Budapest University of
Technology and Economics

Department of Networked Systems and Services

Post-processing techniques for time-of-arrival based quantum random number generators

Ph. D. Thesis

Author:
Balázs Solymos

Supervisor:
Dr. László Bacsárdi Ph. D.
Associate Professor

Budapest, 2024

Acknowledgement

I would like to express my gratitude to my supervisor, Dr. László Bacsárdi, for supporting my journey in the field of quantum communications since my bachelor's thesis. He lent me his wisdom and experienced perspective when I needed it, both regarding technical problems and other everyday aspects of being a researcher. His support, both as my supervisor and as head of the laboratory, allowed me to pursue and stay focused on the research topics I was interested in and finish writing my dissertation in a timely manner.

I want to thank Ágoston Scharnz for the many discussions and advice, as well as his help with managing the physical experimental setup I used. His prior PhD thesis and work provided a solid foundation for my investigations and experiments and sparked the discussion that eventually led to productive cooperation with Miklos Telek. Here, I would also like to thank Miklos Telek for the joint work, during which I learned a lot from him, both in terms of mathematical modeling and general research and publication practices.

I am also grateful to the members of the Mobile Communications and Quantum Technologies Laboratory at BME Department of Networked Systems and Services for providing a friendly and helpful atmosphere.

Abstract

Unpredictability is a key resource for many computational applications like cryptography, statistical simulations, probabilistic games (e.g., gambling), fair selection, or even quantum key distribution. Quantum random number generators promise guaranteed quality, secure, fully random output by utilizing the fundamentally indeterministic nature of quantum measurements. While these generators can be based on any quantum phenomena, advancements in quantum optics are making architectures based on various properties of photons more accessible, with continuously increasing capabilities. One such architecture is photonic time-of-arrival based generators, where the non-deterministic emission times of single photons are measured. One of the main advantages of this scheme is that it can be realized with a relatively simple measurement setup while offering substantial output entropy rates. The expected output of any random number generator is a uniformly distributed series of zero and one bits. Actual physical measurement statistics, however, typically follow some other non-uniform distribution, mandating the need for various post-processing steps. Additionaly, potential imperfections and non-idealities can also influence the measurement statistics and must be handled. Thus, the non-trivial task of post-processing is to transform physical measurement results to a quality output bitstream while striving for optimal bit-generation efficiency with affordable computational costs. After briefly introducing the topic of quantum random number generation in the first chapter, this thesis explores and presents various challenges and possible solutions associated with post-processing the measurement results of time-of-arrival based quantum random number generators. The second chapter presents a scheme for dealing with unwanted correlations due to typical hardware imperfections. The third and fourth chapters present two different post-processing schemes to generate uniform output. The first is inspired by the continuous probability integral transform and can be used in practical scenarios where the measurement setup is well characterized, while the second one is based on universal hashing, offering increased error tolerances at the cost of elevated computational costs. The theoretical results and claims of the thesis are also verified experimentally.

Contents

Chapter 1

Overview of quantum random number generation

1.1 Introduction

To provide motivation for my research focusing on quantum random number generation, I first introduce the main use cases for quantum random number generators (QRNGs), focusing on classical and quantum cryptography and briefly mentioning other fields like statistical simulations. Then, I describe the basic operating principles behind QNRGs and the main idea behind their promised output quality, security, and certifiability. I continue with presenting practical QNRG realizations and their associated challenges, focusing on optical random number generators. I also introduce the most important basic information theoretical measures associated with QRNGs. I finish the chapter by discussing the topic of assessing the correct operation of random number generators via statistical testing tools.

1.2 Use cases for quantum random number generators

Unpredictability is a key resource for many computational applications like cryptography, statistical simulations, probabilistic games (e.g., gambling), or fair selection. Quantum random number generators promise guaranteed secure random output and, therefore, can be a valuable building block for any of these applications.

1.2.1 Cryptography

The goal of cryptographical protocols is to facilitate information security even in the presence of adversaries, with applications such as secure communications, secure data storage, authentication, certification, or commerce. These protocols rely on confidential secrets during operation that allow parties in possession of the secret to decrypt encrypted data or pass security challenges. This secret is often called the secret *key*. If an adversary can successfully calculate, guess, or get possession of this *key* in any way, the security of the whole application is often compromised as the adversary can masquerade using the key as the originally intended party. To avoid this, ideally, the key itself should be as hard to guess as possible, and it should not be possible to gain any information at all about the key during the protocol's operation. For a given space of possible key values, the actual key is the hardest to guess if the probability of a particular value being the key is the same for all the possibilities (equivalently, the distribution from which the key is selected should have maximum entropy and therefore should be uniform), allowing no strategy more effective than simply checking all possibilities until the key is found. Due to this, the generation of uniformly distributed sequences is crucial for secret key generation [1].

Any information about the key (even simply knowing that some values are more probable than others) can lead to more effective adversarial strategies that may reduce the level of security. For example, knowing that a particular key is selected by a human user (e.g., a password), a strategy based on ordering guesses according to word lists of the most popular password phrases is likely to be successful sooner than exhaustively trying all possible values. Similarly, vulnerabilities of supposedly uniformly random sources can also compromise the security of applications using them [2, 3].

Beyond the initial unpredictability of the key itself, security protocols should guarantee that the key stays unpredictable also during operation. This means that protocols shall guarantee that a potential adversary cannot gain any information about the key from data gathered during operation.

A simple example of a practical implementation of this principle is the one-time-pad encryption scheme [4–6]. The one-time-pad (OTP) protocol depends on a shared uniformly random key between the communicating parties. To encrypt data, the sender simply executes a bitwise XOR operation on the key and the secret message. The result is an encrypted message, called ciphertext, that the other party can then decrypt by simply performing the same bitwise operation (XOR) on the ciphertext and the key, giving back the original secret message as the result. Notice that without knowledge of the secret key, the ciphertext appears uniformly random (since

XOR-ing any sequence with a uniformly randomly distributed sequence yields a uniformly randomly distributed sequence.), and thus, the ciphertext contains no information about the key or the secret message. This property is often called perfect secrecy [7, 8], as the protocol is secure against adversaries with unlimited computing resources and time. Trying all possible keys for decryption would only produce all possible secret messages (including the original secret message), but with no way of differentiating between the possibilities, thus gaining no information about the message. The main drawback of the scheme is that the keys can only be used once, must be truly random, and be identical for the communicating parties. This often presents practically challenging demands for generating the shared secret, essentially transforming the original secure communication problem into a problem of secure key distribution.

Asymmetric cryptographic protocols (or public-key protocols) [9] aim to avoid the need for shared secrets by providing different keys for encryption and decryption, where only the key used for decryption must remain a private secret (Therefore, distribution of encryption keys can easily be done on a public channel.). The basis for their security relies on mathematical problems that are easy to compute but have computationally hard inverse problems, making computation of the private decryption key from the encrypted message and encryption key ideally infeasibly hard. Due to their ease of use, these protocols are widely used in many practical applications. The main reason for concern with them is that if an efficient solution is found for the underlying mathematical problem, the security of the whole protocol is compromised. Nowadays, the emergence of quantum computing [10, 11] compounds this potential threat of lack of future security as it is not yet clear what previously hard computational problems might be solvable efficiently (like the case of prime factorization for the RSA protocol [12]) with the new capabilities a quantum computer can offer. The field of post-quantum cryptography [13] aims to deal with this problem with new classical algorithms that are hard to compromise even with a quantum computer, but due to the still rapidly evolving field of quantum computing and quantum algorithms, proving the absolute future security of such proposals is problematic.

1.2.2 Quantum key distribution

While quantum computers [10, 11] might compromise the security of protocols previously thought to be computationally secure, quantum key distribution (QKD) protocols [14] offer promising ways of solving the problem of secure key distribution

that is presently the main challenge for many information-theoretically secure cryptographic protocols, like one-time pad encryption. Quantum key distribution protocols rely on the physical laws of quantum mechanical phenomena to guarantee security, leading to protocols that are future-proof even against any possible new computational capabilities. Paired with information-theoretically secure cryptographic protocols (e.g., one-time pad), they can offer solutions that are ideally secure. Naturally, errors, oversights, or limitations of practical implementation can still introduce vulnerabilities even in these systems [15], especially considering the infancy of the technology used for many experimental setups. While the physical carrier of quantum information can take many forms, photons are used most commonly as qubits in quantum communication due to the availability of quantum optics solutions and the interoperability with classical optical communication infrastructure.

Prepare-and-measure QKD

Prepare-and-measure quantum key distribution protocols aim to utilize the quantum no-cloning theorem [16] as their basis for security and operate on the following principle: The sender, Alina, prepares a quantum state and then sends this state to the receiver, Balázs, who then measures it. Due to the quantum no-cloning theorem, it is impossible for a potential adversary to make a perfect copy of an arbitrary sent state [16], even if it is sent on a public channel. Furthermore, manipulations of an adversary change the state of the sent quantum system due to the laws of quantum mechanics, thus making the adversary's presence detectable. Undetected eavesdropping is only possible if the preparation bases of the sent states are known apriori by the adversary, thus allowing for perfectly recreating the originally sent qubits. This, however, is easily countered by utilizing randomly selected preparation bases at the sender paired with coordinated post-measurement selection at the receiver[1]. Many QKD schemes fall into this category of protocols, including the earliest BB84 protocol [17], with numerous differing experimental realizations. While these practical realizations may differ in many things, like protocol or qubit implementation, the main operating principle largely stays the same (the previously mentioned use of the no-cloning nature of quantum information). This means that these QKD schemes have an inherent need for quality randomness for the secure selection of transmitter and receiver bases, constituting a randomness generation rate requirement proportional to their key generation rate.

[1] If Alina sends qubits prepared in a random basis and Balázs also measures with a random basis, the parties can share their selected basis after measurement and only keep measurement results where the randomly selected bases match.

Entanglement based QKD

There are QKD schemes that use quantum entanglement as their basic operation principle [18]. Entanglement allows multiple qubits to be in a state where if they are measured, the measurement results for the individual qubits are random but correlated with each other (always the same or always different if the measurement basis is the same for the parties). Ideally, without an adversary, simply distributing these entangled states to parties who then measure them in the same basis already realizes key distribution, as the measurement results should be completely random and the same for all the parties and thus can be used as a secret key. Due to the potential of eavesdropping, an additional step of randomly choosing measurement basis and some necessary post-processing and public communication steps are also part of these protocols to make detecting adversaries possible. Similarly to the case of prepare-and-measure protocols, the selection of measurement bases must be truly random to guarantee security, thus presenting a need for quality randomness generators.

Device independent protocols

Utilizing the testability of the quantum behavior of qubits, there are protocols that aim to provide security while relaxing the assumptions of trustworthiness on some or all of the hardware components realizing the protocol [19]. These protocols employ random self-testing (like executing Bell tests to verify that the measured quantum states are indeed entangled states) to verify that the components are working as advertised and the results can be trusted.
While the scope of relaxed security assumptions can range from individual components to whole setups depending on the protocol, in all of the cases, the selection of whether to run a self-test must be decided randomly (an adversary who knows when the self-test happens can alter behavior accordingly, thus avoiding detection). The generation of quality secure randomness is, therefore, a prerequisite of security for these protocols, too.

1.2.3 Other uses for QRNG

Besides cryptography, there are other fields (both classical and quantum) where the use of quality randomness is recommended, and therefore, quantum random number generators could offer efficient solutions.
One field where there is a need for a large quantity of random numbers, but the use of fast but deterministic pseudo-random number generators instead of true

random number generators (like QRNGs) is ill-advised is statistical simulations based on random sampling, like various uses of the Monte-Carlo method [20]. As these simulations are often aimed at gathering previously unknown information about processes or phenomena, using lower-quality randomness sources for the simulation runs the risk of the results being influenced by the non-ideal behavior of the used randomness source rather than the investigated process or phenomena itself.
Other quantum protocols besides quantum key distribution may also need a reliable source of randomness during operation. Similarly to QKD protocols, there are cases where there is a need for random selection during initialization or operation (like in the case of blind quantum computing [21]).
Currently, there are many differing physical realizations for both experimental quantum computers and quantum communication networks aiming for the eventual establishment of a quantum internet [22]. Depending on the application, the major architectural choices may be determined by factors other than what solution is used for random number generation. Since these applications already make use of quantum phenomena, in these cases, investigating the potential reusability of some of the already existing components as QRNGs might also prove beneficial, thus providing potentially new ways to utilize prior research results about the many possible standalone QRNG solutions.

1.3 Quantum random number generators

True random number generators (TRNGs) are considered more secure and potentially higher quality than pseudo-random number generators [23–26], as their core working principle is assumed to be non-deterministic. Generators of this category are based on unpredictable physical processes, like background radiation [27], atmospheric [28], electrical or thermal noises [29], or certain chaotic systems [30]. A diagram of a typical realization of such generators is presented in Fig. 1.1. The main problem with such classical sources is that the quality of randomness may be hard to quantify as the source is hard to certify (as the possibility of more mature future models of operation deterministically explaining these phenomena cannot be excluded.). Quantum systems are inherently non-deterministic, so they can serve as ideal sources for truly random numbers. Furthermore, the laws of quantum mechanics can be used to quantify the quality of randomness, solving the certifiability problem.
Quantum random number generators [31, 32] can be based on any physical phenomena following the rules of quantum mechanics. The earliest QNRG architectures, for example, used the unpredictability of radioactive decay as an entropy source [33, 34].

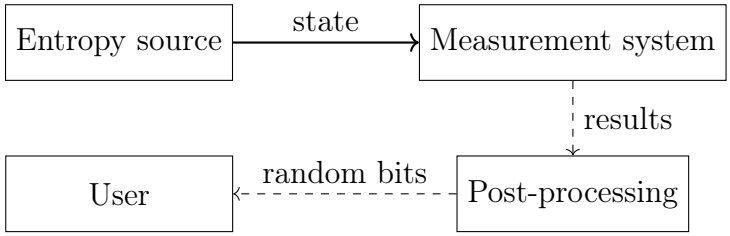

Figure 1.1: Diagram of a typical TRNG setup

Any universal quantum computer can easily realize a circuit of a simple QRNG [35], as it is shown by multiple experiments utilizing early cloud computing platforms [36–38]. Still, purpose-built generators are needed for applications with stricter quality, generation speed, or budget requirements.

1.3.1 Categorization of quantum random number generators

QRNGs are mainly categorized according to two properties. Depending on the trust placed in realizing hardware or depending on the measured quantum phenomena.

Trust level of QRNGs

Trusted QRNGs represent the first category of trust level, where the hardware components are fully trusted to operate as expected. These types of architectures are the most mature, with generation speeds ranging from a few kbps to a few hundred Gbps [39]. They offer a wide range of capabilities that can meet the needs of many use cases like portable devices [40] or even on-chip solutions [41–43]. Besides experimental setups, there are also already commercially available products offered by multiple companies [44–49].

Device-independent QRNG protocols [50–57] aim to generate secure output even when the hardware is untrusted. Their main way of achieving this through self-testing is quite similar to device-independent QKD protocols. Since, for secure self-testing, there is an initial need for randomness [50–54] or a weak randomness source [55–57] to make the testing unpredictable for a potential adversary, these QRNGs can also be thought of as devices realizing secure randomness expansion or amplification. Their other main drawback, beyond the need for initializing randomness, is the relatively low currently achievable output generation speeds (on the order of a few kbps) compared to other traditional QNRG architectures.

Semi-device-independent architectures [58] aim to strike a compromise between the higher generation speed of traditional and the higher security level of device-independent generators. Instead of being fully device-independent, they settle

for the mixed case, where some hardware elements are trusted, but some are not. Typical cases are when either the quantum source [59–61] or the measurement device [62–64] is untrusted, or instead of focusing on one of the components, some overall bound on the system is assumed [65, 66].
While they may offer higher security, they still lag behind in generation speed compared to fully trusted QRNGs. However, they can already achieve speeds up to a few Gbps for some particular realizations [61].

Physical hardware

As mentioned before, any quantum phenomena can be the basis of a QRNG. Due to the popularity of the many generator realizations based on various properties of photons and associated optical hardware, architectures are often categorized into two main groups by hardware: optical and non-optical QRNGs.
Non-optical QRNGs mainly include generators based on radioactive decay [67–69], noise in electronic circuits, like shot or thermal noise [70, 71], some atomic systems like spin noise [72] or any of the architectures aiming to realize universal quantum computers [10, 11]. The main practical drawbacks of radioactive decay based systems are the challenges associated with the materials used and their radioactivity while the output rates offered remain low, in the range of a few kbps. Generators using electrical noise can offer better generation speeds up to multiple Mbps. Ideally, one only wants to extract randomness from the shot noise due to its quantum nature, but in practice, isolating the effects of shot noise from thermal noise (influence of ambiance on the electrical carriers) is often hard. This presents the challenge of practical certifiability. Generator proposals based on atomic ensembles or quantum computer architectures are generally more complex and show low generation rates. Still, with growing interest in the field of quantum computing, they may present more viable solutions in the future as the needed technology matures.
Optical quantum random number generators present attractive architecture choices as the realizing hardware is mostly based on or compatible with already existing classical and quantum optical communications solutions. The technology used is mature, with already market-capable commercially available products [44–48] and promising experimental realizations regarding generation speed, footprint, or availability [39–43]. Additionally, all these generators typically can offer high output bit generation speeds on the order of Mbps or even Gbps. The most common optical QRNG architectures are the following:

- Photon state [73–75]: Various properties of photons can serve as a qubit. With a photonic qubit prepared in a superposition (typically path, phase, or

time superposition) and measured in a sufficient measurement basis, a simple quantum system can be realized that produces equiprobable measurement outcomes. The source of randomness is well-understood in these architectures, and they can offer generation speeds up to a few Mbps. Their main practical drawbacks are associated with the need for reliable single-photon sources and possible issues and imperfections with the detectors used. This is especially true for measurement setups requiring more detectors, as slight differences between individual detectors can lead to biased output distributions.

