Almlöf et al. *EPJ Quantum Technology* (2024) 11:80
https://doi.org/10.1140/epjqt/s40507-024-00288-0

EPJ Quantum Technology
a SpringerOpen Journal

EPJ.org

RESEARCH Open Access

On the randomness of time ordered quantum measurements

Jonas Almlöf[1], Thomas Lettner[2], Samuel Gyger[2], Gemma Vall Llosera[1], Tigge Nilsson[1] and Val Zwiller[2*]

*Correspondence: zwiller@kth.se
[2]Department of Applied Physics, KTH Royal Institute of Technology, Stockholm, Sweden
Full list of author information is available at the end of the article

Abstract

A new method for efficient, high-quality randomness extraction is presented. The method relies on quantum processes such as the emission of single photons and their subsequent detection, where each detection event has an associated detection time. By establishing a list of time differences between a fixed number of events, a unique order can be established.

We note that, by utilising the number of ways to order the resulting list of time differences between the quantum events, the efficiency can be increased many-fold compared to current methods. The method delivers fundamentally uniform randomness and therefore, in principle, does not need debiasing.

Keywords: Randomness; Photon detection

1 Introduction

In many technological areas randomness is an important resource such as in scientific or engineering simulations, in statistical sampling, in probabilistic computation, in quantum and classical cryptography. Optimally, a random sequence should be impossible to predict, i.e., it should be harvested from a nondeterministic process. But often deterministic so called pseudorandom [1, 2] processes are used in combination with some, at least partly unpredictable processes, such as the thermal noise in electronic devices [3] or the amplitude in chaotic oscillators [4]. The shorter nondeterministic string is used as input to produce a longer sequence. The resulting string can, through the application of an iterative one one-way function be engineered to produce a seemingly uniform random distribution. However, pseudorandom processes implemented in computers are fundamentally deterministic, since every specific input produces a specific outcome, and in this work we shall focus on quantum processes, since such processes are believed to be fundamentally random.

A quantum state in the form of an equal superposition of two basis states, e.g., $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ will upon a measurement in the basis $\{|0\rangle, |1\rangle\}$ collapse to either of those two eigenstates, with a probability of 1/2. Such probabilities, the Born probabilities, are fundamentally nondeterministic to the best of our knowledge. For this reason, some of the random number generators on the market are based on this principle, in the form of a which-path binary outcome of a photon as it either passes through or is reflected in a

Springer

balanced beam splitter. The information of which way the photon took decides the random bit value [5]. This process thus generates one random bit per photon.

However, in such practical setups it was noted that if the beam splitter was even slightly off from its balanced setting, the random bit values were somewhat biased towards one of the two outcomes and thus some debiasing had to be performed, i.e., post processing the random bits to restore the balance. Such debiasing, however, involves discarding part of the data, see e.g., von Neumann debiasing [6]. Thus a good randomness extractor would be one that outputs bits that are uniformly distributed, i.e., so called non-biased random values [7]. Other authors too have noted that it is possible to extract a non-biased random bit sequence taking advantage of a particular probability density function stemming from photon emission, which can be modeled as a Poissonian process [8]. Here, the recorded detection times can be transformed into a uniform, or non-biased, distribution. Another interesting feature of the protocol [8, 9] is that for each photon event, considerably more than one random bit can be harvested in this way. Since each equiprobable event corresponds to a particular time bin (while all other bins are empty), the scheme resembles pulse position modulation [10], which also can be used to transmit more than one bit of information per photon [11]. In [12] it was pointed out that the timing precision poses a fundamental limit for how much information per photon can be conveyed, something we shall also find is true in this work (but for random bits).

We will now present a method that can extract uniform, or non-biased randomness, from timing information of quantum events, regardless of the distribution of the emission process. The efficiency (entropy bits per photon) for which this can be done, is significantly more than one and we only require that the timing of the quantum events can be treated as independent and identically distributed (i.i.d.) random variables.

