Fast quantum-optical random-number generators

Thomas Durt,[1,*] Carlos Belmonte,[2] Louis-Philippe Lamoureux,[3] Krassimir Panajotov,[2,4]
Frederik Van den Berghe,[2] and Hugo Thienpont[2]

[1]*Institut Fresnel, Domaine Universitaire de Saint-Jérôme, Avenue Escadrille Normandie-Niémen 13397, Marseille Cedex 20, France*
[2]*B-Phot., Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium*
[3]*QuIC, Ecole Polytechnique de Bruxelles, Universit Libre de Bruxelles, 1050 Brussels, Belgium*
[4]*Institute of Solid State Physics, 72 Tzarigradsko Chaussee Boulevard, 1784 Sofia, Bulgaria*

(Received 8 November 2012; published 25 February 2013)

In this paper we study experimentally the properties of three types of quantum -optical random-number generators and characterize them using the available National Institute for Standards and Technology statistical tests, as well as four alternate tests. The generators are characterized by a trade-off between, on one hand, the rate of generation of random bits and, on the other hand, the degree of randomness of the series which they deliver. We describe various techniques aimed at maximizing this rate without diminishing the quality (degree of randomness) of the series generated by it.

DOI: 10.1103/PhysRevA.87.022339 PACS number(s): 03.67.Dd, 42.50.Lc, 05.40.−a

I. INTRODUCTION

Random numbers play an important role in many applications. For example, Monte Carlo methods rely on random inputs and are important methods in several branches of science. In cryptography, randomness is very often used for the generation of a secret key. On top of that, random numbers have important applications in a variety of different sectors, for example, banks, financial institutions, casinos or gambling halls, and video games. Very often, random series are generated by pseudorandom generators but as their name indicates, such generators are likely, in principle, to be "broken" because the knowledge of a finite number of bits of a pseudorandom series suffices to predict the rest of the series. This is the case, for instance, with many random generators that are used in gambling, which are characterized by a long but finite periodicity. The weaknesses exhibited by pseudorandom generators regarding security led to the development of quantum random-number generators (QRNGs). Indeed, Heisenberg uncertainties show that, in a sense, unpredictability is an intrinsic property of quantum systems: whatever the state of a quantum system is, there always exist observables characterized by a nonzero unpredictability. The price to pay, in comparison with pseudo–random-number generators (RNGs), is that the series generated with quantum generators are always characterized to some extent by undesirable but unavoidable correlations. Indeed, each measuring device is characterized at some level by an amplification process aimed at "bringing quantum fluctuations at the macroscopic level." Now, macroscopic quantities always exhibit some kind of inertia, characterized by an intrinsic memory time or correlation time that limits the production rate of random numbers. Therefore fast quantum random generators are most often accompanied by a hashing procedure that erases the undesired correlations that are present in the "brutto" random series which they generate.

The aim of our paper is to study different types of quantum-optical random-number generators (QORNGs). The first of them was developed by Lamoureux and co-workers at the Université Libre de Bruxelles and is based on statistical fluctuations of the intensity of a laser field; i.e., it requires an optical detector that works in the continuous regime. The other two require optical detectors in the "discrete" regimes, i.e., single-photon detectors). The second one was already conceived in 1994 [1], and it is based on the idea that a photon that enters a symmetric (50:50) beam-splitter has probability one half to be transmitted and one half to get reflected; hereafter we call it the "split" method. It is thus a direct realization of an unbiased "tossing coin" (or Bernoulli) process. We based our study on data collected from an implementation that was realized at the Vrije Universiteit Brussel (VUB), using as the photon source a strongly attenuated laser source, and particular care was brought to the study of the correlations induced by the dead time of the single-photon detectors that were used to collect the data. A third, new type of QORNG was also conceived and realized at the VUB, in which the random nature of the detection time of a photon emitted by a strongly attenuated laser source was exploited in order to generate random bit series that maximize the bit production rate—being given the finite dead time of a single-photon detector.

The paper is structured as follows. In Sec. II we describe the aforementioned generators from a physical point of view. In Sec. III we present some standard tests of randomness that are commonly used in cryptography [the National Institute for Standards and Technology (NIST) tests] and some alternate tests that allow us to gain a more "qualitative" picture of the correlations that are possibly present in the random bit series. In Sec. IV we apply this battery of tests to the characterization of random series generated with the ultrafast QORNG described in Sec. II, and we show how they allow us to develop well-chosen strategies aimed at restoring the randomness of the raw series delivered by the generators without diminishing their speed too much. In Sec. V, we treat the "discrete" generators described in Sec. II in a similar fashion. In Sec. VI, we conclude.

II. PHYSICAL DESCRIPTION OF THREE QORNG's

The first generator studied by us is a high-bit-rate QRNG based on continuous variables which was developed at the

*thomas.durt@centrale-marseille.fr

QuIC of the Université Libre de Bruxelles. Essentially it generates random numbers by measuring fluctuations of a laser intensity, which makes it possible to generate random sequences at bit rates of the order of 1 GHz and beyond (see also Ref. [2] for alternative QORNGs based on laser fluctuations and Refs. [3,4] for a fast classical RNG based on classical fluctuations of a laser in a chaotic regime). The intensity of the laser intensity is measured by a photodiode and converted into a photocurrent $I(t)$, which fluctuates because of underlying quantum fluctuations (high uncertainty in photon number). $I(t)$ is proportional to the number of photoelectrons, and the probability of a given number of photoelectrons in an interval of time follows the Poisson distribution:

$$P(n=k)=\frac{\lambda^k}{k!}e^{-\lambda}. \tag{1}$$

Since $I(t)$ is proportional to the number of photoelectrons, it follows that the photocurrent variance is proportional to the variance of the photon-number fluctuations. As is well known, the variance of the photon-number fluctuations, denoted $(\Delta n)^2$, obeys

$$(\Delta n)^2=\langle n^2\rangle-\langle n\rangle^2=\lambda^2+\lambda-\lambda^2=\langle n\rangle. \tag{2}$$

Because the photocurrent fluctuations follow the photon-number fluctuations, the variance of the photocurrent is (in analogy to the variance of the photon-number fluctuations) proportional to the average photocurrent, i.e., $(\Delta I)^2=c\langle I\rangle$, with c a constant [5]. This was also experimentally confirmed as shown in Fig. 1. It is worth noting that, as can be seen in Fig. 1, there is background noise for $I<10$ mA. This is approximately the lasing threshold, so we may conclude that many sources of unwarranted "classical" noise are present. The experimental curve is significantly above the theoretical shot-noise curve even in the 30-mA range. This excess noise could be an explanation for why the SEQUR QRNG (a prototype version that was used for our experiments) is not an ideal RNG. The classical sources of fluctuations, such as electromagnetic pollution (radio, mobile phones, etc.),