- Photon counting [76–79]: Multi-photon quantum states, such as the photon number of a coherent state, can also serve as a basis for a quantum entropy source. Using a weak coherent photon source (like an attenuated laser), the photon count statistic for a given time interval is expected to be Poissonian. To convert this distribution to the expected uniform output, additional post-processing steps are required after measurement. Generation speeds of these generators can go as high as a few hundred Mbps. Due to the relatively simple experimental setup needed, this architecture can also be easily adapted to multiple use cases, such as generating quantum random numbers on a mobile phone.

- Photon time-of-arrival [80–85]: One can also measure the arrival times of photons from a coherent light source. Typically, for this, only a weak coherent light source, a single photon detector, and suitable timing equipment are needed. These generators are similar to ones based on radioactive decay in the sense that both radioactive decay events and the arrival time differences between individual photon detections are expected to follow an exponential distribution. The main difference is that photons typically can be generated much faster than decay events, and the realizing hardware is often easier to handle. The exponential measurement statistics have to be converted to the expected uniform output, so a post-processing step is also required in this case. The generation speed is limited by the achievable photon counting rate and maximum time resolution of the used hardware, giving output speeds up to a few hundred Mbps.

- Vacuum fluctuations [86–88]: The zero-point fluctuation of the electromagnetic field is a quantum phenomenon and, therefore, can be used for quantum random number generation. The vacuum state can be easily prepared with high fidelity, and measurable uncertainty is present even though the vacuum state has zero photon number. The amplitude and phase quadratures are then

measured repeatedly, yielding continuous variables that can then be used for bit generation. This allows for higher generation rates of up to multiple Gbps. The main drawback of this scheme and other schemes measuring amplified quantum effects is often with certifiability since modeling of the generator is more complicated, and care must be taken to ensure that the main source of randomness is quantum, even in the presence of other possible noises and effects affecting the measurement setups.

- Amplified spontaneous emission [43, 89, 90]: QRNGs can also be based on the phase noise of amplified spontaneous emission. If the average photon number is sufficiently large, the phase uncertainty can be significantly larger than the vacuum noise, allowing for larger generation speeds on the order of multiple Gbps. The main drawbacks and challenges associated with the scheme are similar to the case of vacuum fluctuation QRNGs, which are complicated modeling and harder certifiability of the quantum nature of the output.

- Raman scattering [91–93]: Raman scattering can also be used to gather entropy in the form of randomized amplitude or phase of the output field. Amplifying this to a macroscopic, measurable level, a QRNG scheme very similar to amplified spontaneous emission QRNGs can be realized. A possible advantage of these generators is that they allow the utilization of a higher-bandwidth laser , which can lead to generation rates of hundreds of Gbps. Still, the main drawbacks and challenges of the previous two architectures apply to this scheme, too.

1.4 Information theoretical measures and limits of randomness

Besides output bit generation speed, some other measures may also be of interest regarding random number generation, especially when working with architectures based on physical phenomena. Entropy and statistical distance are two such important measures that are often used to describe a system generating random outputs.

1.4.1 Entropy

In information theory, various kinds of entropy measures aim to quantify the various kinds of uncertainty, surprise, or information content of random variables. The most general notion of entropy is the Rényi entropy, which is defined as follows [94]:

For a discrete random variable X with n possible outcomes, where the probability of the ith ($i \in \mathbb{N}, 0 < i \leq n$) outcome occuring is p_i, the Rényi entropy of order α is:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^\alpha \right), \tag{1.1}$$

where $0 < \alpha < \infty$ and $\alpha \neq 1$. In the special cases of $\alpha = 0, 1, \infty$ it is further defined as:

$$H_\alpha(X) = \lim_{\gamma \to \alpha} H_\gamma(X). \tag{1.2}$$

The base of the logarithm only changes the measurement units used (For example, the choice of 2 corresponds to "bits".). Other more specific entropy measures can then be described using the Rényi entropy. The most common measure of entropy, the Shannon entropy, corresponds to the $\alpha \to 1$ case, meaning

$$H(X) = H_1(X) = \lim_{\alpha \to 1} H_\alpha(X) = -\sum_{i=1}^{n} p_i \log p_i. \tag{1.3}$$

Another entropy measure, essential in the field of randomness extraction, is the min-entropy, to which the Rényi entropy converges in the $\alpha \to \infty$ limit:

$$H_\infty(X) = \lim_{\alpha \to \infty} H_\alpha(X) = \min_i(-\log p_i) = -\log \max_i p_i. \tag{1.4}$$

The min-entropy is important because it quantifies the maximum amount of independent, uniformly distributed random bits that can be generated from a process [95]. Since the Rényi entropies of a given distribution are non-increasing in α, $H_\infty(X) \leq H(X)$, where equality only stands in case X is uniformly distributed.

1.4.2 Statistical distance

Statistical distance quantifies the distance between two statistical objects. In the case of random number generation, it is often used to quantify the quality of the outputs, such as the distance between the ideal uniform discrete distribution and the actual distribution output by the generator. The statistical distance (or the total variation distance) between two discrete probability distributions X and Y is given by

$$\Delta(X, Y) = \frac{1}{2} \sum_v \big| \Pr(X = v) - \Pr(Y = v) \big|. \tag{1.5}$$

We say that X and Y are statistically ϵ-close if $\Delta(X, Y) \leq \epsilon$.

1.5 Statistical testing

Random number generators are expected to produce all possible output strings during normal operation. Due to this, no outputs can be designated as erroneous with certainty. This makes checking for correct operation tricky. Still, with the help of statistical testing [96], some statements about the output can be made nonetheless, albeit with varying confidence.

1.5.1 Hyphothesis testing

The main idea behind using hypothesis testing to verify the correct operation of random number generators is the following: While all output strings may arise during correct operation, the generator is expected to produce a uniformly distributed output and, therefore, should exhibit some well-defined statistical properties with high confidence. These can be checked with hypothesis testing.
Let our $\mathcal{H}_0$ null hypothesis be that the output is from a correctly operating true random number generator and our $\mathcal{H}_a$ alternate hypothesis that it is not. During testing, we gather evidence (check expected statistical behavior) trying to prove $\mathcal{H}_a$, thus refuting $\mathcal{H}_0$. Successful refusal of $\mathcal{H}_0$ means that the output is not from a correctly operating true random number generator with high confidence. Note that due to the inherently probabilistic nature of statistical testing, results must be interpreted carefully when assessing the quality of generators to avoid incorrect conclusions.

Parameters of statistical tests

Depending on the relation of a test result to reality, there are four possible outcomes as presented in Table 1.1.

Table 1.1: Possible test conclusion cases compared to reality

Reality	Test conclusion	
	Accept $\mathcal{H}_0$	Accept $\mathcal{H}_a$
Data from true random number generator	No error	Type I error
Data not from a true random number generator	Type II error	No error

The probability of a Type I error is often called the *significance level* of the test (often denoted as α). It indicates the probability that the test tags an input sequence as non-random, even when it comes from a true random number generator (false positive test result). This also means that output from a correctly working, truly

random generator is expected to fail a test with a significance level of 0.01, 1 out of 100 times on average. Common values for this parameter in cryptography range from 0.001 to 0.05.
Type II errors occur when a test indicates that the input is random even though a non-random source generated it. Calculating the probability of this error (often denoted as β) is often more difficult as it depends on the many different ways a sequence and its corresponding test statistic can be non-random. The probabilities of Type I and Type II errors are related to each other and the length of the tested sequence, so if two of them are specified, the third one can be computed.

Structure of a typical statistical test

Statistical tests investigate whether particular statistical properties of the input sequence align with the expected values for truly random sequences. The investigated property changes depending on the particular test and can range from common statistical measures like counting the occurrences of particular symbols or calculating autocorrelation of samples to more exotic ideas, like investigating compressibility or the distribution of the outcomes of random games randomized by the input sequence. In each case, a test statistic is calculated from the data, which is then compared to a critical value based on the expected ideal distribution, sequence length, and the test's target significance level. The input sequence is rejected if the calculated test statistic falls outside the interval bounded by the critical value. We can also calculate the *p-value*s (the probabilities of obtaining test results at least as extreme as the result actually observed, assuming that the null hypothesis is correct) from the test statistic to summarize the strength of evidence against the null hypothesis [97]. Possible values for *p-value*s range from 0 to 1, and *p-value*s below the significance level indicate input sequence rejection. The *p-value*s themselves can be interpreted as random variables with an expected uniform distribution when the null hypothesis is true, allowing for secondary statistical testing on the collected *p-value*s of large datasets [98].

1.5.2 Test collections

Each individual test can only indicate types of non-randomness associated with its calculated statistical property. There are infinitely many ways of possible non-random behavior and, therefore, possible statistical tests. Due to this, no set of tests can be considered complete. Running statistical tests in practice requires non-negligible computational resources, so choosing a set of tests that can efficiently check the most

probable errors of the tested generator architecture is advised. The four popular test collections I used during my work for assessing random number generators are the following:

- NIST STS [99]: A statistical test suite from the standards institute NIST (National Institute of Standards and Technology), containing 15 different types of statistical tests, looking at statistical properties such as bit frequencies or runs, cumulative runs, short template matches, expected FFT distribution or approximated entropy. In addition to providing a test suite, NIST also published recommendations for best practices for assessing RNGs. Although the originally published documents are currently under revision, they still constitute a solid guide in the field with some caveats.

  During my work, I used a reimplementation [100] of the originally published source code, in which one of the original slight implementation errors was already corrected. Due to the testing parameter set I used (default values except for the number of iterations set to 1024), the other known error [101] of the suite did not affect my results.

- Dieharder [102]: A statistical test suite inspired by and containing the original Diehard tests [103]. Later versions of the suite also contain many other tests (including some from the NIST STS). The main goal of this suite is to facilitate the testing of generators until a close to unambiguous result is reached. Due to this, it supports directly testing datastreams, as well as easy scalability of input length, while often employing secondary testing on results calculated from smaller subsections of input data to be able to give a verdict on a long input sequence with high confidence.

  During my work, I used Dieharder version 3.31.1 compiled from source code, using the `-a` flag with default parameters unless stated otherwise, with only the slight modification of additionally logging the tested data size by the tests.

- TestU01 [104]: TestU01 is a software library offering a collection of utilities for the empirical statistical testing of uniform random number generators. It provides general implementations of the classical statistical tests for random number generators as well as several other tests proposed in the literature and some original tests. It also defines some test collections recommended for various testing needs from the available tests.

  For my work, I created a small custom program to utilize the following collections from the library:

  - *Alphabit* and *Rabbit* batteries: These batteries are recommended for testing finite-length byte sequences from hardware sources, the *Alphabit* battery applying 17 and the *Rabbit* battery applying 38 different statistical tests.
  - *SmallCrush* and *Crush* batteries: These batteries are for testing sequences of random numbers between 0 and 1 (common pseudorandom number generator (PRNG) output). The *SmallCrush* battery is intended as a lightweight and fast collection of tests for a fast initial assessment and, therefore, only contains 10 different statistical tests. For more thorough testing, the *Crush* battery contains 96 tests (although, due to the required input length, running this battery may not be practically feasible for some experimental scenarios).

- ENT [105]: The ENT program differs from the previously mentioned collections in the sense that it only calculates statistics (also presents the ideal expected values) and leaves the rest of the evaluation to the user. The statistics calculated are entropy, reducibility by optimal compression, Chi-square distribution, arithmetic mean, Monte Carlo value for pi, and serial correlation coefficient.

  During my work, I used the ENT program in both bit and byte modes (`-b` flag) to evaluate measurement results.

Chapter 2

Correlation avoidance in single photon detecting quantum random number generators by dead time overestimation

In this chapter, I investigate the likely non-idealities in practical realizations of quantum random number generators based on time differences between single photon detections. To provide motivation for my later work, I first present the mathematical model of generator operation, focusing on the unwanted effects of non-idealities, such as potential correlations in the output. Then, I propose a method for eliminating the unwanted stochastic effects at the cost of reduced output interval generation speed. The proposed method is based on extending the insensitive periods after photon detections to create artificial virtual output intervals that behave as if the measurement was done with an idealized measurement setup with no dead time. I also analyze my proposed method, calculate its main performance measures, and verify the accuracy of my results with simulations. Lastly, at the end of the chapter, I present my experimental measurement results to support my previous theoretical claims further.

2.1 Principle of operation

A whole family of QRNGs operates based on measuring the elapsed time between single photon detections. Ideally, photon arrival times from an attenuated laser source (typically also including attenuation due to detector quantum efficiency) are independent of each other, with exponentially distributed arrival time differences

between consecutive arrival events [106], forming a Poisson point process (PPP) with rate λ, where λ is the average photon detection rate. Although, in reality, other non-ideal photon emission effects (like small superpoissonian contributions from thermal effects) may also be present, the overall effect of these has been shown to be vanishing when introducing attenuation into the system [107]. Typically, the elapsed time is measured utilizing fast-running clock signals, counting the number of elapsed clock periods between detection events. Due to finite time measurement precision in practical scenarios, these measurement statistics, which I will refer to as *discretized time differences* (DTDs), are discretized values and, therefore, should follow a geometric distribution derived from the actual exponential arrival time differences. Depending on the time measurement scheme used, Time-of-Arrival (ToA) QRNGs can be further divided into two groups:

- Utilizing restartable clocks: It is shown [80] that restarting the clock signal after each detection is advantageous. It guarantees a fixed measurement starting point relative to the clock signal, yielding identically distributed, independent result DTDs. The main drawback of this scheme is the need for hardware that can precisely and reliably execute the restart operation.

- Utilizing non-restartable clocks: Without restarting the clock signal after each detection, the starting point of each measurement is not fixed relative to the clock signal; the distributions of consecutive measurement results are not guaranteed to be identical anymore as they vary depending on the timewise location of the measurement starting point relative to the clock signal. More importantly, since each measurement starting point is the detection event of the previous photon, unwanted correlations arise between consecutive measurement results [108]. The hardware requirements of this method, however, are much simpler, offering a potentially cheaper or even faster physical realization, provided that the introduced non-idealities can be handled.

These two methods of operation are shown in Fig. 2.1.

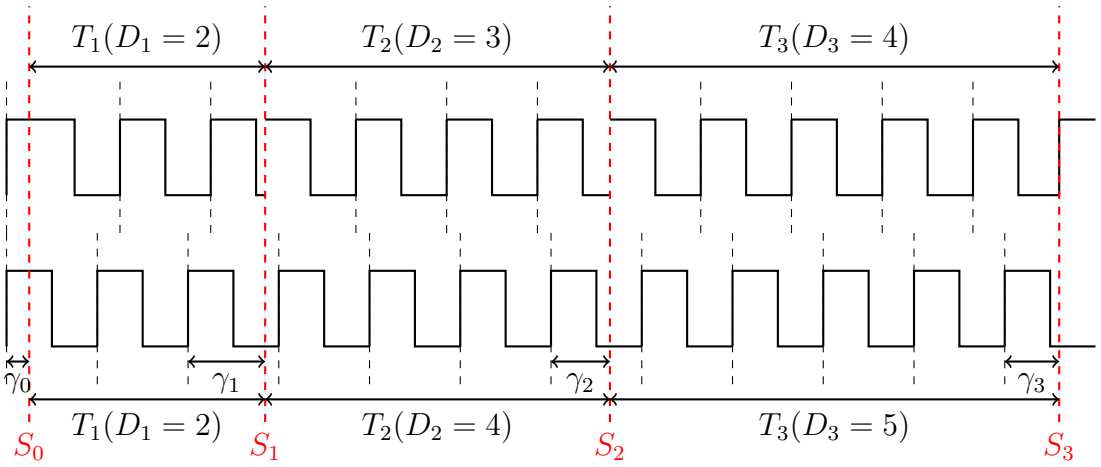

Figure 2.1: Example of restartable (top) and non-restartable clocks (bottom). S_0, S_1, S_2, and S_3 are the actual photon detection times, T_1, T_2, T_3 are the intervals responsible for the D_1, D_2, D_3 counts. Additionally, in the case of non-restartable clocks (bottom), γ_0, γ_1, γ_2, and γ_3 are the phases of the respective photon detections.