2 Method

Consider a random variable X with a probability density function $f(x)$. Now suppose that n samples are drawn from X, i.e., $x_1, x_2, x_3 \ldots x_n$. We start by showing that each of the $n!$ possible ways to order the samples are equally probable and irrespective of $f(x)$. To see this, consider first the simplest case of $n = 2$ and suppose that we have just drawn the first sample x_1. Then the probability of drawing x_2 such that $x_1 < x_2$ is simply

$$P(x_1 < x_2) = \int_{x_1}^{\infty} f(x_2)dx_2 = [F(x_2)]_{x_1}^{\infty} = \underbrace{F(\infty)}_{=1} - F(x_1). \tag{1}$$

If we consider the probability for $x_1 < x_2$ *before* drawing x_1, we need to multiply each of the outcomes for x_2 with the probability for drawing x_1, such that

$$\begin{aligned} P(x_1 < x_2) &= \int_{-\infty}^{\infty} f(x_1)(1 - F(x_1))dx_1 = \underbrace{\int_{-\infty}^{\infty} f(x_1)}_{=1} - \int_{-\infty}^{\infty} f(x_1)F(x_1)dx_1 \\ &= 1 - \left[\frac{F(x_1)^2}{2}\right]_{-\infty}^{\infty} \\ &= 1 - (\underbrace{\frac{F(\infty)^2}{2}}_{=1/2} - \underbrace{\frac{F(-\infty)^2}{2}}_{=0}) = 1/2. \end{aligned} \tag{2}$$

```
WhichOrdering[list_]:= Module[{sortedlist,result,pos},
  result=0;
  sortedlist=Sort[list];
  Do[
    pos=FirstPosition[sortedlist,list[[ii]]][[1]];
    result=result+(pos-1)*Factorial[Length[list]-ii];
    sortedlist=Delete[sortedlist,pos];
    ,{ii,1,Length[list]}
  ];
  result
]
```

Listing 1 Mathematica implementation of how to convert an ordering into an integer

Note that this result *does not depend on* $f(x)$.

For $n = 3$ we can calculate $P(x_1 < x_2 < x_3)$ as

$$\begin{aligned} P(x_1 < x_2 < x_3) &= \int_{-\infty}^{\infty} f(x_1) \int_{x_1}^{\infty} f(x_2) \int_{x_2}^{\infty} f(x_3) dx_3 dx_2 dx_1 \\ &= \int_{-\infty}^{\infty} f(x_1) \left(\frac{1}{2} - F(x_1) + \frac{F(x_1)^2}{2} \right) dx_1 \\ &= \left[\frac{F(x_1)^3}{6} \right]_{-\infty}^{\infty} \\ &= 1/6. \end{aligned} \tag{3}$$

For a general n, $P(x_1 < x_2 < x_3 \ldots < x_n)$ similarly becomes

$$\begin{aligned} P(x_1 < x_2 < x_3 \ldots < x_n) &= \left[\frac{F(x_1)^n}{n!} \right]_{-\infty}^{\infty} \\ &= \frac{1}{n!}, \end{aligned} \tag{4}$$

irrespective of $f(x)$. We also note that the last expression is invariant under interchanging the indices of the samples, meaning all possible orderings of n samples have the same probability $1/n!$. We have thus shown that all $n!$ possible orderings of n samples drawn from an arbitrary (but identical) distribution are equiprobable. This can be exploited in a random number generator by repeatedly drawing blocks of n measured time differences of quantum events and map them to a unique bit-string.

2.1 Mapping orderings to bit strings

We now consider mapping any possible ordering of a sequence s of length n onto a unique integer between 0 and $n! - 1$. The first step is to order s in ascending order into a list s_o. Then, for element $i = 1$ in s, find what position pos it has in s_o and add $(pos - 1) * (n - i)!$ to the result. Then remove element s_i from s_o and continue this procedure with $i = 2, 3, \ldots n$.

The complete implementation is shown in Listing 1 and example input and output are listed in Table 1.