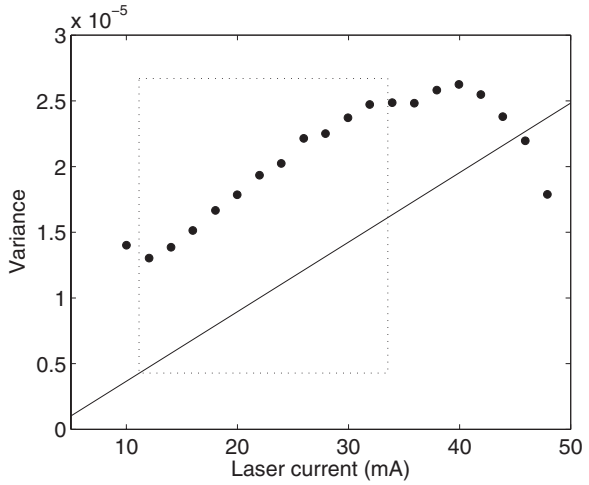

FIG. 1. The variance of the measured laser intensity (circles) is directly proportional to the average value of the laser current according to Eq. (2) (straight line).

relative-intensity noise (RIN), and thermal noise also affect the laser intensity and should be minimized. The electromagnetic pollution can, in principle, be filtered out by Faraday insulation of the generator. In particular, the RIN is the dominating source of classical noise. RIN fluctuations occur at the semiconductor laser relaxation frequency f_{RO}, which is proportional to the square root of the injection current above the threshold: $f_{\mathrm{RO}}=C\sqrt{J-J_{\mathrm{th}}}$ [6]. The center frequency of the RIN is proportional to the laser injection current and is displaced to higher frequencies as the injection current increases. As the current increases, the RIN quickly dissipates because the photodetector's limited bandwidth (2 GHz) no longer measures it. Indeed, $C=2.22\,\mathrm{GHz}/\sqrt{\mathrm{mA}}$ and $J=5.84J_{\mathrm{th}}$, so that $f_{\mathrm{RO}}=14.3\ \mathrm{GHz}\gg f_{3\,\mathrm{dB}}^{\mathrm{PD}}=1\ \mathrm{GHz}$. The remaining noise that is left is shot noise limited. This ensures that most of the contributions to the laser intensity fluctuations come from the quantum regime, i.e., fluctuations of the vacuum. On the other hand, when $I>40$ mA there is a decrease in noise simply because the photodetector becomes saturated. Therefore in order to avoid classical noise sources near the threshold or reduced quantum noise because of photodetector saturation, it is best to operate the laser at around 30 mA.

As the detector measures (displaced) vacuum fluctuations, the obtained random values nevertheless require some additional treatment. In principle, a simple comparator is sufficient to produce random bits. Now, the resolution of the acquisition card (Agilent Acqiris) allowed us to perform an 8-bit discretization step which provided us with more insight into the system at hand. To do so, the measured values are shifted to a strictly positive interval and rounded off in such a way that they become positive-Gaussian-distributed integers within the interval of [0, 255]. These 256 values are finally converted to 8-bit values following the well-known Leibniz's binary decomposition of positive integer numbers, and each one of the bits contributes to a different (random) bit sequence. The first bit reveals, for instance, whether the value is comprised in [0, 127] or [128, 255]; the last (eighth) bit measures the parity of the outcome. All analysis of the SeQuR QRNG that we describe in the following sections was performed on 10 random data (integer) samples of length 10^6. Consequently, this provides 10×8 different bit series of length 10^6 to analyze.

The split QORNG relies on the random choice of a single photon at a beam splitter [1,7]. In this case the randomness is, in principle, guaranteed by the laws of quantum mechanics, thoug, one still has to be very careful not to introduce any experimental artifact that could correlate adjacent bits. Different experimental realizations have been demonstrated [8–10] and one group, ID-Quantique, commercialized the RNG [11]. The Quantis QRNG consists of a single-photon source, a transmission element including a semitransparent mirror where the random process takes place, and two single-photon detectors, each corresponding to one bit state. This system is controlled by triggering electronics for the photon source and acquisition electronics for the single-photon detectors. The processing and interfacing subsystem performs statistical and hardware checks as well as unbiasing of the sequence through a (kept-secret) hashing function. The unbiasing of the physical process in the Quantis QRNG is needed as it is very difficult to guarantee that each detector is set off 50% of the time.

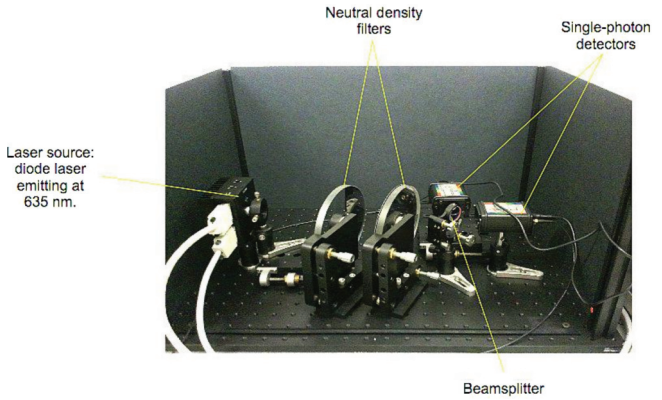

FIG. 2. (Color online) Photo of our setup, which is based on the Quantis QRNG.

According to the developers, the difference between the two probabilities of the detectors being triggered is less than 10%. Quantis produces random bits at a rate of 4 to 16 Mbits/s. More technical details about this quantum random source are given in [11].

We have reproduced the scheme of ID-Quantique [11] without the hashing function in order to have direct access to the raw, brutto, data. Our setup (Fig. 2) is composed of a diode laser emitting at 635 nm, two neutral density filters, a beam splitter, two single-photon detectors, and an acquisition card which is connected to a PC. The single-photon detectors are avalanche photodiodes produced by ID-Quantique. They are characterized by a dead time $T_d \approx 50$ ns.

The high intensity regime is characterized by the appearance of strong deviations from the ideal, Poisson, distribution due to the dead time of the detectors, manifested by peaks separated by a time close to the dead time, as shown in Fig. 3(a). From the interpulse time distribution for large times where it becomes linear [for the right part of Fig. 3(a) the short-time correlations induced by the dead-time mechanism of the detector fade away], we are able to estimate the average time between two photons. Doing so, we find that the average

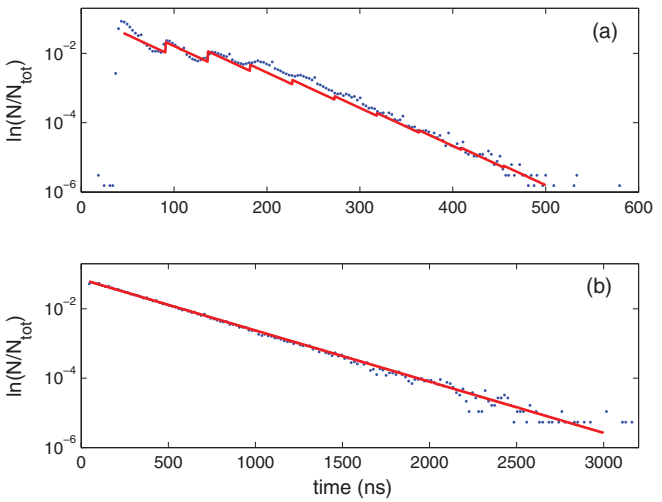

FIG. 3. (Color online) Experimental [filled (blue) circles] and theoretical [solid (red) lines] histogram of time delay between clicks in photon detector 1 at (a) high intensity and (b) low intensity.

time in this case is $\tau = 36$ ns. This is faster than the bit rate of 4 Mbits/s of the Quantis QRNG [11], which gives an average time between photons $\tau = 250$ ns. In order to simulate this regime, we strongly attenuate the laser beam and investigate the corresponding behavior [see Fig. 3(b)]. From the slope in the logarithmic graph we estimate that the average time between clicks is of the order of $\tau = 294$ ns.