2.2 Mathematical model of operation

A single-photon detector detects photons from a suitably attenuated continuous-wave (CW) laser, and a time-tagger card assigns time stamps to detections. The photons are assumed to arrive according to a Poisson point process with rate λ, valid for coherent light sources [106]. Let S_i denote the ith photon arrival time assuming $S_0 = 0$, and $T_i = S_i - S_{i-1}$ the ideally exponentially distributed time elapsed between S_i and S_{i-1}. These times are physically measured by counting the clock signal's leading edges between S_i and S_{i-1} yielding integer values. These values are the *discretized time differences*, discrete random variables denoted by D_i. DTDs undergo well-defined mathematical operations based on the applied random bit generation scheme (e.g., [80]), yielding random bits, ideally forming uniformly distributed, uncorrelated sequences.

The clock's period, which is the time-tagger's resolution, is denoted by τ. Additionally, in the case of a non-restartable time tagging clock, there is a non-zero γ_i time between S_i and the previous leading clock edge, that is, $\gamma_i = S_i - \lfloor S_i/\tau \rfloor \tau$. Consequently, $\gamma_i \in [0, \tau)$ (where $\lfloor \cdot \rfloor$ denotes the floor function, representing the greatest integer less than or equal to its argument). The random variable γ_i is called the *phase* of the ith photon detection. As mentioned before, an example case of operation with a restartable and non-restartable clock showing the above-presented notations is presented in Fig. 2.1.

2.2.1 Distribution of the observed variables

As previously mentioned, the lengths of the T_i time intervals between photon detections from a Poisson point process follow an exponential distribution with parameter λ. While in reality, the λ photon rate is never truly constant owing to e. g., thermal effects, in the following model, I assume it to be constant as my focus will be on other more impactful potential error sources since fluctuations of λ can be handled by properly regulating the laser output intensity. Therefore, in the model the F_T cumulative distribution function (CDF) and the f_T probability density function (PDF) of T_i as a function of elapsed time t are the following:

$$F_T \triangleq \Pr(T_i < t) = \begin{cases} 1 - \mathrm{e}^{-\lambda t} & \text{if } t \geq 0, \\ 0 & \text{if } t < 0, \end{cases} \tag{2.1}$$

and

$$f_T \triangleq \frac{\mathrm{d}}{\mathrm{d}t} \Pr(T_i < t) = \begin{cases} \lambda \mathrm{e}^{-\lambda t} & \text{if } t \geq 0, \\ 0 & \text{if } t < 0. \end{cases} \tag{2.2}$$

Since consecutive T_i values are independent, the CDF and PDF are independent of the lower index i and the same for all T_i. In practice, we do not have access to T_i; therefore, the observable variables upon which the QRNG is built are the D_i samples. More importantly, since these values are the input of the bit generation process, their distribution directly affects the final output of the generator.

Restartable clock

In the case of a restartable measurement clock, the probability mass function (PMF) of the discrete D_i samples can directly be calculated from (2.1):

$$\begin{aligned} \Pr(D_i = n) &= \Pr\big(n\tau \leq T_i < (n+1)\tau\big) = F_T((n+1)\tau) - F_T(n\tau) \\ &= \left(1 - \mathrm{e}^{-\lambda(n+1)\tau}\right) - \left(1 - \mathrm{e}^{-\lambda n\tau}\right) = \mathrm{e}^{-n\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right), \end{aligned} \tag{2.3}$$

where $n \in \mathbb{N}$ (including 0). Substituting $p = 1 - \mathrm{e}^{-\lambda\tau}$ we can rewrite (2.3) in the form of a $\Pr(D_i = n) = (1-p)^n p$ geometric distribution, showing that the discrete D_i DTDs retain the memoryless property [109] of the underlying T_i exponential time differences.

Continuous clock

When utilizing a continuously running measurement clock, the starting points of the T_i time intervals no longer align with the start of clock periods. An additional new γ_i variable describing the *phase* of each detection is needed.

According to Ref. [108], for the distribution of the observed variables and corresponding phases, focusing only on the first arrival, for $x, y \in [0, \tau)$ we can write

$$\begin{aligned} F_n(x,y) &\triangleq \Pr\big(D_1 = n, \gamma_1 < y \mid \gamma_0 = x\big) \\ &= \begin{cases} \Pr(x + T < y) & \text{if } n = 0, \\ \Pr(n\tau \le x + T < n\tau + y) & \text{if } n > 0, \end{cases} \\ &= \begin{cases} \chi_{\{y>x\}} \left(1 - \mathrm{e}^{-\lambda(y-x)}\right) & \text{if } n = 0, \\ \mathrm{e}^{\lambda x} \left(1 - \mathrm{e}^{-\lambda y}\right) \mathrm{e}^{-\lambda n\tau} & \text{if } n > 0, \end{cases} \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} f_n(x,y) &\triangleq \frac{\mathrm{d}}{\mathrm{d}y} \Pr\big(D_1 = n, \gamma_1 < y \mid \gamma_0 = x\big) \\ &= \begin{cases} \chi_{\{y>x\}} \lambda \mathrm{e}^{-\lambda(y-x)} & \text{if } n = 0, \\ \lambda \mathrm{e}^{-\lambda(y+n\tau-x)} & \text{if } n > 0, \end{cases} \end{aligned} \tag{2.5}$$

where χ_A is the indicator of the set A. Note that if $\gamma_0 = 0$ then $F_n(0,\tau) = \Pr\big(D_1 = n \mid \gamma_0 = 0\big) = \left(1 - \mathrm{e}^{-\lambda\tau}\right) \mathrm{e}^{-\lambda\tau n}$ results in a geometric distribution [110], retaining the memoryless property of the underlying exponential distribution as it describes the phase situation equivalent to the previous case of the restartable clock. The unconditional distribution of the DTDs can be obtained by weighting with the stationary $f_\gamma(y) = \frac{1}{\tau}$ distribution of the phase γ (See Appendix A.1), from which the marginal distribution of D is

$$\begin{aligned} \Pr(D = n) &= \int_{x=0}^{\tau} \frac{1}{\tau} \Pr\big(D = n \mid \gamma_0 = x\big)\, \mathrm{d}x \\ &= \begin{cases} \int_{x=0}^{\tau} \frac{1}{\tau} (1 - \mathrm{e}^{-\lambda(\tau-x)})\, \mathrm{d}x & \text{if } n = 0, \\ \int_{x=0}^{\tau} \frac{1}{\tau} \left(1 - \mathrm{e}^{-\lambda\tau}\right) \mathrm{e}^{-\lambda(n\tau-x)}\, \mathrm{d}x & \text{if } n > 0. \end{cases} \\ &= \begin{cases} 1 - \frac{1-\mathrm{e}^{-\lambda\tau}}{\lambda\tau} & \text{if } n = 0, \\ \mathrm{e}^{-\lambda\tau n} \frac{(1-\mathrm{e}^{-\lambda\tau})^2}{\lambda\tau \mathrm{e}^{-\lambda\tau}} & \text{if } n > 0. \end{cases} \end{aligned} \tag{2.6}$$

This way, D is geometrically distributed with irregular initial probability, that is, $\Pr(D = n)$ form a geometric series from $n = 1$ to infinity, but $\Pr(D = 0)$ is different from the 0th element of that geometric series.

Similarly, the conditional joint distribution of $D_1, \ldots, D_\ell$ can be written as

$$\begin{aligned} &\Pr\big(D_1 = n_1, \ldots, D_\ell = n_\ell \mid \gamma_0 = x\big) \\ &= \int_{x_\ell=0}^{\tau} \cdots \int_{x_1=0}^{\tau} \prod_{m=1}^{\ell-1} f_{n_m}(x_{m-1}, x_m)\, \mathrm{d}x_1 \ldots \mathrm{d}x_\ell \end{aligned} \tag{2.7}$$

and the unconditional one, as

$$\begin{aligned}&\Pr(D_1 = n_1, \ldots, D_\ell = n_\ell)\\ &= \int_{x_\ell=0}^{\tau} \ldots \int_{x=0}^{\tau} \frac{1}{\tau} \prod_{m=1}^{\ell-1} f_{n_m}(x_{m-1}, x_m)\,\mathrm{d}x \ldots\, \mathrm{d}x_\ell.\end{aligned} \tag{2.8}$$

The last expression indicates that the $D_1, \ldots, D_\ell$ variables are correlated. Thus, using the $D_1, \ldots, D_N$ DTD sequence for random bit generation might result in correlated bit sequences.

2.2.2 Dead time

Physical devices typically are unable to observe all successive photon arrivals. Detectors and time-tagging electronics usually have a dead time, an insensitive time interval of length ζ after a detected photon arrival, during which they cannot register any new arrivals. While multiple models of dead time exist [111, 112], in my model, I restrict dead time to be constant and non-extendable, assuming it to be in the form $\zeta = k\tau + \delta$, with $0 \leq \delta < \tau$.

This means that after a detection at S_i, no photons arriving before $S_i + \zeta$ are recognized. Consequently, for the observed photon arrivals $S_i > S_{i-1} + \zeta$ holds for $\forall i > 0$. The model assumes that photon arrivals during the dead time interval are undetected, and such arrivals do not reset the dead time. The distribution of the T_i values is then:

$$F_T \triangleq \Pr(T_i < t) = \begin{cases} 1 - \mathrm{e}^{-\lambda(t-\zeta)} & \text{if } t \geq \zeta, \\ 0 & \text{if } t < \zeta, \end{cases} \tag{2.9}$$

and

$$f_T \triangleq \frac{\mathrm{d}}{\mathrm{d}t} \Pr(T_i < t) = \begin{cases} \lambda \mathrm{e}^{-\lambda(t-\zeta)} & \text{if } t \geq \zeta, \\ 0 & \text{if } t < \zeta. \end{cases} \tag{2.10}$$

Additionally, dead time has an effect on measured count rates since $S_i > S_{i-1} + \zeta$ in all cases, ruling out too fast consecutive detections. Therefore, the mean time between photon observations is

$$\mathbb{E}(S_i - S_{i-1}) = \mathbb{E}(T_i) = \frac{1}{\lambda} + \zeta = \frac{1 + \lambda\zeta}{\lambda}. \tag{2.11}$$

As a consequence, the average rate at which the D_i samples are obtained is

$$\begin{aligned}\lambda_\mathrm{d} &= \lim_{c\to\infty} \frac{\text{observed photon arrivals in } [0, c\tau)}{c\tau}\\ &= \frac{1}{\mathbb{E}(S_i - S_{i-1})} = \frac{1}{\mathbb{E}(T_i)} = \frac{\lambda}{1 + \lambda\zeta}.\end{aligned} \tag{2.12}$$

Similarly to the previous cases without dead time, we can compute the distribution of the D_i values using (2.9).

Restartable clock

The PMF of the D_i samples, according to (2.9) with dead time is the following:

$$\begin{aligned}\Pr(D_i = n) &= \Pr\big(n\tau \le T_i < (n+1)\tau\big) \\ &= \begin{cases} 0 & \text{if } n < k, \\ 1 - \mathrm{e}^{-\lambda(\tau-\delta)} & \text{if } n = k, \\ \mathrm{e}^{-\lambda((n-k)\tau)-\delta}\left(1 - \mathrm{e}^{-\lambda\tau}\right) & \text{if } n > k. \end{cases}\end{aligned} \tag{2.13}$$

This distribution depends on k as a shifting factor and is geometric with an irregular initial probability depending on δ. Importantly, since both k and δ are constant and independent of other detections, consecutive D_i values are i.i.d. (independent and identically distributed).

Continuous clock

Similarly to the case without dead time, we can write the conditional distribution as follows:

$$\begin{aligned}F_n(x,y) &= \Pr\big(D_1 = n, \gamma_1 < y \mid \gamma_0 = x\big) \\ &= \begin{cases} 0 & \text{if } n < k, \\ \Pr(x + T + \delta < y) & \text{if } n = k, \\ \Pr\left((n-k)\tau \le x + T + \delta < (n-k)\tau + y\right) & \text{if } n > k, \end{cases} \\ &= \begin{cases} 0 & \text{if } n < k, \\ \chi_{\{x+\delta<y\}}\left(1 - \mathrm{e}^{-\lambda(y-x-\delta)}\right) & \text{if } n = k, \\ \chi_{\{\tau<x+\delta<y\}}\left(1 - \mathrm{e}^{-\lambda(y-x-\delta)}\right) & \\ \quad + \chi_{\{x+\delta<\tau\}}\mathrm{e}^{-\lambda(\tau-x-\delta)}\left(1 - \mathrm{e}^{-\lambda y}\right) & \text{if } n = k+1, \\ \left(\mathrm{e}^{-\lambda((n-k)\tau-x-\delta)}\right)\left(1 - \mathrm{e}^{-\lambda y}\right) & \text{if } n > k+1, \end{cases}\end{aligned} \tag{2.14}$$

and the conditional density is $f_n(x,y) = \frac{\mathrm{d}}{\mathrm{d}y}F_n(x,y)$. According to (2.14) the resulting distribution is dependent on x and therefore γ_0, showing that irrespective of the presence or absence of dead time, intervals tagged with a continuously running clock yield correlated measurement samples.
The unconditional marginal distribution of D_i is then:

$$\Pr(D_1 = n) = \int_{x=0}^{\tau} \frac{1}{\tau} \Pr\big(D_1 = n \mid \gamma_0 = x\big)\, \mathrm{d}x$$

$$= \begin{cases} 0 & \text{if } n < k, \\ \int_{x=0}^{\tau-\delta} \frac{1}{\tau}\big(1 - \mathrm{e}^{-\lambda(\tau-x-\delta)}\big)\, \mathrm{d}x & \text{if } n = k, \\ \int_{x=0}^{\tau-\delta} \frac{1}{\tau}\big(\mathrm{e}^{-\lambda(\tau-x-\delta)}\big)\left(1 - \mathrm{e}^{-\lambda\tau}\right)\, \mathrm{d}x & \\ \quad + \int_{x=\tau-\delta}^{\tau} \frac{1}{\tau}\big(1 - \mathrm{e}^{-\lambda(2\tau-x-\delta)}\big)\, \mathrm{d}x & \text{if } n = k+1, \\ \int_{x=0}^{\tau} \frac{1}{\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda((n-k)\tau-x-\delta)}\, \mathrm{d}x & \text{if } n > k+1. \end{cases} \tag{2.15}$$

$$= \begin{cases} 0 & \text{if } n < k, \\ \frac{\mathrm{e}^{-\lambda(\tau-\delta)} + \lambda(\tau-\delta) - 1}{\lambda\tau} & \text{if } n = k, \\ \frac{\mathrm{e}^{-2\lambda\tau}(\mathrm{e}^{\lambda\tau}-1)(\mathrm{e}^{\lambda\tau}-\mathrm{e}^{\lambda\delta})}{\lambda\tau} + \frac{\lambda\delta - (\mathrm{e}^{\lambda\delta}-1)\mathrm{e}^{-\lambda\tau}}{\lambda\tau} & \text{if } n = k+1, \\ \frac{(\mathrm{e}^{\lambda\tau}-1)^2\mathrm{e}^{\lambda(\delta-(n-k+1)\tau)}}{\lambda\tau} & \text{if } n > k+1. \end{cases}$$

Correlation of measurement samples

The $\rho_{D_i,D_{i+1}}$ correlation between successive DTD samples, D_i and D_{i+1} (which is equivalent to the lag-1 autocorrelation coefficient in DTD sequences), can be derived using

$$\rho_{D_i,D_{i+1}} = \frac{\mathbb{E}(D_i D_{i+1}) - \mathbb{E}(D_i)\,\mathbb{E}(D_{i+1})}{\sqrt{\left(\mathbb{E}\left(D_i^2\right) - \mathbb{E}(D_i)^2\right)\left(\mathbb{E}\left(D_{i+1}^2\right) - \mathbb{E}(D_{i+1})^2\right)}}. \tag{2.16}$$

Detailed calculations of the terms in (2.16) are presented in Appendix A.2.

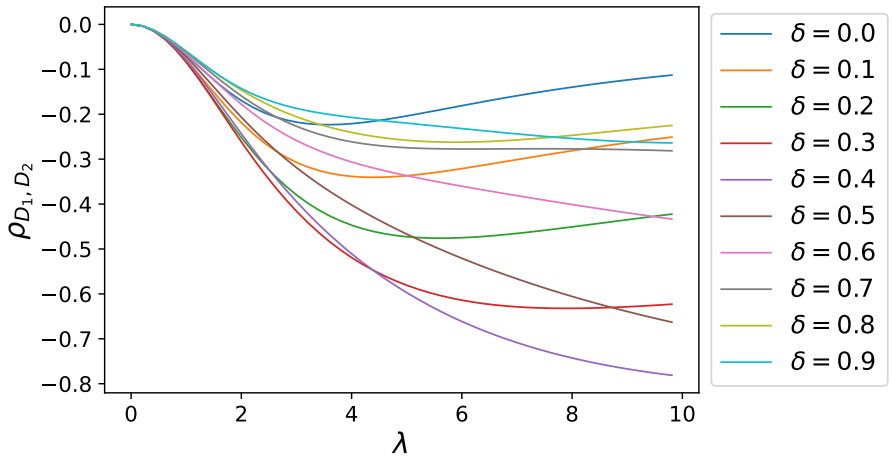

Figure 2.2: Correlation of consecutive DTDs as a function of the input photon rate λ and fractional dead time δ, with $\tau = 1$.

