = H(d_n - d_m)$ where H is Heaviside's function. We thus have a sequence of bits b_n randomly chosen. However, if n and m are too close, the random variables d_n and d_m are not independent. These correlations have various origins, from the dead time of the detector that introduces anti-correlation at short-time, to quantum-statistics effect arising fromthe bosonic nature of photons [13].

3 Experimental implementation

The experimental setup is presented in Figure 2. The laser source is a 50 mW laser diode shining a Thorlabs SPCM50A avalanche photodiode (APD). A cardboard tube is placed between them to reduce the dark photon count to 3 photon/s. Data acquisition was performed using a Python program based on the NI Visa Python library.

The achievable photon flux is limited by the bin time duringwhich photon counting is performed and the dead-time separating two bins. To avoid multiple-photon events, we reduced the bin time to 1 μs, leading to less than 1% of multiple events. The dead-time of the detector τ is 19 μs. For a flux comparable or larger than $1/\tau$, a significantnumber of photons will not be detected, leading to strong anti-correlations in the statistics of arrival times. To avoid this bias, we reduced the photon flux to 1700 counts/s by inserting two neutral density filters decreasing the beam intensity by a factor 10^4 and 10^3, respectively.Then we made sure the beam was centered on the detector, and used the fact that the beam was much largerthan the detector. The detector was 50 μm wide, and our beam was 30 times wider (radius of 1.5 mm) yielding an intensity reduction by an additional factor 900.

4 Experimental results and randomness tests

To obtain a sufficient number of values, we ran our program several times and then concatenated the obtained arrays. Using this sample, we could generate

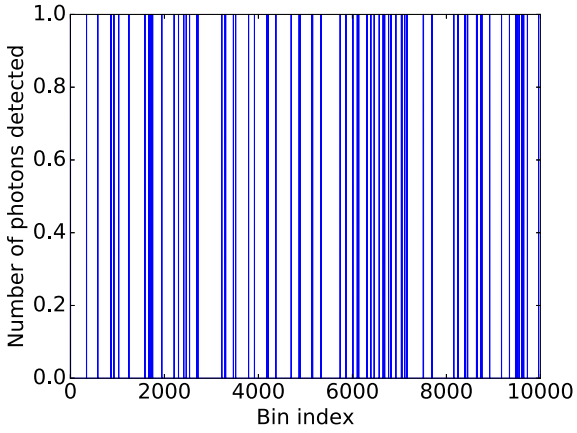

Fig. 2. *Upper panel*: Picture of the setup. The laser source is a 50 mW laser diode attenuated by two neutral density filters. Photons are detected using an avalanche photo-diode (APD) operated in single-photon counting mode. *Lower panel*: Example of detection signal. Two separated bins are separated by a 19 μs dead-time.

a chain of 2 933 920 bits on which we performed a series of statistical tests to assess the randomness of the sequence.

We first display in Figure 3 the histogram of time intervals d_n used to generate the random sequence. It can be fitted by an exponential law, showing that the probability of detection per unit time is constant. This corresponds to a *Poissonnian distribution* of the times between two emissions. We then estimate the average and the variance of the b_n and compare them to the ideal Bernoulli distribution. The experimental average and the variance are respectively 0.49995 and 0.25000. These values are extremely close to those we would theoretically obtain with a probability $\frac{1}{2}$. To make sure the theoretical value was coherent with our results, we computed the trust intervals at 99.7% and 68% using Student's coefficients [14].

$$I_{99.7} = [0.49907, 0.50083],$$

$$I_{68} = [0.49966, 0.50024]. \quad (3)$$

Thus, the generator is a compatible with a Bernoulli random variable of parameter 0.5.