Our hypothesis to explain the appearance of the multiple peaks is that at such a high intensity there is a high probability of afterpulsing, i.e., the probability of detecting a photon increases at $t = T_d, 2T_d, 3T_d, \ldots$. A simple model accounting for this hypothesis is to take the detector receptivity as $\eta = 0$ for $0 \leqslant t \leqslant Td$ and $\eta = p(1 + qn)$ for $nT_d \leqslant t \leqslant (n + 1)Td$ for $n = 0, 1, 2, \ldots$ [for Fig. 3(a), $q = 2$]. Combining this with the interpulse time distribution for an ideal Poisson process, i.e., the probability that the next photon is detected after a time t as given by

$$P(t) = \frac{1}{\tau} \exp\left(\frac{-t}{\tau}\right), \tag{3}$$

we obtain the solid (red) line in Fig. 3. For the case of a low input intensity there is no such afterpulsing phenomenon and the experimental curve is well fitted by the exponential distribution with an exponent of $-1/\tau$ [see Fig. 3 (b)].

On the basis of collected data, it is possible to generate a random binary file. In the split method, the clicks in one photon-detector will be the 1's in the final file, the clicks in the other photon-detector will be the 0's, and all events of no clicks or two simultaneous clicks in the two photon detectors will be removed. This is, roughly, the method proposed by Rarity and co-workers [1] and Szovil [7], which has been implemented in [8,10] (it is used in the Quantis QRNG [11], with the adjunction of a kept-secret hashing function).

The parity QORNG is the second method to generate a random-number series based on the parity of the time (in nanoseconds) for which the events (clicks) occur. If this time is even, the bit will be 0; if this time is odd, the bit will be 1. The principal advantages of this method are (i) that it requires the use of only one photon detector to generate a random number and (2) that even in the high-intensity regime it delivers random series of very high quality, as we shall see in Sec. V. The setup to carry out this method is the same setup used previously (see Fig. 2), except that it requires only one photon detector and no beam splitter. Note that the setup with two detectors and a beam splitter could be used too; it would allow us to, roughly, double the generation rate after elimination of the simultaneous clicks.

III. TESTS OF RANDOMNESS

Despite growing interest in RNGs, few official standards exist that address randomness analysis. We hereafter consider the National Institute for Standards and Technology (NIST) battery of tests, consisting of 16 statistical tests [12], which can be found at the Web page of the Computer Security Research Center. The NIST aims to address the independence (determining whether or not there is any redundancy) and coverage (determining how many distinct types of nonrandomness can be investigated and assess whether or not there are a sufficient number of statistical tests to detect deviation

from randomness). Furthermore, it is worth mentioning that the NIST test suite was one of the cryptographic tools which was involved in the evaluation of the candidates for the Advanced Encryption Standard [13]). The NIST tests calculate a p value: $p \in [0,1]$ is the probability of obtaining a test result at least as extreme as the one that was actually observed, assuming that the null hypothesis is true, i.e., the tested sequence is considered random. A p value $\geqslant 0.01$ indicates that the tested series of bits is random, with a confidence interval of 99%.

In Brussels, we have developed additional, more qualitative tests of randomness [14] that we describe now. The *law of large numbers (LLN) test* stems from the LLN stating that the average value of a random sample converges to the expected value as the size of the sample increases. The LLN is important because when it is fulfilled it predicts the expected dispersion from the mean given the length of the series. More concretely, given a sequence of independent random variables $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ with expected values $E(\varepsilon_1) = E(\varepsilon_2) = \ldots E(\varepsilon_n) = \mu$ and variance $\mathrm{Var}(\varepsilon_i) = \sigma^2$. Let us consider the sample average $\bar{\varepsilon}_n = \frac{1}{n}(\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_n)$. According to the lLLN, $\mathrm{Var}(\bar{\varepsilon}_n)$, the variance of the sample average, is equal to $\frac{\sigma^2}{n}$. Therefore, its standard deviation typically behaves as $\frac{\sigma}{\sqrt{n}}$; for a sequences of random bits 0 and 1, $\frac{\sigma_{\mathrm{bit}}}{\sqrt{n}} = \frac{1}{2\sqrt{n}}$. In our implementation of the test, a sequence of n random bits is divided into M different blocks of size N. As we vary the size of the sample N, M will vary accordingly, i.e., M decreases as N increases ($M = \lfloor \frac{n}{N} \rfloor$). The M different blocks are used to calculate the average of the standard deviations of the sample averages.

The *autocorrelation test* checks the autocorrelation C_m found by comparing the original bit sequence with the same sequence shifted by m bits. To this aim, we first convert the bit sequence $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ into a sequence $X = X_1, X_2, \ldots, X_n$ of $+1$'s and -1's via $X_i = 2\varepsilon_i - 1$, so that correlations in the sequence will have a positive contribution and anticorrelations will have a negative contribution. Thereafter the (normalized) autocorrelation C_m is calculated as

$$C_m = \frac{\sum_{i=1}^{(n-m)} X_i . X_{i+m}}{n-m}, \tag{4}$$

with n the total length of the sequence. The amount of shifted bits m is called the degree of correlation and is varied from order 10 to order 10^4. This test enables us not only to investigate possible memory effects but also to pinpoint at which magnitude they occur.

At the beginning of this section, we invoked the LLN. This law is, in principle, not valid when memory effects are present because the random variables are no longer independently distributed. However, one can show that it is possible to consistently make use of the LLN even when a memory effect is present, provided it is a short-range memory effect. For example, it is highly expected that hardware RNGs will pass the LLN test, although they exhibit short-range memory effects. It is for such effects that one needs other tests, for example, the autocorrelation test described above. Nevertheless, such tests are often not well suited for revealing the existence of long-range memory effects. The existence of long-range memory effects can be revealed by the *Hurst parameter test*, named after the hydrologist Hurst, who was the first to apply it when he studied the fluctuation of the level of the river Nile. It is equal to the difference between the maximum and the minimum of the cumulated sum of a random sample divided by a normalization factor equal to the product of the variance of the distribution and the square root of the length of the sample:

$$H_N = \frac{\mathrm{Max}_j\left[\sum_{i=1}^{j}(\varepsilon_i - \bar{\varepsilon}_N)\right] - \mathrm{Min}_j\left[\sum_{i=1}^{j}(\varepsilon_i - \bar{\varepsilon}_N)\right]}{\sigma\sqrt{N}}. \tag{5}$$

It can be shown [15] that the Hurst parameter must be of the order of unity for large values of N in the case of a memoryless, or short-time-memory, stochastic process. In his study of the fluctuations of the level of the river Nile, Hurst discovered that this parameter significantly differed from unity, which means that no Markovian model (or no non-Markovian model with short-range memory effects) could explain the observed fluctuations.