Figure 2.2 depicts the correlation of consecutive DTDs as a function of the photon arrival rate for selected dead time values. Note that the correlation is independent of the integer part of the dead time, k, and only its fractional part, δ, affects the

values. The figure shows that the correlation tends to zero as the photon arrival rate decreases to zero, but for higher photon arrival rates, the correlation strongly depends on the dead time. Due to this, reducing the input photon rate offers a possible way of mitigating correlations between DTDs. Still, this straightforward solution of reducing optical power can never fully eliminate correlations and comes with the drawback of reduced output speed.

2.2.3 Effect on bit generation

To demonstrate the possible effects of using the different measurement clocks, consider the simple bit generation method of Ref. [80] further analyzed in Ref. [113] and realized in our laboratory test setup.

The bit generation procedure for generating B_i bits from the D_i samples is as follows: Compare subsequent measurement results so that if $D_{2i-1} > D_{2i}$, a zero bit is generated, if $D_{2i-1} < D_{2i}$ a one bit is generated. Due to discretization, equalities can also happen and must be discarded to keep the bit distribution uniform. Therefore, in the case $D_{2i-1} = D_{2i}$, the D_{2i-1} and D_{2i} samples are dropped and no bit is generated, limiting efficiency.

We can formulate this procedure as follows. Subsequent measurements (D_{2i-1} and D_{2i}) are compared, and an R_i sequence is obtained as

$$R_i = \operatorname{sgn}\left(D_{2i} - D_{2i-1}\right) = \begin{cases} -1 & \text{if } D_{2i-1} > D_{2i}, \\ 0 & \text{if } D_{2i-1} = D_{2i}, \\ 1 & \text{if } D_{2i-1} < D_{2i}. \end{cases} \tag{2.17}$$

Let b_n denote the nth non-zero element of the R_i sequence. Then, the ith bit is generated as

$$B_i = \begin{cases} 0 & \text{if } R_{b_i} = -1, \\ 1 & \text{if } R_{b_i} = 1. \end{cases} \tag{2.18}$$

Restartable clock

It can be seen that, the introduced bit generation method produces uniform output for any input D_i distribution as long as the samples are i.i.d. Luckily, according to Sec. 2.2.2, when using a restartable measurement clock, this is the case, even in the presence of dead time, thus producing the expected uniform output.

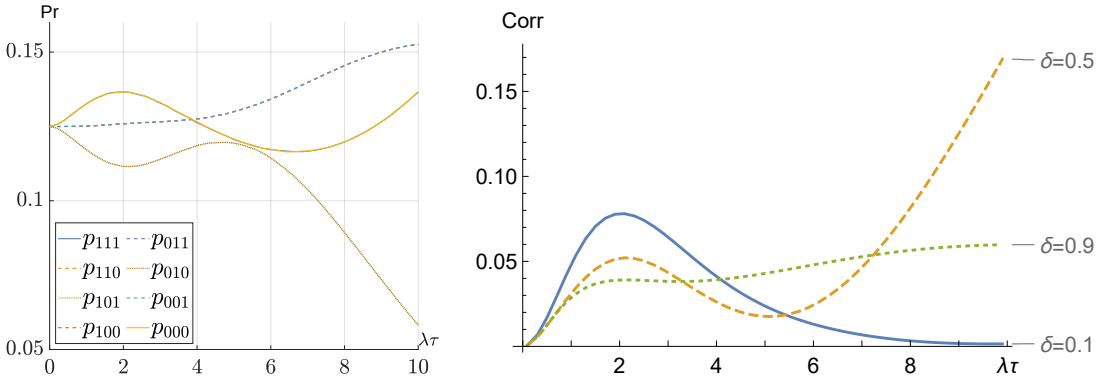

Figure 2.3: Bit triplet probabilities for $\delta = 0.5$ and lag-1 correlation for different δ values as a function of normalized photon arrival rate from Ref. [108].

Continuous clock

According to Sec. 2.2.2, the D_i samples measured with a continuously running clock cannot be considered i.i.d.; therefore, bits generated from the samples are not uniform. Unfortunately, the exact non-uniform bit distribution arising in this non-ideal case cannot be calculated in a straightforward analytical way. To remedy this, we developed an Erlang distribution based stochastical estimation approach (see Ref. [108]) for calculating the relevant performance measures. Results of the approach (presented in Fig. 2.3) clearly show major deviations from uniformity, both in bit series probabilities and correlation between consecutive bits, presenting a major problem when the goal is generating good quality uniform output bit sequences.

2.3 Dead time overestimation

To eliminate the correlation between successive D_i values, I introduce an approach called the *overestimation* of dead time. The approach is based on the following observation. The conditional distribution in (2.14) is such that for $n > k+1$ the conditional characteristic function

$$\begin{aligned}
\bar{F}_n(x,y) &= \Pr\big(D_1 = n, \gamma_1 < y \mid \gamma_0 = x, D_1 > k+1\big) \\
&= \frac{\Pr\big(D_1 = n, \gamma_1 < y \mid \gamma_0 = x\big)}{\sum_{j=k+2}^{\infty} \Pr\big(D_1 = j \mid \gamma_0 = x\big)} \\
&= \frac{\left(\mathrm{e}^{-\lambda\left((n-k)\tau - x - \delta\right)}\right)\left(1 - \mathrm{e}^{-\lambda y}\right)}{\sum_{j=k+2}^{\infty}\left(\mathrm{e}^{-\lambda\left((j-k)\tau - x - \delta\right)}\right)\left(1 - \mathrm{e}^{-\lambda\tau}\right)} \\
&= \mathrm{e}^{-(n-(k+2))\lambda\tau}\left(1 - \mathrm{e}^{-\lambda y}\right)
\end{aligned} \tag{2.19}$$

is independent of x and δ, and satisfies

$$\begin{aligned}&\Pr\big(D_1 = n, \gamma_1 < y \mid \gamma_0 = x, D_1 > k+1\big)\\ &= \underbrace{\Pr\big(D_1 = n \mid \gamma_0 = x, D_1 > k+1\big)}_{\mathrm{e}^{-(n-(k+2))\lambda\tau}\left(1-\mathrm{e}^{-\lambda\tau}\right)} \cdot \underbrace{\Pr\big(\gamma_1 < y \mid \gamma_0 = x, D_1 > k+1\big)}_{\frac{1-\mathrm{e}^{-\lambda y}}{1-\mathrm{e}^{-\lambda\tau}}},\end{aligned} \tag{2.20}$$

that is, D_1 and γ_1 are independent when $D_1 > k+1$. This also means that D_2, which depends on γ_1, will be independent of D_1 as long as $D_1 > k+1$.

Thus, the correlation of the consecutive D_i values comes from the small samples; i.e., when $D_i = k$ or $D_i = k+1$, then D_i and D_{i+1} are correlated. This property can be exploited in the overestimation algorithm to avoid unwanted correlations.

In the following sections, unless the unit of time is specified explicitly, I assume τ and ζ to have arbitrary, unspecified time units, while λ is measured in [counts]/[unit of time].

Let us overestimate the dead time with an interval covering m clock cycles, where $m \in \mathbb{Z}^+$ such that $\zeta = k\tau + \delta \leq m\tau$. I refer to m as the overestimation parameter. After a detection event, an $m\tau$ long safety interval is started from the next rising clock edge. If a photon is detected after the dead time is over but before this safety interval has ended, the detection event is discarded from any further calculations and the safety interval is extended by $m\tau$, counted from the following rising edge. Suppose the safety interval is eventually over because no early detection extends it further. In this case, we continue using our bit generation method as if the previous detection happened at the end of the safety interval. That is, we count the next time difference between the end of the safety interval and the next detection time, then digitize it. See an example in Fig. 2.4. This approach can be thought of as an algorithm taking the $\mathbb{D} = \{ D_1, D_2, \dots, \}$ DTDs as input and outputting the $\mathbb{V} = \{ V_1, V_2, \dots, \}$ *virtual DTDs* (vDTDs). The algorithm (described in Algorithm 1.) has the added benefit of placing the starting points of measurable intervals right to the beginning of a clock cycle, essentially realizing the ideal case of $\gamma_{i-1} = 0$, yielding geometrically distributed vDTDs.

Algorithm 1 Algorithm of the overestimation method

Require: m ▷ Overestimation parameter

1: **while** True **do**
2: get D ▷ Obtain last DTD at a new detection
3: **if** $D > m$ **then** ▷ Check if safety interval is over
4: $V \leftarrow D - (m+1)$ ▷ Generate V virtual DTD
5: **end if**
6: **end while**

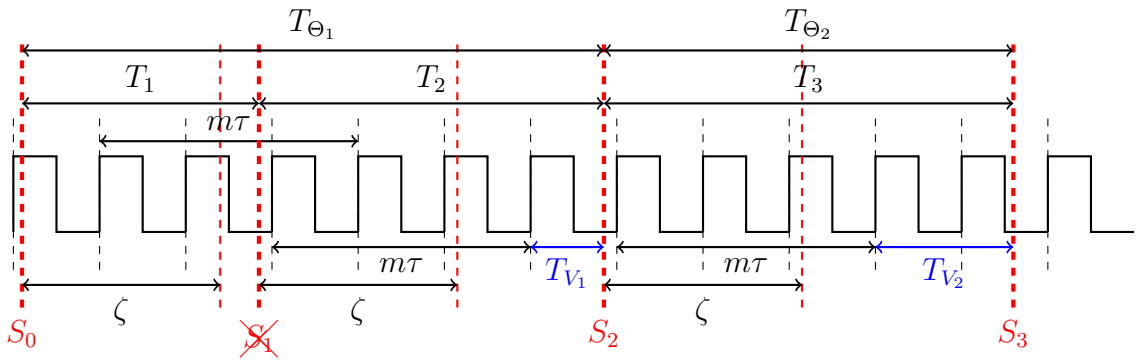

Figure 2.4: Example of the overestimation method with overestimation parameter m and dead time ζ ($m = 3, \zeta = 2.3, \tau = 1$). The square signal represents the measurement clock. Thick red dashed lines at S_0, S_1, S_2, and S_3 denote actual photon detection times, and lighter red lines show the end of the corresponding dead times. T_1, T_2, T_3 are the intervals responsible for the $D_1 = 2$, $D_2 = 4$, $D_3 = 5$ DTDs without overestimation. The photon detected at S_1 arrives before the safety interval is over, which is therefore dropped by the overestimation algorithm. T_{V_1} and T_{V_2} note the resulting virtual intervals considered in the method, responsible for $V_1 = 0$, $V_2 = 1$ virtual DTDs, while T_{Θ_1} and T_{Θ_2} are the intervals responsible for $\Theta_1 = 6$ and $\Theta_2 = 5$, with $\beta_1 = \{2, 4\}$ and $\beta_2 = \{5\}$ respectively. (For the notation β_ℓ, Θ_ℓ, T_{Θ_ℓ}, refer to Sec. 2.3.1.)

Let $\mathbb{S} = \{ S_0, S_1, \dots, \}$ be the observed photon arrival times with dead time ζ (that is, $\forall i$: $S_i > S_{i-1} + \zeta$) and $\mathbb{D} = \{ D_1, D_2, \dots, \}$ be the sequence of measured DTDs associated with $\mathbb{S}$. Let $\mathbb{V} = \{ V_1, V_2, \dots, \}$ be the virtual output DTD sequence generated by Algorithm 1 from $\mathbb{D}$.

Theorem 1. *The virtual output DTD sequence generated by Algorithm 1, $\mathbb{V}$, is composed of i.i.d. elements with geometric distribution:* $\Pr(V_\ell = n) = (1 - \mathrm{e}^{-\lambda\tau})\mathrm{e}^{-\lambda\tau n}$.

Proof. For the distribution of DTDs D_i, we can write:

$$\begin{aligned}
&\Pr\big(D_i = n \mid \gamma_{i-1} = x_{i-1}, D_i > m\big) \\
&= \frac{\left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda((n-k)\tau - x_{i-1} - \delta)}}{\sum_{j=m+1}^{\infty}\left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda((j-k)\tau - x_{i-1} - \delta)}} = \frac{\mathrm{e}^{-\lambda n\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right)}{\mathrm{e}^{-\lambda\tau(m+1)}} \\
&= \left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda(n-(m+1))\tau},
\end{aligned} \tag{2.21}$$

where γ_{i-1} is the arrival phase of S_{i-1}. Using the $V \leftarrow D - (m+1)$ assignment rule in line 4 of Algorithm 1, we have

$$\begin{aligned}
&\Pr\big(V_\ell = n \mid \gamma_{i-1} = x_{i-1}\big) \\
&= \Pr\big(D_i = (m+1) + n \mid \gamma_{i-1} = x_{i-1}, D_i > m\big) \\
&= \left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda(n+m+1-(m+1))\tau} = \left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda n\tau}
\end{aligned} \tag{2.22}$$

for the distribution of the V_ℓ variable, which is independent of the phase γ_{i-1}. □

Note that the resulting PMF of the V vDTDs is the same as the one presented in (2.3) for the case of using a restartable measurement clock without dead time. Also note that without dead time, the choice of $V \leftarrow D-1$ would be sufficient since it removes the first fractional clock period, which is responsible for the correlation of successive samples in this case. Additionally removing m full-length clock periods does not affect the discrete distribution of samples [110].

Also note that the idea of reducing optical power (presented in Sec. 2.2.2) to mitigate correlations works, since by decreasing the optical power, the probability $\Pr(D_i \leq k+1)$ decreases, consequently reducing the number of DTDs causing correlations. However, this probability will never become identically zero (unless λ is set to zero, preventing bit generation). Algorithm 1, on the other hand, removes every problematic DTD, yielding a theoretically correlation-free sequence of virtual DTDs.

Using Algorithm 1 comes at a cost, as the time used to overestimate the dead time cannot be used for bit generation, leading to decreased vDTD and, therefore, decreased bit generation rate.

2.3.1 Virtual DTD generation rate

For the performance assessment of Algorithm 1, let us define the u-long subsequence of $\mathbb{D}$, $\beta_\ell = \{ D_i, D_{i+1}, ..., D_{i+u-1} \}$, responsible for generating the ℓth vDTD, V_ℓ. According to the algorithm, β_ℓ starts with an uninterrupted run of zero or more DTDs smaller than or equal to m and ends with a single element greater than m ($D_{i-1} > m$ and $D_{i+u-1} > m$, but $D_t \leq m \ \forall t \in (i, i+u-2)$, $t \in \mathbb{N}$). Note that the set of all such subsequences, $\{ \beta_\ell \}$, is a partition of $\mathbb{D}$, since $\forall i: \ D_i \in \bigcup_\ell \beta_\ell$ and $\big(D_i \in \beta_x \wedge D_i \in \beta_y\big) \Rightarrow \big(\beta_x = \beta_y\big)$.

The number of elapsed clock signal edges between generating $V_{\ell-1}$ and V_ℓ is $\Theta_\ell = \sum_{k=0}^{u-1} D_{i+k}$, where u is the length of β_ℓ and Θ_ℓ is the sum of β_ℓ's elements.