To further test the quality of the generator, we estimate first the autocorrelation function of the data defined by

$$g(r) = \langle d_i d_{i+r} \rangle - \langle d_i \rangle^2. \quad (4)$$

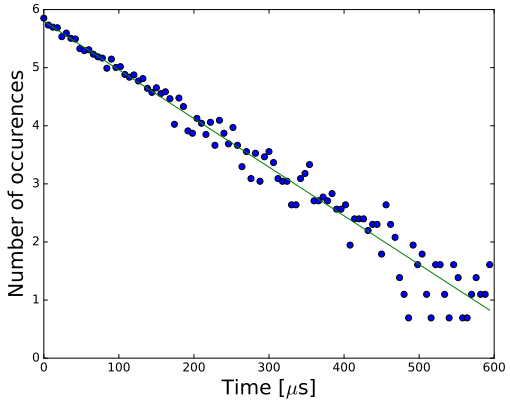

Fig. 3. Histogram of the duration between subsequent arrival times. Dots: experimental data; solid line: time between two emissions for the laser used in our setup.

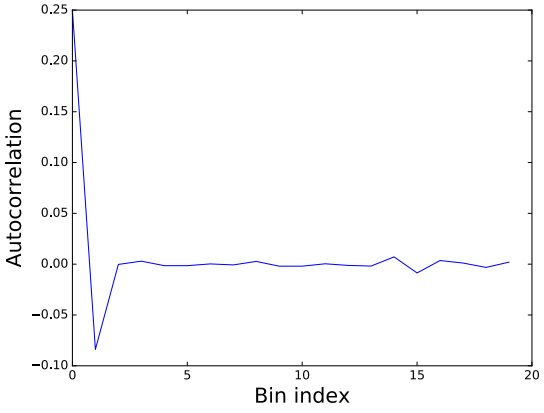

Fig. 4. Autocorrelation function of the random sequence.

In Figure 4 we plotted the autocorrelation of the first bits. We observe that correlations vanish after the first bins.

To refine the characterization of the random sequence, we run two common statistical tests. Firstly, we use the Monte-Carlo method to estimate the numerical value of π, by randomly sampling points in a unit square. The fraction of points at a distance to the origin lower than 1 is then an approximation of $\pi/4$ and any bias in the random sequence translates into a systematic error on the determination of the computed value of π. The result is shown in Figure 5 where we generated the random sequence by comparing d_n and d_{n+i}. We can see that the result of the method is acceptably comparable to Python's random as soon as we take $i = 15$, which is a sharper condition than the one obtained solely from Figure 4. To be precise, in this case the error produced by our random number generator is around 0.3% and it is comparable to that obtained using Python's built-in generator.

As a final test, we implemented the birthday spacing test [15], which is a commonly-used process used to test random numbers generators: if random points are chosen on an interval; the spacing between the points should be exponentially distributed (this is the source of the so-called birthday paradox demonstrating our poor grasp of random phenomena). The test generates "birthdays" based on the

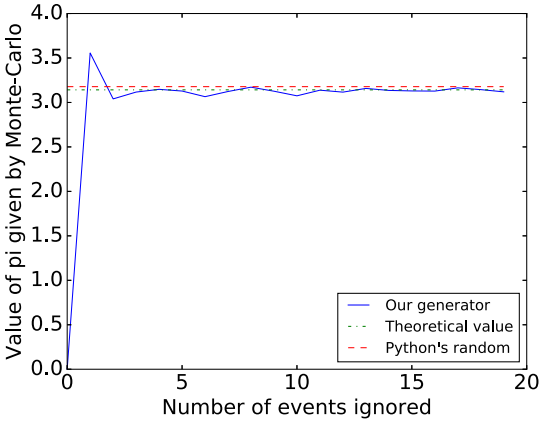

Fig. 5. Results of the Monte-Carlo method for the calculation of π using the durations d_n and d_{n+k}, where k is the number of ignored events, for the determination of each bit.

output of our generator and compares them to the theoretical distribution by computing the chi-square and the associated p-value. If the p-value is greater than 0.05, the test is generally considered has being successful. Just like with the Monte-Carlo test, our generator passed the test when we took only one duration out of 15.