Besides, the floods of the river Nile exhibit another effect called the Joseph effect [15], according to which seven wet years when more land is flooded by the Nile are followed by seven drier years when less land is flooded. Hurst discovered that indeed some trends tend to persist over time. The *persistence test* measures such trends by looking at the sign of the departure from the average value of the data over M successive blocks (subsamples). If this parameter is positive or negative, respectively, persistence or antipersistence occurs. The persistence parameter is calculated as follows. Let us start by converting the bit sequence $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ into a sequence $X = X_1, X_2, \ldots, X_n$ of $+1$'s and -1's. Next, consider

$$\tilde{P}_m = \frac{\sum_{j=m}^{N-m} Z_j}{N - 2m + 1}, \tag{6}$$

with $Z_j = A_j . B_j$, where

$$A_j = \sum_{i=j-m+1}^{j} X_i \quad \text{and} \quad B_j = \sum_{i=j+1}^{j+m} X_i. \tag{7}$$

It is clear that since $X_i = \pm 1$, both $\sum X_i$ represent the binomial sum over a block of length m. Moreover, $\tilde{P}_m$ is Gaussian distributed, with mean $\langle \tilde{P}_m \rangle = 0$ and standard deviation $\sigma(\tilde{P}_m)$. Besides, one can easily see that the mean $\langle Z_j \rangle = \langle A_j \rangle \langle B_j \rangle = 0$, that the squared mean $\langle Z_j^2 \rangle = \langle A_j^2 \rangle \langle B_j^2 \rangle$, and that $\langle A_j^2 \rangle = \langle B_j^2 \rangle$. Consequently, we have that

$$\begin{aligned}
\sigma(\tilde{P}_m) &= \frac{\sigma(Z_j)}{\sqrt{N-2m+1}} = \frac{1}{\sqrt{N-2m+1}}\sqrt{\langle Z_j^2 \rangle - \langle Z_j \rangle^2} \\
&= \frac{1}{\sqrt{N-2m+1}}\sqrt{\langle A_j^2 \rangle \langle B_j^2 \rangle} = \frac{1}{\sqrt{N-2m+1}}\langle A_j^2 \rangle \\
&= \frac{1}{\sqrt{N-2m+1}}\sigma^2(A_j).
\end{aligned}$$

Now, the distribution of A_j can be estimated on the basis of a direct analogy with coin tossing. Let us assume therefore that we toss a coin m times (in other words, we generate a random bit value m times) and associate the value head (H) with $+1$ and the value tail (T) with -1. Formally, $(T + H)^m = \Sigma_{q=0}^{m} T^q \cdot H^{m-q} \cdot \frac{m!}{q!(m-q)!}$. Obviously, the probability

of getting $+1$ q times and -1 $m-q$ times is equal, up to a global normalization factor C, to $\frac{C \cdot m!}{q!(m-q)!}$. It is easy to check that $C = \frac{1}{2^m}$, temporarily assigning the value $+1$ to H and T in the expression above. Accordingly, the probability of getting $A_j = q - (m-q) = 2q - m$ is $\frac{m!}{2^m q!(m-q)!} = \frac{m!}{2^m \frac{(A_j+m)}{2}! \frac{(m-A_j)}{2}!}$, with A_j varying from $-m$ to $+m$ by even increases: $A_j \in \{-m, -m+2, -m+4, \ldots, m-2, m\}$. Now that we know how A_j is distributed, we can compute its variance $\sigma^2(A_j)$. The computation goes as follows:

$$\begin{aligned}\sigma^2(A_j) &= \frac{1}{2^m}\sum_{q=0}^{m}(2q-m)^2 \cdot \frac{m!}{q!(m-q)!} \\ &= \frac{1}{2^m}\sum_{q=0}^{m}(4q^2 - 4qm + m^2) \cdot \frac{m!}{q!(m-q)!} \\ &= m.\end{aligned}$$

If we now normalize $\tilde{P}_m$ by

$$\sigma(\tilde{P}_m) = \frac{\sigma^2(A_j)}{\sqrt{N-2m+1}}, \tag{8}$$

then we are left with a Gaussian random variable with mean 0 and standard deviation 1. Therefore, we define the persistence parameter P_m as follows:

$$P_m = \frac{(\sqrt{N-2m+1})\tilde{P}_m}{m}. \tag{9}$$

In what follows we specify the persistence parameter P_m with respect to the block size as our algorithm calculates it for successively larger block values up to $m = 1000$.

For each of the aforementioned VUB tests we assign a p value; this is illustrated hereafter for the LLN test. First, let us convert the binary sequence $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ to a ± 1 sequence $X = X_1, X_2, \ldots, X_n$. Considering that in the qualitative version of this test, the sequence is divided into M different blocks of length N ($n = M.N$), we introduce $\bar{y}_j$, i.e., the average value over the jth block:

$$\bar{y}_j = \frac{1}{N}\sum_{i=1}^{N} X_{(j.N)-N+i}, \quad \text{with} \quad j = 1 \ldots M. \tag{10}$$

Introducing $u_j = \bar{y}_j^2$ we have that, according to the LLN, for large M,

$$\frac{1}{M}\sum_{j=1}^{M} u_j = \langle u_j \rangle + O\left(\frac{\sigma_{u_j}}{\sqrt{M}}\right). \tag{11}$$

Using here $\sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2$ leads to

$$\begin{aligned}\frac{1}{M}\sum_{j=1}^{M} \bar{y}_j^2 &= \langle \bar{y}_j^2 \rangle + O\left(\frac{\sigma_{u_j}}{\sqrt{M}}\right) \\ &= \langle \bar{y}_j \rangle^2 + \sigma_{\bar{y}_j}^2 + O\left(\frac{\sigma_{u_j}}{\sqrt{M}}\right) \\ &= \langle \bar{y}_j \rangle^2 + \frac{\sigma_X^2}{N} + O\left(\frac{\sigma_{u_j}}{\sqrt{M}}\right),\end{aligned}$$

with $\langle \bar{y}_j \rangle = \langle X_i \rangle = \sum_{i=1}^{n} X_i$ and $\sigma_X^2 = (\frac{1}{n}\sum_{i=1}^{n} X_i^2 - \langle X \rangle^2)$. So we have that

$$\underbrace{\left(\frac{1}{M}\sum_{j=1}^{M} y_j^2\right) - \langle \bar{y}_j \rangle^2 - \frac{\sigma_X^2}{N}}_{\tilde{Z}} = O\left(\frac{\sigma_{u_j}}{\sqrt{M}}\right), \tag{12}$$

i.e., $\tilde{Z}$ is Gaussian distributed around 0, with variance $\frac{\sigma_{u_j}}{\sqrt{M}}$. Now let us consider a normalized version of $\tilde{Z}$ by defining the variable Z as

$$Z = \frac{\sqrt{M}\left(\left(\frac{1}{M}\sum_{j=1}^{M} y_j^2\right) - \langle \bar{y}_j \rangle^2 - \frac{\sigma_X^2}{N}\right)}{\sigma_{u_j}}. \tag{13}$$

Consequently, Z is Gaussian distributed, with mean 0 and variance 1. Moreover, the cumulative distribution function $\Phi(z)$ gives us the probability that the random variable Z is not larger than a given value z and is defined as

$$P(Z \leqslant z) = \Phi(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z} e^{\frac{-t^2}{2}} dt = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)\right), \tag{14}$$

with erf the so-called error function. For positive z Eq. (14) becomes

$$P(|Z| \leqslant z) = 2\phi(z) - 1 = \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right), \tag{15}$$

so that we can obtain the p value as

$$p_{\mathrm{LLN}} = 1 - \mathrm{erf}\left(\frac{|Z|}{\sqrt{2}}\right) = \mathrm{erfc}\left(\frac{|Z|}{\sqrt{2}}\right), \tag{16}$$

with erfc the so-called complementary error function. It is worth noting that in our qualitative approach, we get a complete overview of the behavior of the standard deviation in the sequence, whereas in the quantitative approach we obtain information for a single sequence length N. Let us now consider the expressions for the p value that can be associated with the three other aforementioned qualitative tests.