Similar to λ_d, define λ_v, the *virtual count rate* at which the vDTDs are generated, as

$$\lambda_\mathrm{v} = \lim_{c\to\infty} \frac{\text{number of vDTDs } V_\ell \text{ generated in } [0, c\tau)}{c\tau}. \tag{2.23}$$

Theorem 2. *The virtual count rate λ_v can be expressed as*

$$\lambda_v = \frac{\mathrm{e}^{-\lambda((m+1)\tau-\zeta)}\left(\mathrm{e}^{\lambda\tau}-1\right)}{\tau(\lambda\zeta+1)}. \tag{2.24}$$

Proof. Consider the $\{ Z_0, Z_1, \dots \}$ sequence, where for $i \geq 0$

$$Z_i = \begin{cases} 0 \text{ if } D_i \leq m, \\ 1 \text{ if } D_i > m. \end{cases} \tag{2.25}$$

The sum $S_N = \sum_{i=0}^{N} Z_i$ then gives the number of vDTDs generated by Algorithm 1 from an original N-long $\{ D_0, D_1, \ldots, D_N \}$ DTD sequence. We can then write

$$\begin{aligned} &\Pr\big(Z_i = 1 \mid \gamma_{i-1} = x_{i-1}, D_{i-1} = n_{i-1}\big) \\ &= \Pr\big(D_i > m \mid \gamma_{i-1} = x_{i-1}, D_{i-1} = n_{i-1}\big) \\ &= \sum_{n=m+1}^{\infty} \mathrm{e}^{-\lambda(n\tau-\zeta-x_{i-1})} \left(1 - \mathrm{e}^{-\lambda\tau}\right) \\ &= \Pr\big(D_i > m \mid \gamma_{i-1} = x_{i-1}\big), \end{aligned} \tag{2.26}$$

and

$$\begin{aligned} &\Pr\big(Z_i = 0 \mid \gamma_{i-1} = x_{i-1}, D_{i-1} = n_{i-1}\big) \\ &= 1 - \Pr\big(Z_i = 1 \mid \gamma_{i-1} = x_{i-1}, D_{i-1} = n_{i-1}\big). \end{aligned}$$

Consequently, Z_i only depends on γ_{i-1}, in the sense that

$$\begin{aligned} &\Pr\big(Z_i = 1 \mid \gamma_{i-1} = x_{i-1}\big) = \\ &\Pr\big(Z_i = 1 \mid \gamma_{i-1} = x_{i-1}, D_{i-1} = n_{i-1}, \ldots, D_1 = n_1, \gamma_0 = x_0\big). \end{aligned}$$

That is, the $\{ Z_0, Z_1, \ldots, Z_N \}$ sequence is dependent on an underlying $\{ \gamma_0, \gamma_1, \ldots, \gamma_N \}$ phase sequence. According to (2.14), the consecutive γ_i values form a Markov chain, since $\Pr\big(\gamma_i < x_i \mid \gamma_{i-1} = x_{i-1}\big) = \Pr\big(\gamma_i < x_i \mid \gamma_{i-1} = x_{i-1}, \ldots, \gamma_0 = x_0\big)$.
Consider the $G(x, y) = \Pr\big(\gamma_1 < y \mid \gamma_0 = x\big)$ and $g(x, y) = \frac{\mathrm{d}}{\mathrm{d}y} G(x, y)$ CDF and PDF of γ_i conditioned on γ_{i-1}. They can be calculated using (2.14), giving

$$G(x, y) = \Pr\big(\gamma_1 < y \mid \gamma_0 = x\big) = \sum_{n=0}^{\infty} F_n(x, y), \tag{2.27}$$

$$g(x, y) = \frac{\mathrm{d}}{\mathrm{d}y} \Pr\big(\gamma_1 < y \mid \gamma_0 = x\big) = \sum_{n=0}^{\infty} f_n(x, y). \tag{2.28}$$

The stationary distribution of the γ_i phase Markov chain satisfies

$$f(y) = \int_{x=0}^{\tau} f(x) g(x, y) \mathrm{d}x. \tag{2.29}$$

The solution of (2.29) is $f(y) = \frac{1}{\tau}$ for $y < \tau$ as presented in Appendix A.1.
Additionally consider the following: For τ being an integer multiple of Δ, $k \in \{ 1, \ldots, \tau/\Delta \}$ and $i \in \{ 1, \ldots, N \}$, let

$$C_i(k) = \begin{cases} 1 \text{ if } (k-1)\Delta < \gamma_i < k\Delta, \\ 0 \text{ otherwise,} \end{cases}$$

be a counting process that counts the number of phase occurrences in $\{\gamma_0, \gamma_1, \ldots, \gamma_N\}$ that fall into the kth Δ long interval of the possible phase values between 0 and τ. The total count for each of these Δ intervals is $\mathcal{G}_k = \sum_{i=1}^{N} C_i(k)$. For $N \to \infty$, the ratio of phase counts in each of these intervals is then $\lim_{N\to\infty,\Delta\to 0} \frac{\mathcal{G}_k}{N} = \lim_{N\to\infty} \Pr\big((k-1)\Delta < \gamma_N < k\Delta\big) = f((k-1)\Delta)\Delta + \sigma(\Delta)$, showing that the occurrence ratios of particular γ_n values are governed by the stationary phase distribution, where the error term $\sigma(\Delta)$ diminishes as $\lim_{\Delta\to 0} \sigma(\Delta)/\Delta = 0$.
Therefore, (due to the ergodicity of the γ_i Markov chain) as N tends to infinity, the number of samples in the $\{\gamma_0, \gamma_1, \ldots, \gamma_N\}$ phase sequence which falls into the $(x, x+\Delta)$ interval is proportional to $f(x)\Delta$. Using this, the ratio of accepted intervals can be written as:

$$\begin{aligned}\mathcal{S} &\triangleq \lim_{N\to\infty} \frac{S_N}{N} = \int_{x=0}^{\tau} \frac{1}{\tau} \Pr\big(D_i > m \mid \gamma_{i-1} = x\big)\, \mathrm{d}x \\ &= \int_{x=0}^{\tau} \frac{1}{\tau} \sum_{n=m+1}^{\infty} \mathrm{e}^{-\lambda(n\tau-\zeta-x)} \left(1 - \mathrm{e}^{-\lambda\tau}\right) \mathrm{d}x \\ &= \sum_{n=m+1}^{\infty} \frac{(\mathrm{e}^{\lambda\tau}-1)^2 \mathrm{e}^{-\lambda\left(n+1\right)\tau-\zeta)}}{\lambda\tau} \\ &= \frac{(\mathrm{e}^{\lambda\tau}-1)\mathrm{e}^{-\lambda\left((m+1)\tau-\zeta\right)}}{\lambda\tau}.\end{aligned} \tag{2.30}$$

The expected virtual count rate of Algorithm 1 can then be calculated as:

$$\begin{aligned}\lambda_{\mathrm{v}} = \mathcal{S} \cdot \lambda_{\mathrm{d}} &= \frac{(\mathrm{e}^{\lambda\tau}-1)\mathrm{e}^{-\lambda\left((m+1)\tau-\zeta\right)}}{\lambda\tau} \cdot \frac{\lambda}{1+\lambda\zeta} \\ &= \frac{\mathrm{e}^{-\lambda((m+1)\tau-\zeta)}\left(\mathrm{e}^{\lambda\tau}-1\right)}{\tau(\lambda\zeta+1)},\end{aligned} \tag{2.31}$$

where λ_{d} is the original rate with dead time, as obtained in (2.12). □

Let $\Theta = \lim_{\ell\to\infty} \Theta_\ell$ be the stationary number of leading clock edges between generating consecutive V_ℓ values. Theorem 2 defines its mean as $\mathbb{E}(\Theta) = 1/(\lambda_{\mathrm{v}}\tau)$. The expected time for generating a vDTD with Algorithm 1, T_Θ, can then be written as

$$\mathbb{E}(T_\Theta) = \tau \cdot \mathbb{E}(\Theta) = \frac{1}{\lambda_{\mathrm{v}}} = \frac{\tau(\lambda\zeta+1)}{\mathrm{e}^{-\lambda((m+1)\tau-\zeta)}\left(\mathrm{e}^{\lambda\tau}-1\right)}. \tag{2.32}$$

The vDTD sample generation rate computed according to Theorem 2 is depicted in Fig. 2.5.

2.3.2 Computation of general performance indices

Beyond the λ_{v} virtual count rate, there may be other secondary statistical performance indices of interest (e.g., the squared coefficient of variation). While for the general

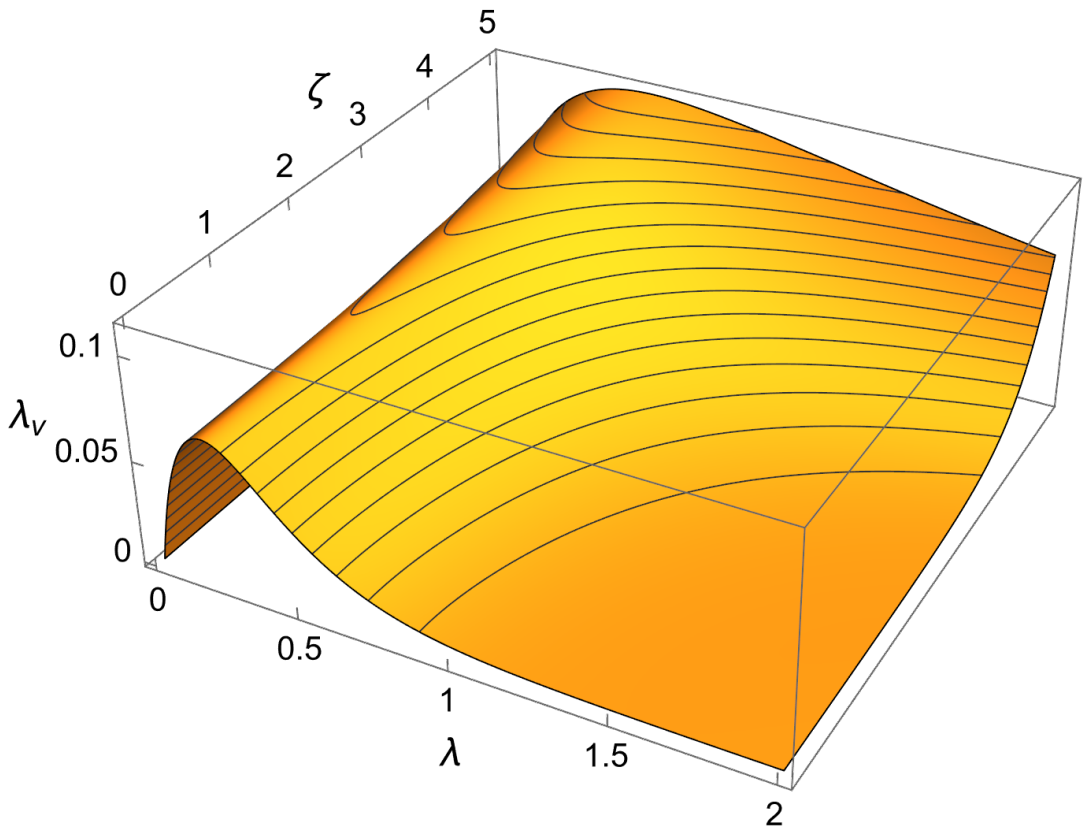

Figure 2.5: Virtual count rate λ_v as a function of input photon rate λ and dead time ζ, with fixed $\tau = 1, m = 5$ parameters.

case, a closed analytical formula similar to the case of calculating λ_v may not exist, a general analysis and estimation approach based on using an Erlang estimated clock is presented in Appendix A.3.

2.4 Simulation and evaluation

To further investigate and validate the correctness of the proposed overestimation method, I created multiple simulation cases with a custom-built Python program. With it, I validated the obtained analytical results against simulations of the performance indices. For sample interval generation, I utilized Python's built-in pseudorandom "random" library[2] to simulate photon emission times for particular λ and τ parameters. I also simulated the effect of a constant ζ dead time (emissions in the dead time period are not registered as detections). Then, I used these intervals

[2]Although pseudorandom number generators cannot provide truly random numbers, the output they produce is still suitable for initial investigations, as this output is expected to mimic the statistical properties of truly random sequences.

as the input for a Python function implementing Algorithm 1 to generate simulated vDTD distributions and calculate various statistics of the simulation results. Each simulation run consisted of 1 million consecutively generated intervals, while each data point is obtained by taking the mean of 20 independent simulation runs. In the figures, the standard deviation of the statistic is also denoted with a blue error bar based on the 20 samples, although this value is mostly too small for graphical visibility.

2.4.1 Correctness of simulation

First, I verified the validity of simulations using the theoretical lag-1 correlations calculated in (A.11). The dead time in the simulation had zero integer part ($k = 0$) and a fractional part δ varying between 0 and 0.9. The clock resolution was set to $\tau = 1$, and I swept the value of λ between 0 and 10. The results in Fig. 2.6 show excellent agreement between theory and simulations.

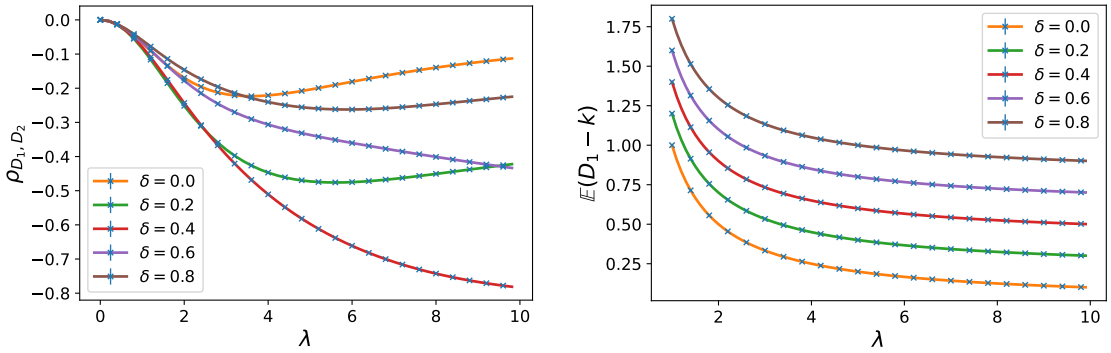

Figure 2.6: Comparison of theoretically calculated (solid lines) and simulated results (markers) for the correlation between successive DTDs (left) and the mean value of DTDs (right), as a function of λ and selected fractional dead times δ. The simulation uses $\tau = 1$, and the step size for λ is 0.1, but only every fourth data point is shown here for better visibility.

2.4.2 Virtual DTD generation rate

Simulation results support the validity of the theoretical model presented in Theorem 2 since the theoretical and simulation results align as expected. Fig. 2.7 shows two example simulation cases.

2.4.3 Performance loss

To demonstrate the performance cost of using Algorithm 1, I compared the output DTD and vDTD rates. Comparing λ_{d} and λ_{v} indicates that for low values of $\lambda\tau$

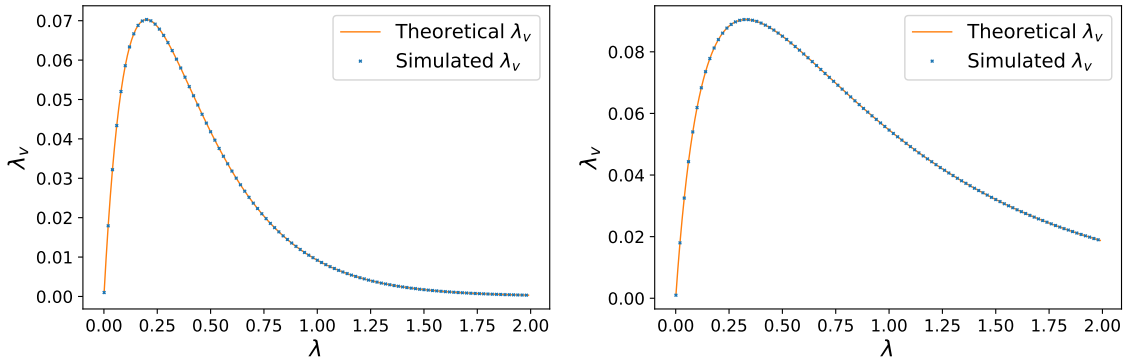

Figure 2.7: Theoretically derived and simulated results for the virtual count rate λ_v as a function of the λ input photon rate, for different dead times ($\zeta = 1.8, 4.2$, left to right) with $m = 5, \tau = 1$. The simulation step size for λ is 0.05, but only every second data point is shown here for better visibility.

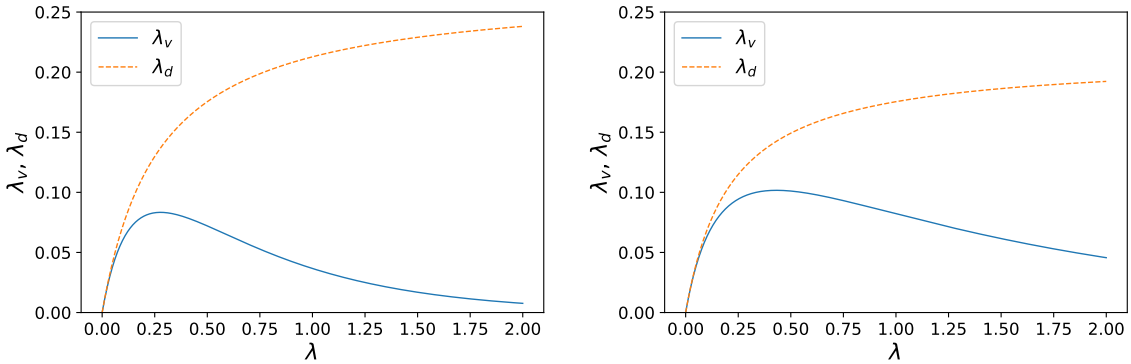

Figure 2.8: Comparison of achievable output rates at λ input photon rates for different dead times ($\zeta = 3.7, 4.7$, left to right) with (λ_v) and without (λ_d) using Algorithm 1 with $\tau = 1$, $m = 5$.

($\lambda\tau \ll 1$), the difference in output rates is not substantial, but with growing $\lambda\tau$ ($\lambda\tau \sim 1$), the performance cost of using Algorithm 1 becomes apparent as can be seen on Fig. 2.8.