We note that sometimes it is not possible to find a perfect mapping of all orderings to all *bit sequences* of length k, since

$$n! = 2^k \tag{5}$$

Table 1 We list all ways to order 3 unique samples and assign an integer to each possibility. In this implementation we use Mathematica's *Sort* function which is based on lexicographic ordering, however in principle any 1-to-1 mapping of orderings onto unique integers will do

s	WhichOrdering[s]
(1, 2, 3)	0
(1, 3, 2)	1
(2, 1, 3)	2
(2, 3, 1)	3
(3, 1, 2)	4
(3, 2, 1)	5

has only one nontrivial solution for $n = 2$ and $k = 1$. This combination was exploited in [7] where the 2 time differences formed between 3 adjacent detector clicks were used as to generate one random bit. Thus, if no clicks are re-used, the scheme has an efficiency of 1/3 bits per click and it is the main message of this work to show that we can build a bias-free random number generator with considerably better efficiency.

For other combinations of n and k, not all integers $0, 1, \ldots, n! - 1$ can be uniquely mapped to one element in $0, 1, \ldots, 2^k - 1$, while at the same time exhausting all of the latter elements. From the perspective of generating random bits *efficiently*, it is therefore best to choose an n and a k such that $n!$ is just slightly larger than 2^k, i.e., when $\log_2(n!)$ has only a small fractional part, e.g., $\log_2(65!) = 302.018$ or $\log_2(959!) = 8122.00016$. In the latter example we can achieve a bias-free randomness extraction efficiency of about 8.46 entropy bits per sample, i.e., about 25 times more than when just comparing 2 time differences. The average efficiency expressed as the number of random bits per photon, $\mathcal{E}$, we calculate as

$$\mathcal{E} = \frac{\text{floor}\left(\log_2(n!)\right) 2^{-\text{fractionalpart}\left(\log_2(n!)\right)}}{n+1}. \tag{6}$$

In Fig. 1, $\mathcal{E}$ is plotted to illustrate the general dependency on n, and that some choices of n are better for mapping onto a binary number system. Fig. 1, however, does not include any effect from a finite precision d and collisions that could occur due to this.

Note that the n samples under consideration correspond to the time differences between $n + 1$ quantum events, such as the detection of photons (See Fig. 2 for an illustration). Consequently, since in our scheme we chose not to re-use any detections (clicks), we have $n + 1$ in the denominator of (6).

2.2 Fixing collisions

So far we have disregarded of any of the possibilities $x_i = x_j,\ i, j \in 1 \ldots n$, which will have a nonzero probability to occur in a realistic setting. The probability for such events depends on the precision d of the measurement and the distribution $f(x)$, i.e., for two given samples x_1 and x_2 we have

$$\begin{aligned} P(x_1 - d/2 < x_2 < x_1 + d/2) &= \int_{-\infty}^{\infty} f(x_1) \int_{x_1 - d/2}^{x_1 + d/2} f(x_2) dx_2 dx_1 \\ &= \int_{-\infty}^{\infty} f(x_1) \left(F(x_1 + d/2) - F(x_1 - d/2)\right) dx_1 \\ &\approx \int_{-\infty}^{\infty} d \cdot f(x_1)^2 dx_1. \end{aligned} \tag{7}$$