A p value associated with the autocorrelation test can be derived in a similar way as with the frequency test. Recall Eq. (4), and let us introduce the parameter C_m defined by

$$C_m = \frac{\sum_{i=1}^{(n-m)} X_i . X_{i+m}}{n-m} = \frac{\sum_{i=1}^{(n-m)} Y_i}{n-m}, \tag{17}$$

with m the degree of the correlation and n the length of the tested bit sequence. As one can see from Eq. (17), the sequence Y is again a sequence of ± 1 and, conversely, a binary series. If no correlations are present, the proportions of 0's and 1's in the latter sequence should be approximately the same. By the De Moivre–Laplace theorem, for sufficiently large data sets (in this case, of length $n-m$), the probability distribution of the binomial sum C_m, normalized by its standard deviation $\sigma(C_m)$, is closely approximated by a standard normal cumulative distribution $\Phi(z)$. Since $\langle Y \rangle = 0$ and $\langle Y^2 \rangle = n - m$, we have

that

$$\sigma(C_m) = \sqrt{\frac{1}{(n-m)^2}\sum_{i-1}^{n-m}\sigma^2(Y_i)} \tag{18}$$

$$= \sqrt{\frac{\langle Y^2\rangle - \langle Y\rangle^2}{n-m}} = \frac{1}{\sqrt{n-m}}. \tag{19}$$

Thus

$$P\left(\frac{\sum_{i=1}^{(n-m)} Y_i}{\sqrt{n-m}} \leqslant z\right) = \Phi(z) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)\right). \tag{20}$$

This implies that, for positive z,

$$P\left(\frac{|\sum_{i=1}^{(n-m)} Y_i|}{\sqrt{n-m}} \leqslant z\right) = 2\Phi(z) - 1 \tag{21}$$

$$= \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right), \tag{22}$$

with erf the error function. Filling in the test statistic z with our observed value Z,

$$Z = \frac{\left|\sum_{i=1}^{(n-m)} Y_i\right|}{\sqrt{n-m}}, \tag{23}$$

we obtain the p value

$$p_{\mathrm{corr}} = 1 - \mathrm{erf}\left(\frac{Z}{\sqrt{2}}\right) = \mathrm{erfc}\left(\frac{Z}{\sqrt{2}}\right), \tag{24}$$

with erfc the complementary error function, (16). Again, if the p value is ∼0.01, then one should consider that the sequence is not random.

In order to establish a p value for the Hurst parameter, let us first recall Eq. (5), and let us introduce the following abstraction, known as the R/s statistics:

$$\frac{\mathrm{Max}_j\left[\sum_{i=1}^{j}(x_i - \bar{x}_N)\right] - \mathrm{Min}_j\left[\sum_{i=1}^{j}(x_i - \bar{x}_N)\right]}{\sigma\sqrt{N}} = \frac{R}{s}\frac{1}{\sqrt{N}}, \tag{25}$$

with $s = \sigma$ the standard deviation of the random sample. Although an exact distribution of the R/s statistic is complicated, it was Feller [16] who found the asymptotic distribution of Eq. (25) for the case of independent (not necessarily normally distributed) values of x. Later in 2000, Conniffe and Spencer [17] improved greatly the right-hand-tail accuracy of the distribution. Since the hypothesis test implies the rejection of the test statistic if Eq. (25) exceeds the order of unity, this right-hand-tail approximation fits our purposes:

$$P\left(\frac{1}{\sqrt{N}}\frac{R}{s} > z\right) = 2\left(4\left(z + \frac{1.4}{\sqrt{N}}\right)^2 - 1\right)e^{-2(z+\frac{1.4}{\sqrt{N}})^2}. \tag{26}$$

Inserting the observed value Z

$$Z = \frac{R}{s\sqrt{N}}, \tag{27}$$

which is our test statistic, the p value is calculated as

$$p_{\mathrm{Hurst}} = 2\left(4\left(Z + \frac{1.4}{\sqrt{N}}\right)^2 - 1\right)e^{-2(Z+\frac{1.4}{\sqrt{N}})^2}. \tag{28}$$

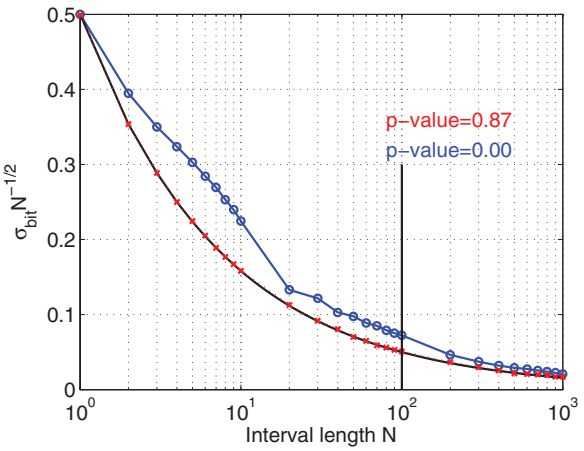

FIG. 4. (Color online) Law of large numbers [vertical (black) line] and SeQuR QRNG: raw data for the bit1 [open (blue) circles and solid (blue) line] and bit3 [(red) X's and dashed (red) line] series. The bit1 series reveals a substantial irregularity of the standard deviation, while the bit3 series behaves as would be expected for a random bit series. This is also confirmed by the example p values calculated for both series.

Establishing a p value for the persistence test is rather straightforward. Recall from Eq. (9) that P_m is Gaussian distributed, with mean 0 and standard deviation 1. Therefore,

$$P(|P_m| \leqslant z) = \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right), \tag{29}$$

with erf the error function [cf. Eq. (16)]. Consequently, the p value is obtained as

$$p_{\mathrm{pers}} = 1 - \mathrm{erf}\left(\frac{|P_m|}{\sqrt{2}}\right) = \mathrm{erfc}\left(\frac{|P_m|}{\sqrt{2}}\right), \tag{30}$$

with erfc the complementary error function.

IV. CHARACTERIZATION AND UNBIASING OF THE SEQUR QORNG

The NIST and VUB batteries of tests were first performed on what we call raw data: those generated by the SEQUR QORNG bit series without any filtering or post-treatment. In general, the eight different bit series obtained from the SeQuR generator failed a lot of randomness tests of the NIST test battery. Nevertheless, despite the poor results for all eight bit series, the two first bit series, i.e., "bit1" and "bit2"(which correspond to the sequences obtained from the two most significant bits of the integer data), performed worse than the six less significant ones. This can be observed from Table I, where we list the average p values for the different NIST tests for the bit1 and bit3 series. Furthermore, this is also revealed by the qualitative tests that we developed complementary to the NIST tests (see, for instance, Figs. 4 and 5 for the performance regarding the LLN and autocorrelation tests). Indeed, looking at Fig. 4, we observe that the standard deviation for the bit1 series is too large [open (blue) circles and solid (blue) line]. A similar result is observed for the bit2 series. The last six bit series obtained from the raw data behave similar to the bit3 series shown by the (red) X's and dashed (red) line in Fig. 4.