We can further define the $\lambda_\mathrm{v}/\lambda_\mathrm{d}$ overestimation ratio to quantify this performance loss:

$$\frac{\lambda_\mathrm{v}}{\lambda_\mathrm{d}} = \frac{\mathrm{e}^{-\lambda\left((m+1)\tau-\zeta\right)}\left(\mathrm{e}^{\lambda\tau}-1\right)}{\tau(\lambda\zeta+1)} \cdot \frac{1+\lambda\zeta}{\lambda} = \frac{\left(\mathrm{e}^{\lambda\tau}-1\right)\mathrm{e}^{-\lambda\left((m+1)\tau-\zeta\right)}}{\lambda\tau}. \tag{2.33}$$

Eq. (2.33) indicates that the critical defining factor for performance loss is the difference $m\tau - \zeta$ (which I will call the *accuracy of overestimation*), corresponding to how much ζ is overestimated with $m\tau$. While $m\tau$ needs to be strictly greater than ζ for Algorithm 1 to provide uncorrelated vDTD output, it is beneficial to choose $m\tau$ as close to ζ as possible. This effect is further illustrated in Fig. 2.9.

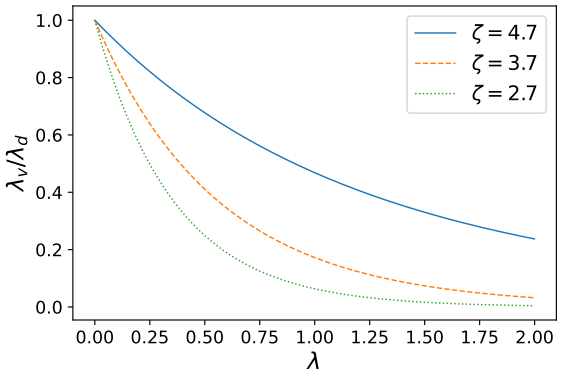

Figure 2.9: The performance cost ratio $\lambda_\mathrm{v}/\lambda_\mathrm{d}$ as a function of λ input photon rate for different dead times ($\zeta = 2.7, 3.7, 4.7$) and $\tau = 1, m = 5$.

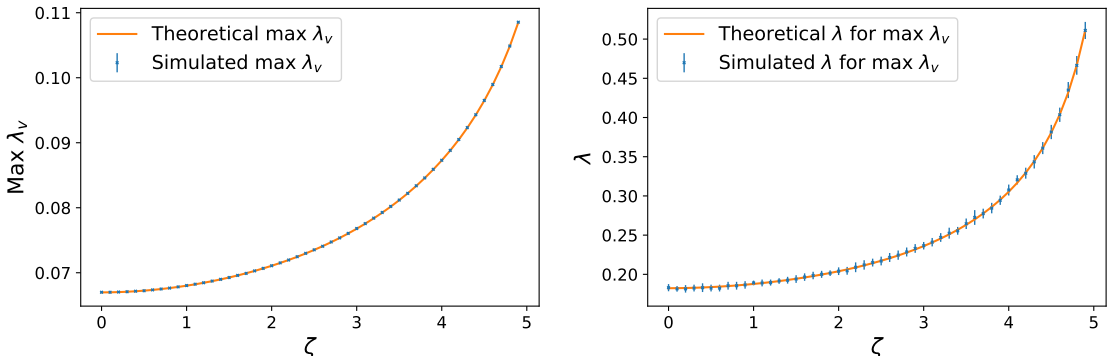

Figure 2.10: Maximally achievable virtual count rates (λ_v) and the corresponding input rates (λ) for different dead times (ζ) with fixed $m = 5$ and $\tau = 1$ parameters.

2.4.4 Maximum achievable output virtual count rate

When generating vDTDs with Algorithm 1, increasing the λ input photon rate beyond a certain point decreases the final virtual count rate as the probability of detections corresponding to smaller D_i values rises. Thus, finding the optimal input λ corresponding to the maximally achievable output λ_v is important.
Using (2.31) we can find this maximum by solving

$$\begin{aligned}\frac{\partial \lambda_\mathrm{v}}{\partial \lambda} &= \frac{\partial}{\partial \lambda}\frac{\mathrm{e}^{-\lambda((m+1)\tau-\zeta)}\left(\mathrm{e}^{\lambda\tau}-1\right)}{\tau(\lambda\zeta+1)} \\ &= \frac{\left(\mathrm{e}^{\lambda\tau}-1\right)\left[\zeta-(m+1)\tau\right]\mathrm{e}^{-\lambda((m+1)\tau-\zeta)}}{\tau(\lambda\zeta+1)} \\ &- \frac{\zeta\left(\mathrm{e}^{\lambda\tau}-1\right)\mathrm{e}^{-\lambda((m+1)\tau-\zeta)}}{\tau(\lambda\zeta+1)^2} + \frac{\mathrm{e}^{\lambda\tau-\lambda((m+1)\tau-\zeta)}}{\lambda\zeta+1} = 0\end{aligned} \tag{2.34}$$

for λ. Unfortunately, this equation has no algebraic solution but can still be solved numerically. Solutions for the example parameter set are compared to simulation results in Fig. 2.10.

The accuracy of the overestimation $(m\tau - \zeta)$ has a critical effect on maximum achievable rates as well. This reinforces the importance of choosing $m\tau$ as close to ζ as possible.

2.4.5 Entropy of the output counts

Due to Algorithm 1, the vDTDs are independent and identically geometrically distributed with

$$\Pr(V = v) = p_v = p(1-p)^v,\ v \in \mathbb{Z}^+ \tag{2.35}$$

probabilities where $p = 1 - \mathrm{e}^{-\lambda\tau}$. Consequently, the min-entropy of a vDTD is

$$H_\infty(V) = \min_v(-\log_2 p_v) = -\log_2\left(1 - \mathrm{e}^{-\lambda\tau}\right) \tag{2.36}$$

and its (Shannon) entropy is

$$\begin{aligned} H(V) &= -\sum_v p_v \log_2 p_v = \frac{-(1-p)\log_2(1-p) - p\log_2 p}{p} \\ &= \frac{\lambda\tau \cdot \log_2(\mathrm{e}) \cdot \mathrm{e}^{-\lambda\tau} - \log_2(1-\mathrm{e}^{-\lambda\tau}) \cdot (1-\mathrm{e}^{-\lambda\tau})}{1-\mathrm{e}^{-\lambda\tau}}. \end{aligned} \tag{2.37}$$

The min-entropy of a random variable provides the upper bound of uniform bits that can be extracted from the variable [95] and can never exceed its Shannon entropy, making it a more efficient measure when assessing random number generators. The other main factor determining the achievable raw entropy generation speed is the rate at which measurement samples are obtained. When using Algorithm 1 this rate is the λ_v virtual count rate, as it determines the speed at which Algorithm 1 generates vDTDs. The (min-)entropy rates, defined as the (min-)entropy generated per unit time, are the products of the (min-)entropy per random variable and the rate at which random variables are generated. Their values can be calculated as $h(V) = \lambda_\mathrm{v} \cdot H(V)$ and $h_\infty(V) = \lambda_\mathrm{v} \cdot H_\infty(V)$, respectively.

2.4.6 Handling non-constant dead time

The assumption of constant non-extendable dead time ζ may not be applicable to some real systems. Here, I also consider cases when ζ is a random variable to further evaluate the limits of the practical applicability of the presented overestimation scheme.

Monotonicity of the λ_v virtual count rate

I first show that the λ_v virtual count rate is monotonic in ζ, then provide limits for λ_v assuming finite-support dead time distributions. The λ_v virtual count rate is monotonic in ζ, since

$$\begin{aligned}\frac{\partial \lambda_v}{\partial \zeta} &= \frac{\partial}{\partial \zeta}\frac{e^{-\lambda((m+1)\tau-\zeta)}\left(e^{\lambda\tau}-1\right)}{\tau(\lambda\zeta+1)} \\ &= \frac{\lambda\zeta^2 e^{-\lambda((m+1)\tau-\zeta)}\left(e^{\lambda\tau}-1\right)}{\tau(\lambda\zeta+1)^2} > 0,\end{aligned} \tag{2.38}$$

because $\lambda > 0$, $\zeta \geq 0$, and $\tau > 0$ by definition, which also makes $e^{\lambda\tau} > 1$, therefore (2.38) holds true for all valid ζ.

Bounded ζ

For the case of finite-support ζ distributions, we can use the upper bound of the distribution to set m adequately. In contrast, due to the monotonicity in ζ, we can use the lower bound of ζ to calculate the worst-case performance characteristics of Algorithm 1 for the chosen m. More precisely, given an upper bound ζ_U and lower bound ζ_L for ζ, we can substitute $\zeta = \zeta_L, m = \lfloor \zeta_U/\tau \rfloor + 1$ into previous formulae to get the worst-case result (with respect to the achievable λ_v). Since the m overestimation parameter is set according to ζ_U, and λ_v is maximal when $m\tau - \zeta$ is minimal, the constant $\zeta = \zeta_U$ distribution corresponds to the best case scenario. Substituting these into (2.24), we obtain

$$\begin{aligned}&e^{-\lambda\left[\left(\left\lfloor\frac{\zeta_U}{\tau}\right\rfloor+2\right)\tau-\zeta_L\right]}\left(e^{\lambda\tau}-1\right)\cdot\frac{1}{\tau(\lambda\zeta_L+1)} \leq \lambda_v \quad \text{and} \\ &\lambda_v \leq e^{-\lambda\left[\left(\left\lfloor\frac{\zeta_U}{\tau}\right\rfloor+2\right)\tau-\zeta_U\right]}\left(e^{\lambda\tau}-1\right)\cdot\frac{1}{\tau(\lambda\zeta_U+1)}.\end{aligned} \tag{2.39}$$

This way, even if the exact value or distribution of ζ is unknown, a lower and upper estimate for achievable λ_v virtual count rates is still obtainable.

Unbounded dead time distributions

For a fixed value of m, a particular sample from an arbitrary ζ distribution can fall into two categories:

$$A_1 : \ \zeta \leq m\tau,$$
$$A_2 : \ \zeta > m\tau,$$

where A_1 and A_2 are mutually exclusive and complete. Due to the law of total probability, the stationary distribution of the V vDTDs can be written as

$$\begin{aligned}\Pr(V=v) = \Pr\big(V=v \mid \zeta \leq m\tau\big) \cdot \Pr(\zeta \leq m\tau) \\ + \Pr\big(V=v \mid \zeta > m\tau\big) \cdot \Pr(\zeta > m\tau),\end{aligned} \tag{2.40}$$

where the first part of the sum corresponds to A_1 and the second part to A_2. In the case of A_1, the corresponding distribution of V is the same as in Sec. 2.3 since $\zeta \leq m\tau$ and in this case, $\Pr\big(V=v \mid \zeta \leq m\tau\big)$ is independent of ζ and equals to (2.22).

In the case of A_2, $\Pr\big(V=v \mid \zeta > m\tau\big)$ is no longer independent of ζ; therefore, V is no longer ensured to be uncorrelated and may show unwanted correlations. However, the probability of potentially correlated samples is $\Pr(\zeta > m\tau)$ and can be adjusted by the choice of m. Larger m values result in lower vDTD generation rate λ_v but lower probability of correlated samples; the opposite holds for small m values. An appropriate trade-off can be set by the proper choice of m.

Not restricting the distribution of ζ (allowing non-constant and non-parallelizable dead time models, too), for the general $\Pr(V=v)$ case, the presented overestimation method will still work as intended for cases in A_1. Note that according to (2.40), the actual occurrence probability of A_1 ($\Pr(\zeta \leq m\tau)$) is still governed by F_ζ and may have various dependencies.

2.5 Experimental results

In addition to the simulation results, I also tested Algorithm 1 with the physical setup available in our laboratory. Here, I present the physical measurement system, as well as experimental results regarding the operation of Algorithm 1.

2.5.1 Physical measurement system

The setup I used to validate all my results experimentally is the same as the one presented in Ref. [113]. The main physical properties of this ensemble are the following:

The photon source is a semiconductor laser (Thorlabs LP520-SF15) emitting around a central wavelength of 519.9 nm, managed by a dedicated driver circuit. The optical fiber cables (Thorlabs 460HP) are designed to aid single traverse mode propagation in the wavelength band of interest and have a core diameter of 2.5 µm. The light coming from the laser source is attenuated first with a voltage-controlled variable optical attenuator (VOA, Thorlabs V450F), then by using the 1% output port of an optical

splitter (Thorlabs TW560R1F1), and then finally another voltage-controlled variable optical attenuator (Thorlabs V450F) before reaching the single photon detector (PMT, PicoQuant PMA-175 NANO). The detector detects incoming photons with approximately 21% quantum efficiency at 520nm and outputs a voltage pulse with 180 ps FWHM transit time spread, which is then time-tagged by a time-to-digital converter (TDC, PicoQuant TimeHarp 260) with a base resolution of $\tau = 250$ ps operating with a continuous (non-restartable) clock signal. Processing the resulting time tags is software-based, running on a computer directly accessing the TDC card. A diagram of the main components setup can be seen in Fig. 2.11, while Fig. 2.12 shows the actual physical hardware without the protective covers.

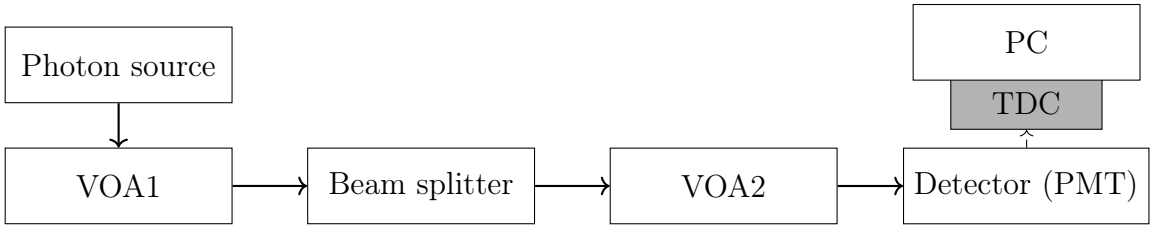

Figure 2.11: Diagram of the experimental measurement setup. VOA: variable optical attenuator; PMT: photomultiplier tube; TDC: time-to-digital converter card. (Beam splitter functions as an additional 20 dB attenuator)

The variable optical attenuators are controlled by a designated microcontroller board, while the splitter provides a fixed 20 dB of attenuation (The other 99% port of the splitter can be used for monitoring purposes but remains unused in my experiments.). The purpose of the variable attenuators is twofold. On one hand, they can be used for tuning the photon rate, allowing for the investigation of a wide range of parameter values. On the other hand, they are responsible for protecting the single-photon detector from overly high input power, as the maximum allowed output count rate of the detector is only $5 \cdot 10^6$ counts per second (cps).

To minimize outside noise the hardware components are covered by a black box, as well as a black cloth in the case of the attenuators and the detector. Additionally, the PMT has quite advantageous noise characteristics, with negligible afterpulsing probability ($\sim 0\%$, as the device is more sensitive to wavelengths towards the blue end of the visible spectrum) and a specified dark count rate of $<$50 cps at room temperature. The average experimentally measured noise level was $<$10 cps. This is significantly lower than any input photon rates of practical interest for this particular experimental setup.

The PMT remains sensitive after detection and thus has no direct dead time, only the finite width of the voltage pulse denoting a detection, meaning that detections happening sooner than 1.5 ns cannot be discriminated. The main contributor

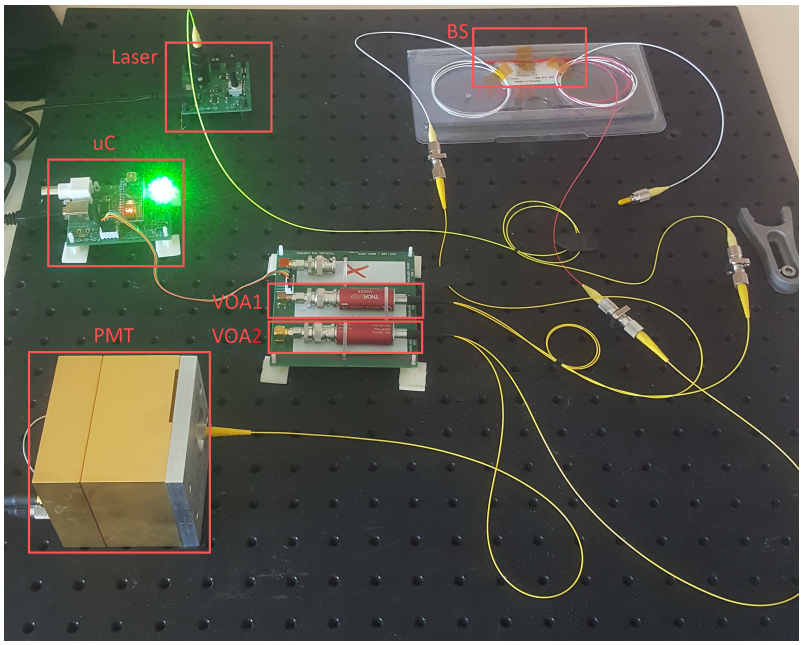

Figure 2.12: Photo of the physical setup from Ref. [114]. uC: microcontroller controlling VOAs, BS: beam splitter, VOA: variable optical attenuator, PMT: photomultiplier tube. Photons travel along the Laser-VOA1-BS-VOA2-PMT optical path.

to the dead time of the system is the time-tagger card, with an average value around $\zeta \approx 2\,\text{ns} = 8\tau$ based on its datasheet. Prior measurements and histograms of interarrival times, however, suggest that the actual dead time might exceed 2 ns, and its value is not constant.