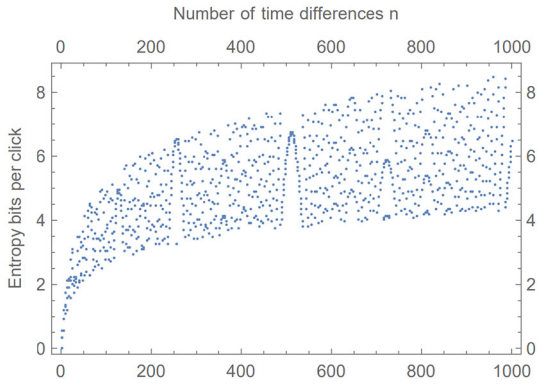

Figure 1 Efficiency $\mathcal{E}$ as a function of number of ordered samples n

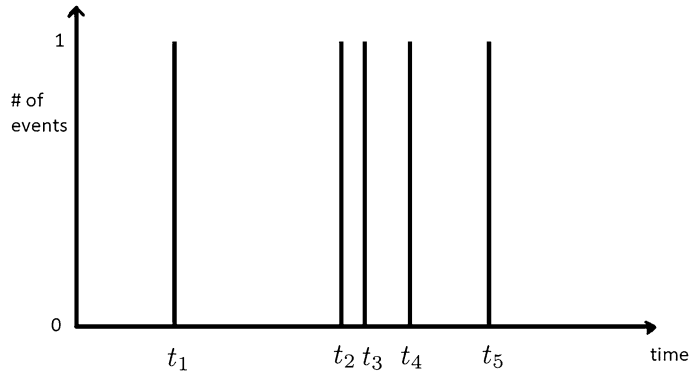

Figure 2 From 5 quantum events, 4 time differences can be formed and subsequently ordered. In the illustrated example we have $t_3 - t_2 < t_4 - t_3 < t_5 - t_4 < t_2 - t_1$, which is just one of 4! = 24 equally possible orderings

We shall "fix" such collisions as they appear, using stored random bits. If two samples are deemed equal we can consume a random bit that was previously extracted: if the bit has value 0, adjust the second value in the identical pair with a small value $+\epsilon < d$ and if the value is 1, adjust the first value with $+\epsilon$. This procedure will reassign a valid order and the randomness extraction method can proceed, at the cost of 1 random bit per collision. In most cases it is advantageous to use one saved bit for each collision rather than discarding the whole block, since, e.g., for $n = 65$, nominally floor($\log_2(65!)$) = 302 bits can be extracted.

In order to fix the situation when 3 time differences have the same value, we note that each of the 6 possible orderings need to have the same probability. Consuming 3 random bits can achieve this by adjusting the identical values by $\{(-\epsilon, 0, +\epsilon), (-\epsilon, +\epsilon, 0), (0, -\epsilon, +\epsilon), (0, +\epsilon, -\epsilon), (+\epsilon, -\epsilon, 0), (+\epsilon, 0, -\epsilon)\}$ for the bit values $000, 001, 010, 011, 100, 101$ respectively, as long as the value is not 110 or 111 (6 or 7, see Table 1). In such cases, a new set of 3 bits

need to be consumed, i.e., the expected number of consumed bits is

$$\begin{aligned}\langle \text{bits used} \rangle_{\text{triple}} =& 3P_s + 6(1-P_s)P_s + 9(1-P_s)^2 P_s \ldots \\ =& 3P_s \sum_{k=1}^{\infty} (1-P_s)^{k-1} \cdot k \\ =& 3/P_s \\ =& 4,\end{aligned} \tag{8}$$

where $P_s = 6/8$, i.e., the usable fraction of the possible bit values of the 3 bits we have drawn.

3 Experiment and results

3.1 Limitations and planning

Our contribution lies in delineating a protocol for randomness generation that does not, in principle, hinge on what method is used for generating and detecting the quantum events. Thus, when we tested the protocol in our lab we did not hesitate to use our existing and rather expensive equipment. A realistic device should take into consideration cost, noise, size, power and other parameters, which we did not consider. More experimentation is needed to see how cheaper, room-temperature devices will adhere to our protocol assumptions, i.e., we assume events whose time-between-detection is independent and identically distributed (i.i.d.).

Another consideration is if a continuous wave (CW) laser of pulsed laser is better for generating the source photons. Although we have used a pulsed laser, it seems plausible that using a CW laser can reduce the collision probabilities, especially when the experiment is operating at a high photon rate.