TABLE I. SeQuR QRNG, raw data: Results for NIST tests considering the "bit1" and "bit3" series. For each test the average p value and number of sequences that passed the test are listed.

	Bit1		Bit3	
Test name	p value	No. successes	p value	No. successes
Frequency	0.00	0/10	0.00	0/10
Block frequency	0.00	0/10	0.15	7/10
Runs	0.00	0/10	0.10	3/10
Longest run	0.00	0/10	0.45	10/10
Binary matrix rank	0.52	10/10	0.51	9/10
Spectral	0.00	0/10	0.50	10/10
Nonoverlapping template	0.04	1/10	0.48	10/10
Overlapping template	0.00	0/10	0.49	10/10
Universal statistical	0.00	0/10	0.54	10/10
Linear complexity	0.46	10/10	0.55	10/10
Serial	0.00	0/10	0.48	10/10
	0.00	0/10	0.60	10/10
Approximate entropy	0.00	0/10	0.35	10/10
Forward cum. sums	0.00	0/10	0.00	0/10
Backward cum. sums	0.00	0/10	0.00	0/10
Random excursions	0.00	0/10	0.00	0/10
Random exc. variant	0.00	0/10	0.00	0/10

The behavior of the latter sequences is as expected from a random bit sequence.

The deviation from the LLN for the first two (most significant) bit series reveals a memory effect in the system that we attribute to inertia of the implied physical quantities. Considering the autocorrelation of the bit signal with itself, we can also observe this memory effect in the bit sequences constructed from the two most significant bits. In Fig. 5 we plot the autocorrelation for the bit1 series (the bit2 series behaves similarly). The sequences coming from the six least significant bits behave as they should according to the LLN test; as an example, the bit3 series is shown in Fig. 5. Furthermore, we observe a decrease in autocorrelation as the order of the correlation augments. For the bit1 and bit2 series, the largest autocorrelation is present at the level of, say, 10^2 successive bits. Consistently, the eighth significant bit (parity), corresponding to 128 times smaller fluctuations than the first significant bit, no longer exhibits significant autocorrelation. We remark that as the SeQuR QRNG measures a gigahertz signal, correlations of order 100 bits, for example, correspond to the influence of a signal at frequency $\frac{1\ \mathrm{GHz}}{100} = 10$ MHz. Similar conclusions can be drawn from the study of persistence and Hurst tests [14].

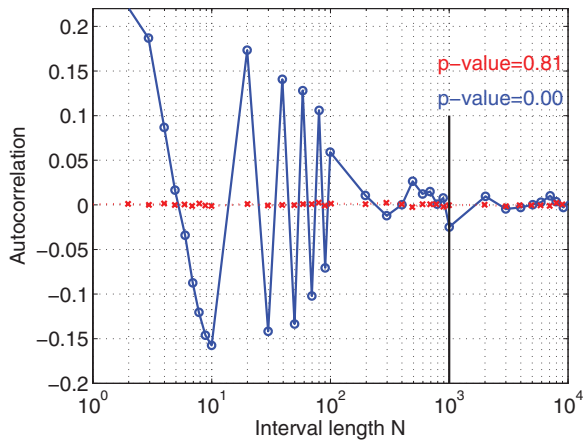

FIG. 5. (Color online) Autocorrelation plot for SeQuR QRNG raw data: the bit1 series [open (blue) circles and solid (blue) line] reveals a strong correspondence within the random sequence, while the bit3 series [(red) X's and dashed (red) line] behaves as would be expected for a random bit series. This is confirmed by the p values.

Without any doubt, the raw data from the SeQuR QRNG are biased, with some memory effect. Because the sampling is performed at a high frequency (GHz), and the autocorrelation test indicates that the largest memory effect occurs from order 10 to order 100 bits, filtering out these low-frequency components should effectively improve the quality of the bit series. As a lot of low-frequency memory is due to electromagnetic pollution such as radio waves and mobile phone radiation, we repeated our analysis of the SeQuR QRNG data after shielding the generator within a Faraday cage. The following results were obtained: all eight bit sequences created from the Gaussian data of the SeQuR QRNG exhibit, according to the NIST tests, similar success rates compared to their unfiltered counterparts. Nevertheless, we observed an improvement in the data when we perform our qualitative tests. This improvement is mainly noticeable in the bit1 and bit2 sequences. Looking at Fig. 6, for instance, it is clear that the bit1 sequence exhibits a slight departure from the LLN for small data samples. The same behavior can be witnessed in the sequences of bit2. The six other bit series obtained from the filtered data behave similarly to the bit3 series (see Fig. 6), which is as expected from a random bit sequence. It is worth noting that certain NIST tests do not always work, although certain of our qualitative tests do work. For instance, it can

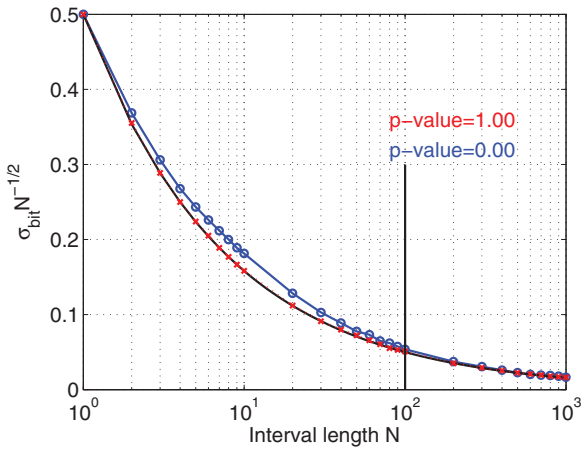

FIG. 6. (Color online) Same as Fig. 4, but for SeQuR QRNG filtered data: the bit1 sequence exhibits a departure from ideal randomness for data samples smaller than 300, while the bit3 series behaves as would be expected for a random bit series. The respective p values confirm this result.

happen that NIST tests based on the frequency test fail in the case that the distribution of 0's and 1's is slightly unbiased, although the LLN test does not fail. This is so because the LLN test aims at confirming that the standard deviation of the average bit value decreases like $\sigma/\sqrt{N}$, independently of the value of σ, while the NIST frequency test measures that σ is very close to 0.5.

The autocorrelation of the bit signal also shows a substantial improvement after filtering the raw signal (see Fig. 7). Although the correlation within the bit sequences constructed from the two most significant bits of the Gaussian samples is still present, it shows a substantial reduction and stays within the order of 10 bits (compare with Fig. 5). The bit series coming from the six least significant bits show no autocorrelation. Similar results can be found after performing the persistence and Hurst tests. The presence of the autocorrelation only in the small order regime, i.e., $O(10)$, gives us much information for

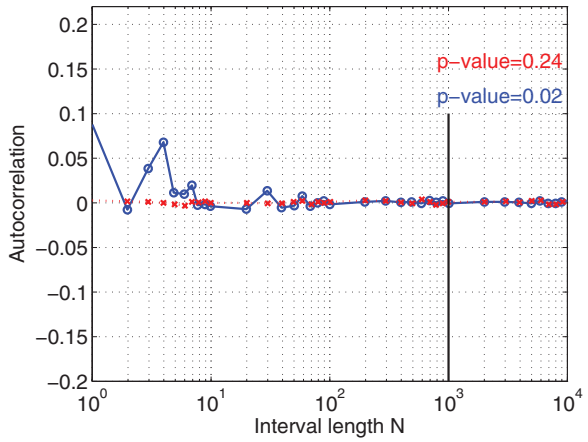

FIG. 7. (Color online) Same as Fig. 5, but for SeQuR QRNG filtered data: the autocorrelation plot for the bit1 sequence shows strong correlations, of order 10, while the bit3 series behaves as would be expected for a random bit series. The p values show that the bit 3 sequence passes the test for an interval length of 10^3.

the improvement of the random data, as well as an indication of possible bias causes. A highly probable cause for this type of correlation within the signal could be a sampling rate that is too fast for the given hardware.