Knowing the parameters of the physical setup, one can argue that the dead time can be considered non-extendable in this particular case: Let us assume that the detection system is paralyzable (dead time is extendable). If the mean time between photon arrivals is significantly larger than the dead time ζ, such that:

$$\zeta \ll \frac{1}{\lambda} \iff \lambda\zeta \ll 1, \tag{2.41}$$

the probability of any new arrivals and thus extending the dead time duration is small. Therefore, extension is very unlikely and the detection system behaves approximately as non-extendable. From the operating parameters presented earlier, it can be seen that (2.41) holds for the physical setup even when operating with maximum allowable photon detection rates.

2.5.2 Investigating the output interval distribution

To experimentally check the resulting V_ℓ vDTD distribution I collected measurement data of $2 \cdot 10^9$ observed photon arrival times with a mean detection rate of

1.05 ± 0.01 Mcps. Rescaling after accounting for the typical dead time of the system according to (2.11) results in an input photon rate of $\lambda = 1.052$ Mcps.
Since the measurement setup is limited to a maximum detection rate of 5 Mcps with a fixed τ time resolution of 250 ps, I also created time-binned versions of the original, unbinned measurement data to investigate possible $\lambda\tau$ statistics beyond the setup's physical operational limits. To do so, I used data recorded with the device's own τ time resolution and created lower resolution versions of the same experiment as if I used a longer, $\tau' = K_\mathrm{b} \cdot \tau$ clock period, where K_b is a positive integer. The binning method is presented in Algorithm 2.

Algorithm 2 Binning algorithm

Require: $\mathbb{D}$ ▷ Original DTD samples
Require: K_b ▷ Integer
$\mathbb{D}' = [\,]$ ▷ Array of binned DTDs
$c_1 := 0$ ▷ Carry
for $i = 1$ to length($\mathbb{D}$) **do**
    get $D_i = \mathbb{D}[i]$
    $\mathbb{D}'[i] = \lfloor (D_i + c_i)/K_\mathrm{b} \rfloor$
    $c_{i+1} = D_i + c_i \pmod{K_\mathrm{b}}$
end for

I obtained additional *binned datasets* corresponding to $K_\mathrm{b} = 2,\ 5,\ 10,\ 100,\ 1000$. Note that the absolute value of ζ is fixed, even when rescaling the $\lambda\tau$ product with the binning algorithm, and thus, ζ remains an unchanging physical parameter for all measurement cases. Then, I applied Algorithm 1 to the unbinned and binned raw datasets. The output of Algorithm 1 is referred to as *overestimated data*. For the unbinned data ($K_\mathrm{b} = 1$), I set $m = 1000$ as a safe overestimation parameter,[3] and $m' = 500,\ 200,\ 100,\ 10,\ 1$ for the binned data with $K_\mathrm{b} = 2,\ 5,\ 10,\ 100,\ 1000$, respectively, following the rule $m' = 1000/K_\mathrm{b}$.[4]

[3] Examining the measurement data, I concluded that $\zeta < 1000\tau$ with high enough certainty that this choice of m can be considered safe, faithfully overestimating the dead time.

[4] The binning algorithm rescales the necessary overestimation parameter by $1/K_\mathrm{b}$, as the dead time of the underlying process is unchanged. If $\zeta < m\tau$, then $\zeta < (m/K_\mathrm{b}) \cdot (K_\mathrm{b} \cdot \tau)$ holds trivially. The choice of $m' = m/K_\mathrm{b}$ yields a comparable dataset to the unbinned set overestimated by m; using the original overestimation parameter for the binned sequence would result in a greatly reduced λ_v.

I evaluated the *raw* and *overestimated* (both unbinned and binned) datasets in the following ways:

1. By calculating the autocorrelation of (v)DTD sequences.
2. By counting single (v)DTD occurrences. As the distribution of values (the histogram) is expected to be geometrically distributed, I fit it to the expected form. I then calculated the goodness of fit and checked the fitting parameters.
3. By counting the relative frequencies of consecutive (v)DTDs' value pairs. Measured pair statistics are compared to the expected value of the ideal, independent case (calculated as the product of relative frequencies of single (v)DTDs) via hypothesis testing.

Autocorrelation of (v)DTD sequences

Autocorrelation coefficients of the datasets are denoted as a_1 and a_1^{o} for raw and overestimated data, respectively. The unbinned raw dataset shows correlation coefficients in the order of 10^{-5}. The half-width of the 95% confidence interval for zero correlation is

$$\frac{\sqrt{2}\cdot \mathrm{Erf}^{-1}(0.95)}{\sqrt{2\cdot 10^9}} = \frac{1.96}{\sqrt{2\cdot 10^9}} = 4.38\cdot 10^{-5}$$

for $2\cdot 10^9$ samples, where $\mathrm{Erf}^{-1}(\cdot)$ is the inverse error function. Obtaining such small correlation coefficients is expected even without overestimation when $\lambda\tau \ll 1$ (recall that correlations become noticeable as the product increases). Table 2.1 lists the lag-1 coefficients of raw and overestimated datasets. The only coefficient exceeding 10^{-4} in absolute value is the lag-1 coefficient for the dataset with the largest $\lambda\tau$, using $K_{\mathrm{b}} = 1000$, which shows a significant and sudden increase, leaping above 10^{-3} in magnitude.

After overestimation, lag-1 coefficients remained in the order of 10^{-5}, within the 95% confidence interval for zero correlation (even without considering the slight growth of the confidence interval due to the reduced number of samples in the overestimated datasets).[5] All of the overestimated sequences show lower magnitude autocorrelation coefficients than their unprocessed counterparts. The difference is most notable for the sequence with binning parameter 1000, which was originally heavily correlated. When overestimated, the sequence performs significantly better. Note that sequences have similar values after being passed through the algorithm. This is expected since

[5] E.g., for the shortest dataset ($K_{\mathrm{b}} = 1000, m' = 1$) with $1.37\cdot 10^9$ samples, the magnitude of the 95% confidence interval increases to $\sqrt{2}\cdot \mathrm{Erf}^{-1}(0.95)/\sqrt{1.37\cdot 10^9} = 1.96/\sqrt{1.37\cdot 10^9} = 5.29\cdot 10^{-5}$.

all of them are discretized from the same realization of the underlying PPP, and all use the same overestimation parameter after adjusting for dead time, $m' \cdot K_\mathrm{b}$.

Table 2.1: Lag-1 autocorrelation coefficients of raw (a_1) and overestimated (a_1^o) datasets. Overestimation successfully reduced the absolute values of correlation coefficients for all data.

K_b / m'	$\lambda\tau'$	a_1	a_1^o
1 / 1000	$2.630 \cdot 10^{-4}$	$4.324 \cdot 10^{-5}$	$-7.811 \cdot 10^{-6}$
2 / 500	$5.261 \cdot 10^{-4}$	$4.322 \cdot 10^{-5}$	$-8.175 \cdot 10^{-6}$
5 / 200	$1.315 \cdot 10^{-3}$	$4.311 \cdot 10^{-5}$	$-7.692 \cdot 10^{-6}$
10 / 100	$2.630 \cdot 10^{-3}$	$4.273 \cdot 10^{-5}$	$-1.109 \cdot 10^{-5}$
100 / 10	$2.630 \cdot 10^{-2}$	$-1.474 \cdot 10^{-5}$	$-1.233 \cdot 10^{-5}$
1000 / 1	$2.630 \cdot 10^{-1}$	$-5.737 \cdot 10^{-3}$	$-1.987 \cdot 10^{-5}$

Frequencies of (v)DTD values

Histograms show an even more noticeable contrast between the raw and overestimated cases. I fit the function $y = A \cdot \mathrm{e}^{-Ax} + C$ to the histogram data using the least squares method.[6] Ideally, fitting would yield $A = \lambda\tau'$ and $C = 0$. Note that this is a discretized version of the exponential probability density function $f_T(t) = \chi_{\{t\geq 0\}}\lambda \cdot \mathrm{e}^{-\lambda t}$.[7] The histograms and results of the fitting are shown in Fig. 2.13. Histograms show deviations from a geometric distribution for the raw datasets, noticeable even by visual inspection, while overestimated datasets do not. The fitting error statistics of overestimated datasets are at least 3 orders of magnitude better compared to their raw counterparts, both in the case of *mean square error*s (MSEs) and *coefficient of determination* parameters (R^2; perfect fit is $R^2 = 1$). The resulting A parameters for the overestimated data are also in agreement with the expected $\lambda\tau'$ values,[8] although slightly larger. This is most probably because the expected $\lambda\tau'$ values were calculated

[6] I utilized the Scipy python library's "curve_fit" method with initial guiding guesses determined by the expected $\lambda\tau'$ parameter, and 10^5 maximum evaluations.

[7] As shown in (2.35) and Ref. [110], sampling exponentially distributed time intervals with parameter λ (using a restartable clock with resolution τ and no dead time) yields geometrically distributed samples. Thus, an equivalent exponential fit is also a valid substitute for this geometric fit. The additional C parameter is introduced because I only considered data in the histograms corresponding to the first part of the distribution that fits into the predetermined amount of histogram bins.

[8] For the $K_b = 100$ and $K_b = 1000$ cases, bigger deviation of the fit parameters are expected due to smaller sample sizes (since the number of histogram bins was also scaled with K_b for comparability of results) and higher impact of the C fitting parameter.

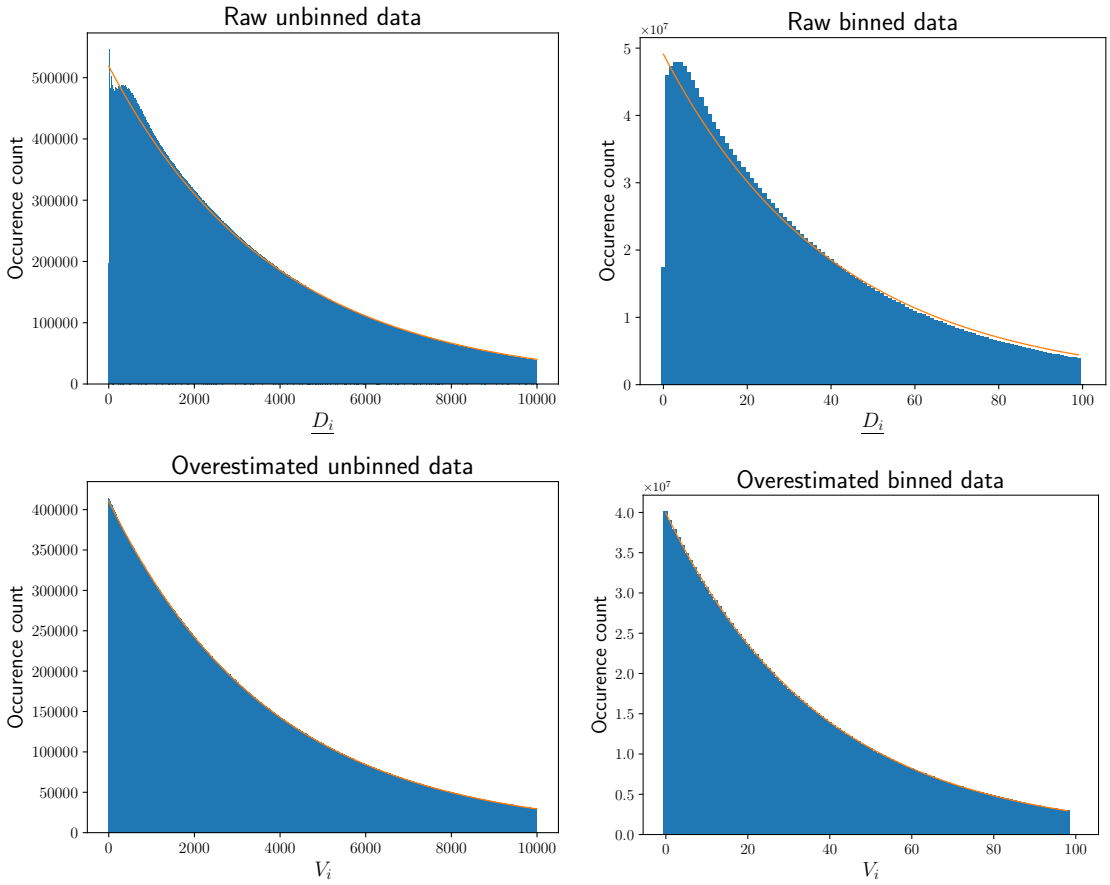

Figure 2.13: Histograms and the results of curve fitting for measurement data before and after overestimation. Due to the effect of dead time, I shifted the histogram left before fitting, not including the originally empty bins for smaller D_i values. I denote these shifted values by $\underline{D_i}$. Figures on the left correspond to the original measurement data (unbinned), while figures on the right correspond to a binned case with $K_\mathrm{b} = 100$. The top row shows histograms of the unprocessed data, while the bottom row shows the resulting histograms after using Algorithm 1. The orange lines indicate the results of the attempted curve fitting.

with the spreadsheet dead time value of 2 ns, but in reality, the actual dead-time-like imperfections of the measurement setup caused a bigger reduction of the effective rate than what the constant $\zeta = 2\,\mathrm{ns}$ correction accounted for. The fitting results are summarised in Tables 2.2 and 2.3.

Hypotheses testing

According to Sec. 2.2.2 and (2.20), the original measurement datasets are expected to show correlations for consecutive count statistics where D_i is smaller. To investigate this, I devised a hypothesis testing scheme where I counted the relative frequency of pairs consisting of consecutive (v)DTDs and compared them to the relative frequency of single (v)DTDs calculated from the measurements.

Table 2.2: A parameters of curve fitting before and after overestimation

Data	Raw A	Overestimated A	Expected ($\lambda\tau'$)
$K_\mathrm{b} = 1$	$2.578 \cdot 10^{-4}$	$2.638 \cdot 10^{-4}$	$2.630 \cdot 10^{-4}$
$K_\mathrm{b} = 2$	$5.154 \cdot 10^{-4}$	$5.276 \cdot 10^{-4}$	$5.261 \cdot 10^{-4}$
$K_\mathrm{b} = 5$	$1.285 \cdot 10^{-3}$	$1.319 \cdot 10^{-3}$	$1.315 \cdot 10^{-3}$
$K_\mathrm{b} = 10$	$2.553 \cdot 10^{-3}$	$2.636 \cdot 10^{-3}$	$2.630 \cdot 10^{-3}$
$K_\mathrm{b} = 10^2$	$2.440 \cdot 10^{-2}$	$2.609 \cdot 10^{-2}$	$2.630 \cdot 10^{-2}$
$K_\mathrm{b} = 10^3$	$1.751 \cdot 10^{-1}$	$2.396 \cdot 10^{-1}$	$2.630 \cdot 10^{-1}$

Table 2.3: MSE and $1 - R^2$ values of curve fitting before and after overestimation

Data	Raw		Overestimated	
	MSE	$1 - R^2$	MSE	$1 - R^2$
$K_\mathrm{b} = 1$	$5.445 \cdot 10^{-11}$	$1.242 \cdot 10^{-2}$	$9.242 \cdot 10^{-14}$	$2.053 \cdot 10^{-5}$
$K_\mathrm{b} = 2$	$2.278 \cdot 10^{-10}$	$1.299 \cdot 10^{-2}$	$2.447 \cdot 10^{-13}$	$1.359 \cdot 10^{-5}$
$K_\mathrm{b} = 5$	$1.780 \cdot 10^{-9}$	$1.624 \cdot 10^{-2}$	$1.111 \cdot 10^{-12}$	$9.866 \cdot 10^{-6}$
$K_\mathrm{b} = 10$	$1.164 \cdot 10^{-8}$	$2.658 \cdot 10^{-2}$	$3.814 \cdot 10^{-12}$	$8.472 \cdot 10^{-6}$
$K_\mathrm{b} = 10^2$	$2.861 \cdot 10^{-6}$	$6.683 \cdot 10^{-2}$	$6.545 \cdot 10^{-10}$	$1.457 \cdot 10^{-5}$
$K_\mathrm{b} = 10^3$	$9.155 \cdot 10^{-4}$	$2.959 \cdot 10^{-1}$	$2.508 \cdot 10^{-6}$	$5.848 \cdot 10^{-7}$

If the individual (v)DTDs are independent, then the joint probabilities satisfy

$$\begin{aligned} \Pr\left(D_i = k, D_{i+1} = \ell\right) &= \Pr\left(D_i = k\right) \cdot \Pr\left(D_{i+1} = \ell\right) \text{ and} \\ \Pr\left(V_i = k, V_{i+1} = \ell\right) &= \Pr\left(V_i = k\right) \cdot \Pr\left(V_{i+1} = \ell\right). \end{aligned} \tag{2.42}$$

This can be used to define a hypothesis test, where the $\mathcal{H}_0$ null hypothesis is that the tested data is from an ideal binomial trial, with expected individual success probability given by (2.42). I then gather evidence trying to refute $\mathcal{H}_0$ from the measurement results, where successful rejection of $\mathcal{H}_0$ (the calculated p-value of the test is small) most likely means that the tested dataset does not follow the ideal geometric goal distribution.