We used an APE Pico Emerald pulsed laser at 80 MHz, attenuated to approximately 10,000 counts per second. We detected the photons using a superconducting nanowire single-photon detector from Single Quantum (the EOS model). This detector can not distinguish between single and multiple photons in each pulse. The laser we used was attenuated so that each event would fulfil our initial assumption, that all events are independent. In our experiment, this meant that only one of 8000 pulses gives a click in the detector, on average. The time of each detector click were then recorded using a quTag time tagger (from quTools) with picosecond digital resolution, i.e., our precision was $d = 10^{-12}$ s. A list of n samples was formed from the $n+1$ first recorded clicks, each record denoting the time between two adjacent quantum events.

3.2 Results

Three experiments were performed using different list lengths, $n = 2$, $n = 65$ and $n = 959$ to establish orderings, each with a different efficiency tabulated in Table 2.

To establish a bit sequence from each of the possible (and uniformly distributed) $n!$ orderings, the algorithm described in Listing 1 was used to establish a unique integer between 0 and $n! - 1$. The binary representation of such an integer was used to create the output bits. All data processing was done after the experiment had finished.

Finally, the three bit sequences were fed into the NIST test suite [13], a number of algorithms that checks binary data and reports "fail" if some non-randomness was found and otherwise "pass". In Table 3–5, we present the results for the data generated.

Table 2 The values for n used and the corresponding (maximal) efficiency in bits per photon

n	$\mathcal{E}$, number of bits per photon
2	1/3 = 0.33
65	302/66 = 4.58
959	8122/960 = 8.46

Table 3 Proportion of passes for each individual test in the NIST test suite. The extraction used $n = 2$, thus producing 1/3 bits of entropy per click

Frequency	10/10
Block Frequency	10/10
Cumulative Sums 1	10/10
Cumulative Sums 2	10/10
Runs	9/10
Longest Run	10/10
Rank	8/10
FFT	10/10
Non Overlapping Template	9.92/10*
Overlapping Template	10/10
Universal	9/10
Approximate Entropy	9/10
Random Excursions	5.0/5*
Random Excursions Variant	5.0/5*
Serial 1	10/10
Serial 2	10/10
Linear Complexity	10/10

*Average of multiple versions of the specific test.

Table 4 Proportion of passes for each individual test in the NIST test suite. The extraction used $n = 65$, thus producing $\log_2(65!)/66$ bits of entropy per click

Frequency	10/10
Block Frequency	10/10
Cumulative Sums 1	10/10
Cumulative Sums 2	10/10
Runs	9/10
Longest Run	10/10
Rank	10/10
FFT	9/10
Non Overlapping Template	9.90/10*
Overlapping Template	10/10
Universal	10/10
Approximate Entropy	10/10
Random Excursions	6.0/6*
Random Excursions Variant	6.0/6*
Serial 1	9/10
Serial 2	10/10
Linear Complexity	10/10

*Average of multiple versions of the specific test.

4 Discussion

To establish the independence of the quantum events used in the randomness harvesting process, we have set a relatively large average time-between-clicks using an attenuated pulsed laser. The idea behind this is to let any quantum process "cool down", so that loosely speaking, any memory of a previous event is erased from the process that generates the quantum events.

Table 5 Proportion of passes for each individual test in the NIST test suite. The extraction used $n = 959$, thus producing $\log_2(959!)/960$ bits of entropy per click

Frequency	10/10
Block Frequency	10/10
Cumulative Sums 1	10/10
Cumulative Sums 2	10/10
Runs	10/10
Longest Run	10/10
Rank	10/10
FFT	10/10
Non Overlapping Template	9.85/10*
Overlapping Template	10/10
Universal	10/10
Approximate Entropy	10/10
Random Excursions	6.0/6*
Random Excursions Variant	6.0/6*
Serial 1	10/10
Serial 2	10/10
Linear Complexity	10/10

*Average of multiple versions of the specific test.

For a practical implementation, the independence criterion could alternatively be maintained by e.g., using parallel (but physically separated) source-detector setups.