When we look at possible improvements at the software level, several alternatives exist. The most obvious solution is to dilate the sample rate by a factor of the order of the autocorrelation length (10 in this case). Consequently, this would reduce the speed of the generator by the same factor, which is not a satisfactorily strategy. A more clever strategy consists of an XOR operation between bits of the same bit series that are separated by a distance longer than the correlation length (for instance, 200). Such bits are *a priori* decorrelated, and the XOR procedure has the advantage that although it only diminishes the bit generation rate by 2, it efficiently suppresses the correlation between close neighbors in a series. Note that due to the Gaussian shape of the distribution from which the eight-bit series are extracted, correlations exist between the series obtained by realizing an XOR operation between bits from different bit series. We checked that these correlations bias the series obtained by XOR-ing, for instance, bit series 1 and 2, so this strategy should be avoided.

An alternative hashing function is the Von Neumann hashing function (also called the Von Neumann extractor), which is particularly appropriate when the frequency of 1's (0's) in the series is not exactly one half (feedback may help to solve this problem, as reported in [18] regarding the QORNG developed in Vienna). It considers successive pairs of consecutive bits from the input stream, i.e., it decreases the bit generation rate by 2. If the two bits match, no output is generated. If the bits differ, the value of the first bit is output.

Considering the NIST test suite, we observe an improvement in the quality of the series after application of the XOR $(\text{bit1}_i, \text{bit1}_{i+200})$ hashing function and/or of the Von Neumann unbiasing procedure. However, we note that if we apply one of these hashing functions alone, the results of the NIST test suite are still globally negative. Finally, we remark that a combination of the XOR $(\text{bit1}_i, \text{bit1}_{i+200})$ hashing function and of the Von Neumann unbiasing procedure (or vice versa) sufficed to unbias the random series. This is confirmed by the fact that the series obtained after performing the two hashing procedures successively successfully pass all NIST tests and, also, the qualitative tests. We remark that these qualitative tests are of great importance in the design of such improvement operations, as they provide us with valuable information on the behavior of the random series (for instance, the autocorrelation length is necessary for optimizing the XOR hashing function).

V. CHARACTERIZATION AND UNBIASING OF THE "SPLIT" AND "PARITY" QORNG

As ishown in Sec. II, the most relevant problem using the split method is that when the attenuation is low, there is a strong correlation between neighboring bits. We have checked this effect in the "slow" (strongly attenuated) regime for the first time, because in this regime the quality of random series is optimal (this corresponds to the bit generation rate offered by the commercialized generator of ID-Quantique). Even then, some autocorrelation is still present at the level of the brutto bit series (see Fig. 8), and one can show that the majority

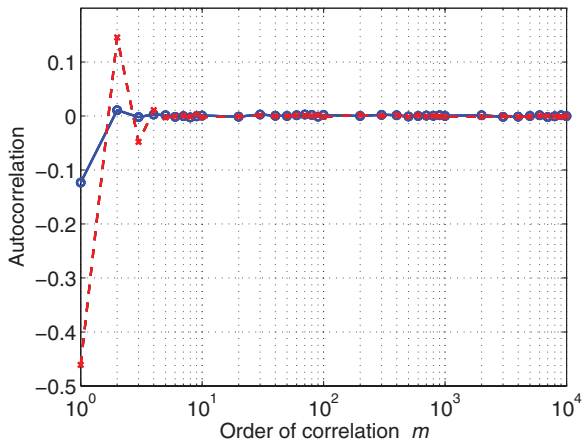

FIG. 8. (Color online) Autocorrelation in the random file generated using the split method, in the "slow" and "fast" regimes, e.g., strongly and less strongly attenuated light, with $\tau = 300$ ns and $\tau = 190$ ns, respectively: solid (blue) line with open (blue) circles and dashed (red) line with (red) X's, respectively.

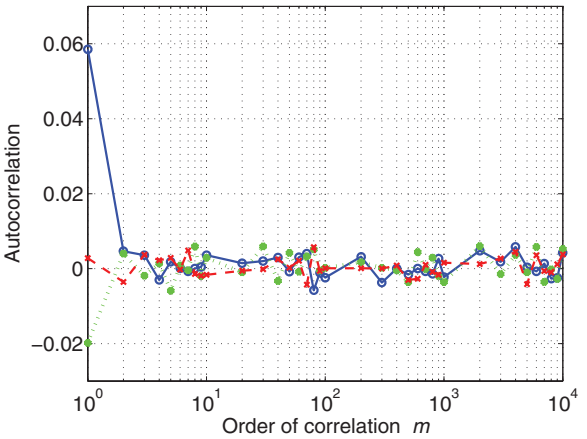

FIG. 9. (Color online) Autocorrelation in the random file generated using the split method, in the "slow"regime and after applying the Von Neumann extractor [solid (blue) line with open (blue) circles], the Von Neumann extractor and then XOR [dashed (red) line with (red) X's], and XOR and then the Von Neumann extractor [dotted (green) line with filled (green) circles].

of NIST tests fail. It is easy to check that this correlation is due to the dead time: its magnitude can be estimated as the dead time of the photon detector divided by the average time between two photons. For the strongly attenuated regime this gives 45 ns/300 ns = 0.15—in good agreement with the value at 10^0 (the correlation between successive clicks) for the solid (blue) line with (blue) circles in Fig. 8. The dashed (red) line with (red) X's corresponds to an average time between two clicks of the order of 190 ns, and indeed the autocorrelation between first neighbors is then at a ratio 3/2 of the one obtained for 300 ns. The negative sign is due to the dead time: after a click occurs in one detector, it is more likely to be followed by a click in the other detector, which constitutes an anticorrelation ($+1$ followed by -1, and vice versa).

We applied the Von Neumann extractor [19] to the bit series of the split generator in order to improve the quality of the random series, according to the strategy outlined in the previous section. The NIST tests revealed an improvement of the results, but the series still failed most of the tests. The autocorrelation test [see the solid (blue) line with open (blue) circles in Fig. 9] also shows that we still have a strong correlation in neighboring bits. After applying the Von Neumann extractor, the autocorrelation becomes positive because the probability of obtaining 11 or 00 in the hashed series (that is, 1010 or 0101 in the raw series) is greater than the probability of obtaining 10 or 01 (that is, 1001 or 0110 in the raw series), always due to the dead time of the avalanche photodiode detectors.

According to the strategy outlined in the previous section, we considered another improvement, which consists of an XOR operation between two bits that are not correlated; we take bits separated by a distance longer than the autocorrelation length. After applying the Von Neumann extractor and then the XOR (but not the other way around) and checking the randomness of the resulting series, the file passes all the NIST tests. This is illustrated by the dashed (red) line with (red) X's in Fig. 9.