5 Conclusion and outlook

In this article, we have presented a quantum random-number generator based on the arrival time of individual photons on an avalanche photodiode. We have shown that the optimal protocol is the result of a trade-off between the bit-rate and the quality of the random sequence. Based on the outcome of the Monte-Carlo test, the maximum bit-rate allowed by our setup is about 20 bits·s^{-1}, for 1700 photons per second photon count. The main limiting factor is probably the dead-time of the APD but further study is required to confirm this assumption. In the spirit of [12], one can increase the bit-rate using the same protocol by using an array of single-photon detectors, using for instance an EMCCD (electron-multiplying CCD) camera.

In the course of the project, we also considered the possibility of using a pair of APD and place them at the two output ports of a 50/50 beam-splitter. We would have then defined the value of each bit from the output channel of each detected photon. This solution is actually used commercially by Swiss company ID Quantique but the cost of the two APD prevented us from implementing it experimentally.

The work presented in this article was originally realized for the 2016 edition of the International Physicists' Tournament. The authors thank the whole ENS IPT team and leaders for their help through the preparation of the tournament.

References

1. R.L. Rivest, L.M. Adleman, A. Shamir, Cryptographic communications system and method, US Patent 4 405 829, 1983

2. *Information technology – security techniques – encryption algorithms – part 3: block ciphers. Standard* (International Organization for Standardization, Geneva, CH, 2010)
3. A. Kerckhoffs, La cryptographie militaire, J. Sci. Mil. **9**, 38 (1883)
4. Z. Gutterman, L. Dorrendorf, B. Pinkas, Cryptanalysis of the random number generator of the windows operating system, ACM Trans. Inf. Syst. Secur. **13**, 1 (2009)
5. L. Torvalds, Linux kernel random.c (2005), https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/drivers/char/random.c
6. P. L'Ecuyer, Uniform random number generators (Addison-Wesley, Reading, MA, 2010)
7. P.C.M. Owens, J.G. Rarity, P.R. Tapster, Quantum random-number generation and key sharing, J. Mod. Opt. **41**, 2435 (1994)
8. T. Symul, S.M. Assad, P.K. Lam, Real time demonstration of high bitrate quantum random number generation with coherent laser light, Appl. Phys. Lett. **98**, 231103 (2011)
9. K.L. Vodopyanov, A. Marandi, N.C. Leindecker, R.L. Byer, All-optical quantum random bit generation from intrinsically binary phase of parametric oscillators, Opt. Express **20**, 19322 (2012)
10. R. Lausten, G. Wu, I.A. Walmsley, P.J. Bustard, D. Moffatt, B.J. Sussman, Quantum random bit generation using stimulated raman scattering, Opt. Express **19**, 25173 (2011)
11. D.J. Moffatt, J. Nunn, R. Lausten, D.G. England, P.J. Bustard, B.J. Sussman, Efficient raman generation in a waveguide: a route to ultrafast quantum random number generation, Appl. Phys. Lett. **104**, 051117 (2014)
12. B. Sanguinetti, A. Martin, H. Zbinden, N. Gisin, Quantum random number generation on a mobile phone, Phys. Rev. X **4**, 031056 (2014)
13. G. Grynberg, A. Aspect, C. Fabre, Introduction to quantum optics: from the semi-classical approach to quantized light (Cambridge University Press, 2010)
14. P. Lewicki, T. Hill, Statistics: methods and applications: a comprehensive reference for science, industry, and data mining (StatSoft, 2006)
15. G. Marsaglia, Keynote address: a current view of random number generators, in Proceedings, Computer Science and Statistics: 16th Symposium on the Interface (Elsevier, 1985)

Cite this article as: Hugo Roussille, Lionel Djadaojee, Frédéric Chevy, A simple quantum generator of random numbers, Emergent Scientist **1**, 7 (2017)