Although we present a scheme for increasing efficiency in randomness harvesting, in our study we have not explored at what (higher) frequency the quantum events are no longer independent. Such undertaking, since using a higher quantum event frequency will also increase the efficiency, will be very useful for randomness extraction. We hope that future work will address this question.

To incorporate this method in a realistic device would require that the data could be processed in real time. The most computation intensive steps include calculation of large factorials and sorting, which could require tailored software since standard algorithms are typically not suited for storing and manipulating large integers such as 65!. However, such adaptations are straightforward and could be implemented on, e.g., a Field Programmable Gate Array(FPGA).

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1140/epjqt/s40507-024-00288-0.

Additional file 1. (ZIP 1.3 GB)

Acknowledgements
The authors would like to thank Gunnar Björk and Rémi Robert for valuable feedback.

Author contributions
J.A. suggested the new randomness extraction method. All authors wrote the main manuscript text. G. V.-L. and V. Z. provided their laboratory and equipment. T.N. executed the NIST test suite programs. All authors reviewed the manuscript.

Funding
Open access funding provided by Royal Institute of Technology. Vinnova grant no 2021-03865.

Data availability
The recorded times for the photon detections, the extracted randomness files and their corresponding test report can be found in the Additional file 1.

Declarations

Ethics approval and consent to participate
No humans or animals were involved in the research which is purely theoretical. Hence, ethical approval and consent to participate are irrelevant.

Consent for publication
No human subjects were involved in the research which is purely theoretical. All authors and their employers agree to publish the results.

Competing interests
The authors declare no competing interests.

Author details
[1]Ericsson Research, Ericsson AB, Stockholm, Sweden. [2]Department of Applied Physics, KTH Royal Institute of Technology, Stockholm, Sweden.

Received: 1 December 2022 Accepted: 31 October 2024 Published online: 25 November 2024

References

1. Yao AC. Theory and applications of trapdoor functions. 23rd IEEE symposium on foundations of computer science. 1982. p. 80–91.
2. Blum M, Micali S. How to generate cryptographically strong sequences of pseudo-random bits. SIAM J Comput. 1984;13:850–64.
3. Millenson JR, Sullivan GD. A hardware random number generator for use with computer control of probabilistic contingencies. Behav Res Meth Instrum. 1968;1:194–6.
4. Reidler I, Aviad Y, Rosenbluh M, Kanter I. Ultrahigh-speed random number generation based on a chaotic semiconductor laser. Phys Rev Lett. 2009;103:024102.
5. Jennewein T, Achleitner U, Weihs G, Weinfurter H, Zeilinger A. A fast and compact quantum random number generator. Rev Sci Instrum. 2000;71:1675.
6. von Neumann J. Various techniques used in connection with random digits. J Res Nat Bur Stand Appl Math. 1951;12:36–8.
7. He Y, Zhang W, Zhou H, You L, Lv C, Zhang L, Liu X, Wu J, Chen S, Ren M, Wang Z, Xie X. Bias-free true random number generation using superconducting nanowire single-photon detectors. Supercond Sci Technol. 2016;29(8):085005.
8. Yan Q, Zhao B, Hua Z, Liao Q, Yang H. High-speed quantum-random number generation by continuous measurement of arrival time of photons. Rev Sci Instrum. 2015;86(7):073113.
9. Stipčević M, Rogina BM. Quantum random number generator based on photonic emission in semiconductors. Rev Sci Instrum. 2007;78(4):045104.
10. Wikipedia contributor. Pulse-position modulation — Wikipedia, the Free Encyclopedia. https://en.wikipedia.org/wiki/Pulse-position_modulation (2022). [Online; accessed 2024-09-13].
11. McEliece R. Practical codes for photon communication. IEEE Trans Inf Theory. 1981;27(4):393–8.
12. Butman S, Katz J, Lesh J. Bandwidth limitations on noiseless optical channel capacity. IEEE Trans Commun. 1982;30(5):1262–4.
13. Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert A, Dray J, Vo S. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf (2010). Accessed on 2024-09-13.

Publisher's Note