Besides, after applying the improvements to the binary file generated by the split method (Von Neumann and XOR or the other way around), in the high-intensity regime, the generated series still failed certain NIST and qualitative tests. We see here that the advantage regarding the speed that could be obtained by operating at a rate close to the inverse of the dead time of the detectors does not compensate the loss of randomness due to the too fast sampling rate.

The second method for generating a random-number file is based on the parity of the time (in nanoseconds) for which the events (clicks) occur. If this time is even, the bit will be 0; on the other hand, if this time is odd, the bit will be 1. The principal advantage of this method is that it suffices to use only one photon detector in order to generate a random bit series. Besides, we checked that the generated bit sequence passed all the NIST and qualitative tests. If we check, for instance, the autocorrelation of the series generated through the parity method that is shown in Fig. 10, we find that there is no correlation at all with this method, even with low attenuation, although applying the split method, we observed a high correlation in the same regime.

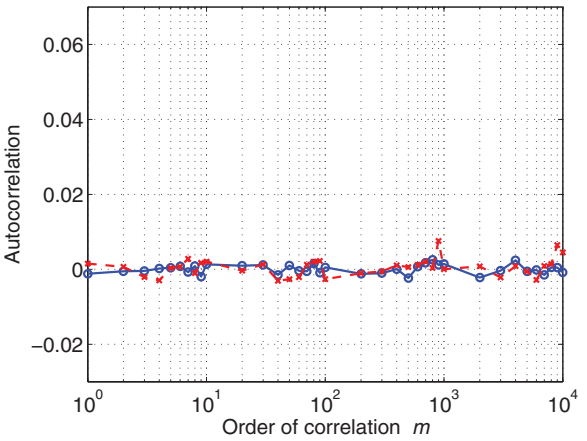

FIG. 10. (Color online) Autocorrelation in the random file generated using the parity method, in the "slow" and "fast" regimes: dashed (red) curve and solid (blue) curves, respectively.

VI. CONCLUSIONS

From our analysis of the SeQuR and "split" QORNG, we conclude that the combination of two hashing functions—(i) a Von Neumann extractor, followed by (ii) an XOR between bits separated by a distance larger than the correlation distance revealed by the autocorrelation test—provides a strategy that is not too expensive regarding the decrease in the bit generation rate (the speed of the generator) and suffices to restore the loss of randomness caused by the internal correlations of the bit series. This strategy could be systematized and applied to other types of RNGs in the future. Presently we are developing user-friendly software that was inspired by our study. It incorporates a "diagnose" program that, after having been fed by a random series, automatically delivers the results of the quantitative and qualitative tests described in our paper, but also opens the way to an online "randomness restoring" process aimed at improving the quality of biased series, in other words, an online "optimal unbiaser."

We also discuss a method, called the parity method, that opens the way to efficient competitive QRNGs. To realize this, it suffices, in principle, to have at one's disposal (i) a single-photon detector (of dead time D; for instance, D is of the order of 50 ns in our implementation); (ii) a clock (of resolution sufficiently smaller than the dead time of the single-photon detector, say, 1 ns in our case); (iii) an attenuated laser source that produces mostly single photons separated at a rate equal to at least $1/D$; and (iv) a chip that will connect i and ii and load a buffer. Combining these elements, it is possible, in principle, to produce a QORNG that works at a rate of the order of $1/D$, which is better than what can be achieved with the "split" method. It is worth stressing at this level that the bit series generated by the parity method passes successfully all NIST tests as well as our qualitative tests without imposing the use of any kind of hash functions. This approach opens the way to a new type of (optimally fast) discrete QORNG based on single-photon detectors. In our case we can reach a bit generation rate of the order of $1/D \approx 2 \times 10^7$ Hz. In a version with two detectors, we would reach more or less 4×10^7 Hz. On the other hand, the ultrafast SeQuR QORNG that works in the continuous regime makes it possible to reach, after application of the unbiasing procedures, a rate of the order of 1 GHz/4 $= 2.5 \times 10^8$ Hz. Compared to the commercially available QORNG produced by ID-Quantique [11], which operates at a rate of the order of 4×10^6 Hz, these prototypes thus offer promising perspectives. It is worth noting, however, that, although the parity method delivers series of a quality comparable to that of series delivered by the generator of ID-Quantique, without imposing the use of hashing techniques, it imposes the use of a clock which operates at a rate faster than the dead time of the single-photon detectors.

ACKNOWLEDGMENTS

The authors acknowledge financial support from the Impulse Programme of the Brussels Capital Region for the NICT sector via research project CRYPTASC and from IAP-BELSPO Grant No. IAP P7-35. Two of us (T.D. and L.L.) acknowledge enriching discussions with and logistic support from Guy Verschaffelt (TONA VUB) regarding the measurement of the frequency distribution of the noise at the level of the detector involved in the SeQuR QRNG.

[1] J. G. Rarity, P. C. M. Owens, and P. R. Tapster, J. Mod. Opt. **41**, 2435 (1994).
[2] T. Symul, S. M. Assam, and P. K. Lam, Appl. Phys. Lett. **98**, 231103 (2011); F. Xu, B. Qi, X. Ma, H. Xu, H. Zheng, and H.-K. Lo, Opt. Express. **20**(11), 12366 (2012); http://www.comm.utoronto.ca/~hklo/QRNG/Quantoss.html.
[3] A. Uchida, K. Amano, M. Inoue, K. Hirano, S. Naito, H. Someya, I. Oowada, T. Kurashige, M. Shiki, S. Yoshimori, K. Yoshimura, and P. Davis, Nat. Photon. **2**, 728 (2008).
[4] I. Reidler, Y. Aviad, M. Rosenbluh, and I. Kanter, Phys. Rev. Lett. **103**, 024102 (2009).
[5] Mark Fox, *Quantum Optics: An Introduction* (Oxford University Press, New York, 2008).
[6] L. A. Coldren and S. W. Corzine, *Diode Lasers and Photonic Integrated Circuits* (Wiley, New York, 1995).
[7] K. Svozil, Phys. Lett. A **143**, 433 (2011).
[8] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, Rev. Sci. Instr. **71**, 1675 (2000).
[9] E. Hildebrand, Ph.D. thesis, Johann-Wolfgang Goethe-Universitat, 2001.
[10] A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden, J. Mod. Opt. **47**, 595 (2000).
[11] Random number generation using quantum physics; http://www.idquantique.com/images/stories/PDF/quantis-random-generator/quantis-whitepaper.pdf.
[12] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, NIST Spec. Publ. **22**, 800 (2001).
[13] Federal Information Processing Standards Publication 197. Advanced Encryption Standard (AES) (2001), http://csrc.nist.gov/publications/PubsFIPS.html.
[14] F. Vanden Berghe, Ph.D. thesis, Vrije Universiteit Brussels, 2011.
[15] B. Mandelbrot, *The Fractal Geometry of Nature*, 1st ed. (W. H. Freeman, New York, 1982).
[16] W. Feller, Ann. Math. Stat. **22**, 427 (1951).
[17] D. Conniffe and J. E. Spencer, Econ. Soc. Rev. **31**, 237 (2000).
[18] C. Calude, M. Dinneen, M. Dumitrescu, and K. Svozil, http://arxiv.org/pdf/1004.1521.pdf.
[19] Von Neumann extractor; http://en.wikipedia.org/wiki/Randomness_extractor.