I applied this binomial statistical test on each of the $\{D_i = k, D_{i+1} = \ell\}$ pair statistics for $k, \ell \in \{0, 1, \ldots, 19\}$. This way, there are 400 result p-values for each of the 400 possible pairings per dataset. Since from these multiple test results I'm looking to find the most outlying cases that may facilitate the rejection of $\mathcal{H}_0$, I use the Bonferroni correction [115] to account for the multiple comparisons problem that occurs when one considers a set of statistical inferences simultaneously [116]. This way, for a target *comprehensive* significance level per dataset of 0.01, the corrected *individual* significance levels for each of the 400 tests in the group are $0.01/400 = 2.5 \times 10^{-5}$. If any of the p-values are lower than the *individual* significance level, then the whole dataset fails with the *comprehensive* significance level.

The results of the statistical tests show a clear contrast between the raw and the overestimated data. The raw data scoring minimum p-values of 1.6×10^{-5} without binning ($K_\mathrm{b} = 1$) and 5.9×10^{-7}, 3.4×10^{-13}, 9.2×10^{-31}, $0, 0$ with binned raw datasets (for $K_\mathrm{b} = 2, 5, 10, 100, 1000$), which are orders of magnitude below the individual significance level. The minima of overestimated values range from 6×10^{-4} (with $K_\mathrm{b} = 100$) to 0.01 (with $K_\mathrm{b} = 1000$), which, unlike results from the raw data, are all above the individual significance level for rejection.

Further measurement results

The previously presented measurement results signified that the overestimation algorithm can transform distorted distributions into distributions very close to exponential/geometric. Additional datasets measured with detected photon rates of ∼400, ∼600, and ∼800 kcps were also evaluated with the previously presented methodology, yielding similar results, emphasizing the gains.

I also calculated the experimental ratio of measured input count rates to the virtual count rates achieved by Algorithm 1. Note that I only have measurement data available corresponding to low values ($\sim 10^{-4}$) of $\lambda\tau$, but the experimental results

all stay within the bounds given by (2.39), using $\zeta_{\mathrm{L}} = 10\tau$ and $\zeta_{\mathrm{U}} = 999\tau$. The experimental output/input rates of Algorithm 1 range from 0.774 (for $\sim 1\,\mathrm{Mcps}$ input rate) to 0.906 (for $\sim 400\,\mathrm{kcps}$ input rate), which is a tolerable performance loss for eliminating the correlations within the generated DTD series.

2.5.3 Effect on generated bits

To experimentally investigate a simple example of the practical applicability of Algorithm 1, I paired it with the simple bit generation method introduced in Sec. 2.2.3 and tested the quality of the output bits. For this, I used eight measurement datasets with differing $\lambda\tau$ values, indexed alphabetically as $\mathbb{D}_{\mathrm{A}}$ to $\mathbb{D}_{\mathrm{H}}$. Due to the limitations of the physical hardware, to achieve higher $\lambda\tau$ values, the previously presented binning algorithm was used. Therefore, datasets $\mathbb{D}_{\mathrm{E}}$ to $\mathbb{D}_{\mathrm{H}}$ are binned versions of dataset $\mathbb{D}_{\mathrm{D}}$. I chose multiple m overestimation parameters contained in the set $\mathbb{M}_{\mathrm{ID}}$ (ID going from A to H alphabetically) for each dataset, with the goal of having m values that are too low, safely high, and in between. For example, the $m = 10$ choice for the datasets measured using the $\tau = 250\,\mathrm{ps}$ original resolution corresponds to an overestimation of $m\tau = 2.5\,\mathrm{ns}$ close to the datasheet value of $\zeta = 2\,\mathrm{ns}$. Similar to the previous measurement case, binning also affects the choice of m here. To differentiate between datasets created using overestimation with parameter m, the notations A_m to H_m are introduced (A_0 to H_0 denoting sequences generated without overestimation). Table 2.4 contains the relevant measurement parameters of the datasets.

Table 2.4: Datasets and corresponding parameters

$\mathbb{D}_{\mathrm{ID}}$	λ [cps]	τ [ns]	$\lambda\tau$	K_{b}	$\mathbb{M}_{\mathrm{ID}}$
$\mathbb{D}_{\mathrm{A}}$	$5.080 \cdot 10^5$	0.25	0.000 127	1	$\{ 10, 50, 100, 500 \}$
$\mathbb{D}_{\mathrm{B}}$	$1.238 \cdot 10^6$	0.25	0.000 310	1	$\{ 10, 100, 500 \}$
$\mathbb{D}_{\mathrm{C}}$	$2.199 \cdot 10^6$	0.25	0.000 550	1	$\{ 10, 100, 500 \}$
$\mathbb{D}_{\mathrm{D}}$	$4.111 \cdot 10^6$	0.25	0.001 028	1	$\{ 10, 100, 500, 1000 \}$
$\mathbb{D}_{\mathrm{E}}$	$4.111 \cdot 10^6$	1.25	0.005 139	5	$\{ 2, 10, 100 \}$
$\mathbb{D}_{\mathrm{F}}$	$4.111 \cdot 10^6$	12.5	0.051 387	50	$\{ 2, 4, 20 \}$
$\mathbb{D}_{\mathrm{G}}$	$4.111 \cdot 10^6$	125	0.513 865	500	$\{ 1, 4 \}$
$\mathbb{D}_{\mathrm{H}}$	$4.111 \cdot 10^6$	300	1.233 277	1200	$\{ 1, 2, 3, 4, 5, 10 \}$

In the following, I present an overview of relevant experimental results obtained with this measurement setup. For the interested reader, Ref. [117] contains a more

in-depth presentation and discussion of the following results.

Distribution of bits and correlation

According to Sec. 2.2.3, the output bits created from DTDs of the original raw measurement cases (without using Algorithm 1) are not independent and should show deviation from the ideal uniform case in correlation between successive bits and therefore the distribution of distinct bit n-tuples. Following expectations, the lag-1 correlations in the raw datasets increase with increasing $\lambda\tau$, with values of -0.00003, -0.00001, 0.00019, 0.00126, 0.00127, 0.0014, 0.01, 0.04 for cases from A_0 to H_0, showing noticeable differences from the ideal zero correlation case for cases after D_0. In contrast, correlation coefficients of datasets overestimated with the safely high m values all stay within their corresponding 95% confidence interval for zero correlation.[9]

Bit triplet probabilities show a similar pattern for the raw datasets, with growing maximum deviations of 0.00038, 0.00038, 0.00044, 0.00461, 0.01285 for cases from D_0 to H_0 (A_0 to C_0 show no notable deviation), from the ideally expected 0.125 uniform probability as $\lambda\tau$ grows. In contrast, safely overestimated datasets show no notable deviation for any of the cases.

Statistical testing

I also tested the output bit sequences with the statistical test suites introduced in Sec. 1.5.2. I used the following parameters for the test suites:

- NIST STS: default parameters, 1024 teststreams.
- Dieharder: default parameters.
- TestU01: Alphabit and Rabbit batteries with default parameters.
- ENT: default parameters, run in both bit and byte mode.

Unfortunately, due to limited data storage capacities, some tested sequences could not fully satisfy the data requirements of some test cases. Thus, in cases with shorter output bit sequences, the NIST STS and TestU01 suites could not be completed successfully, while for some Dieharder tests, the data file was rewound several times, yielding skewed results. Nonetheless, even after considering this limitation, I observed

[9]Bit sequence length can vary significantly between the evaluated cases since generated bit sequence length is a function of both $\lambda\tau$ and the overestimation parameter m.

clear differences between the results corresponding to sequences generated from raw and overestimated datasets.

Overestimated sequences are expected to pass the statistical tests if the chosen m overestimation parameter is large enough to majorate the dead time. I found that for each of the $\mathbb{D}$ measurement cases, sequences with the largest corresponding m parameters passed the applicable statistical tests[10]. For the raw datasets, I found that cases with lower $\lambda\tau$ values are still passing the suites successfully (as argued in [113]), while from $\mathbb{D}_{\mathrm{C}}$, all of the sequences corresponding to raw unprocessed data fail. The primary failing trials for each of the used suites were tests investigating "runs of bits". This is in line with observations of Sec. 2.2.3, as "runs of bits" of bits tests calculate statistics of uninterrupted sequences of identical bits (which are expected to be skewed in case of correlation between consecutive bits).

Interval and bit generation ratios

To quantify the change in generated bits for the cases using the overestimated datasets, I define the bit retention efficiency η_{b} as

$$\eta_{\mathrm{b}} = \frac{\text{bits generated from } \mathbb{D} \text{ using overestimation}}{\text{bits generated from } \mathbb{D} \text{ by the raw method}}. \tag{2.43}$$

This is a readily available figure of merit that helps to quantify losses due to the overestimation algorithm. While the bit retention efficiency quantifies the change in generated bits, the previously introduced (Sec. 2.4.3) $\lambda_{\mathrm{v}}/\lambda_{\mathrm{d}}$ overestimation ratio quantifies the change in generated (v)DTDs. In the following, I denote the experimental statistics corresponding to the overestimation ratio with η_{o} and define it as:

$$\eta_{\mathrm{o}} = \frac{\text{vDTDs generated from } \mathbb{D} \text{ using overestimation}}{\text{number of measured DTDs in } \mathbb{D}}. \tag{2.44}$$

Table 2.5 summarizes the η_{b} and η_{o} values for some overestimated datasets with chosen values of m. The results show that for cases with lower $\lambda\tau$ values, the η_{o} overestimation rate is a fairly good estimate of the η_{b} bit retention efficiency, but as $\lambda\tau$ grows, the two values align less and less. This is most likely due to the correlations causing fewer interval equalities in the raw datasets and, therefore, dropped intervals during bit generation. This notion is further backed by the fact that for higher $\lambda\tau$ values, the achieved raw bit generation efficiencies[11] of 0.4876, 0.4057 and 0.3506 for

[10] Beside file rewinds in some Dieharder trials, $\mathbb{D}_{\mathrm{G}}$ had insufficient data for running the NIST STS, while $\mathbb{D}_{\mathrm{H}}$ had insufficient data for the NIST STS and TestU01 assessments.

[11] Where bit generation efficiency is defined as (total number of bits generated)/(total number of DTDs used).

datasets $\mathbb{D}_{\mathrm{F}}$ to $\mathbb{D}_{\mathrm{H}}$ are higher than the theoretical ones of 0.4871, 0.3743 and 0.2256 calculated according to Ref. [113] for the ideal case of using a restartable clock with no dead time.[12] This shows another symptom of lower-quality output when using the raw datasets for bit generation.

Table 2.5: Bit retention efficiencies, and overestimation rates. Not every element in a column corresponds to the same m.

	Overestimation parameter							
$\mathbb{D}_{\mathrm{ID}}$	$\{\mathbb{M}_{\mathrm{ID}}\}_1$		$\{\mathbb{M}_{\mathrm{ID}}\}_2$		$\{\mathbb{M}_{\mathrm{ID}}\}_3$		$\{\mathbb{M}_{\mathrm{ID}}\}_4$	
	η_{b}	η_{o}	η_{b}	η_{o}	η_{b}	η_{o}	η_{b}	η_{o}
$\mathbb{D}_{\mathrm{A}}$	0.9999	0.9999	0.9959	0.9959	0.9904	0.9903	0.9453	0.9457
$\mathbb{D}_{\mathrm{B}}$	0.9999	0.9999	0.9764	0.9765	0.8655	0.8655	–	–
$\mathbb{D}_{\mathrm{C}}$	0.9997	0.9997	0.9582	0.9582	0.7718	0.7719	–	–
$\mathbb{D}_{\mathrm{D}}$	0.9995	0.9995	0.9221	0.9221	0.6109	0.6110	0.3558	0.3558
$\mathbb{D}_{\mathrm{E}}$	0.9986	0.9986	0.9643	0.9643	0.6095	0.6096	–	–
$\mathbb{D}_{\mathrm{F}}$	0.8996	0.9010	0.8150	0.8166	0.3460	0.3466	–	–
$\mathbb{D}_{\mathrm{G}}$	0.4296	0.4719	0.0868	0.0952	–	–	–	–
$\mathbb{D}_{\mathrm{H}}$	0.1010	0.1624	0.0282	0.0453	0.0078	0.0126	0.0022	0.0035

2.6 Summary of the results

The proposed overestimation method offers a way to avoid the otherwise present correlations between consecutive measurement samples when using a continuously running clock with a photonic time-of-arrival based QRNG. Furthermore, the distribution of the output vDTDs of the method is the same as in the ideal case of zero starting phase and no dead time, thus preserving the memoryless property of the underlying physical process. Therefore, the method can also be used to avoid other unwanted effects of dead time. While these advantages come at the cost of reduced output speeds, depending on the architecture and bit generation method used, the benefits of my proposed scheme can often outweigh this cost. In the chapter, I also calculated and then evaluated and experimentally verified the main performance measures of the overestimation algorithm, showing a comprehensive overview of the capabilities, benefits, costs, and limits of using the scheme. Results presented in the chapter form the basis of Thesis I.

[12] Using $\lambda\tau$ values given in Table 2.4.

Another notable but previously not emphasized practical benefit of Algorithm 1 is its low complexity. This can make it an attractive solution for a wide range of practical realizations, especially in cases where generator operation is continuous and bit generation happens real-time. While I provided the main performance measures of the scheme when looking at it as an isolated standalone "building block" in the chapter, investigating its integrability with concrete physical architectures and bit generation schemes offers topics of potential further research.

Chapter 3

Post-processing for random number generators inspired by the probability integral transform

For a continuously distributed random variable X, the probability integral transform can be used to create a uniformly distributed $Y = F_X(X)$ transformed random variable. Unfortunately, this result only holds for variables with a continuous distribution, while the measurement statistics of QRNGs can typically be described by discrete random variables. In this chapter, I present a method inspired by the continuous probability integral transform to transform the discretely distributed measurement statistics of a QRNG device to a distribution that is close-to-uniform with an upper bound on the statistical distance between the transformed and ideal uniform distribution. Since the main goal of QRNGs is to generate uniformly distributed output bitstrings, the presented method can be used for post-processing with QRNG setups. First, I introduce my proposed method and calculate bounds on the bias of the samples and statistical distance of the output distribution from the uniform distribution in an idealized errorless scenario. I also show that by using the joint distribution of multiple independent samples, the bias and statistical distance can be lowered arbitrarily. Then, I calculate the previous bounds for non-ideal scenarios, showing the effects of generalized errors. I show that in this case, in contrast to the ideal scenario, the approximation errors of the scheme can no longer be made arbitrarily low and give a formula for finding optimal parameter sets corresponding to the lowest achievable statistical distance. I also investigate the practically interesting case of time-of-arrival QRNGs. Lastly, I present a simple practical bit generation scheme and use it to experimentally validate my previous theoretical results.

3.1 Post-processing based on the probability integral transform

The main idea of my proposed method is very similar to the one used for post-processing in Ref. [68] , where the authors paired a radioactive decay-based analog QRNG setup with a special analog integrator circuit to acquire uniformly distributed measurement results. In my work, I consider discrete random variables as inputs instead of continuous ones coming from an analog measurement. This, however, comes at the cost of approximation errors, for which I give various bounds in the following sections. To introduce the proposed method, let us first consider the idealized case of a fully known input distribution without any potential error or noise sources.

3.1.1 Transformations of discrete random variables

Let X be a discrete random variable with possible x_i values from the set $A = \{ x_i : i = 0, 1, 2, \dots \}$ with $p_i = \Pr(X = x_i)$ probabilities. Let $t : A \to B$ be an arbitrary function, $B = \{ y_j : j = 0, 1, 2, \dots \}$. Then, the $Y = t(X)$ discrete random variable has $y_j \in B$ possible values with

$$\Pr\big(Y = y_j\big) = \sum_{\forall x_i : t(x_i) = y_j} p_i = \sum_{i \,|\, t(x_i) = y_j} p_i. \tag{3.1}$$

If the function $t(\cdot)$ is bijective meaning its inverse $t^{-1}(\cdot)$ exists, then $X = t^{-1}(Y)$ and the previous equation can be also written as

$$\Pr(Y = y) = \Pr\big(t^{-1}(Y) = t^{-1}(y)\big) = \Pr\big(X = t^{-1}(y)\big). \tag{3.2}$$

3.1.2 Approximate uniform mapping

To introduce the proposed method, let us first consider the idealized case of a fully known input distribution without any potential error or noise sources. Consider the following transformation based on the continuous probability integral transform [118] for processing measurement samples:
Let D be a discrete random variable with possible d_i outcomes from the set $\{ d_i : i = 0, 1, 2, \dots \}$ with $p_i = \Pr(D = d_i)$ probabilities. Define a bijective $t(\cdot)$ function that assigns a c_i coordinate value to each possible d_i outcome, such that

$$c_i = t(d_i) = \sum_{j=0}^{i} \Pr\big(D = d_j\big) = \sum_{j=0}^{i} p_j. \tag{3.3}$$

This way, the $C = t(D)$ transformed discrete random variable has possible $c_i \in [0,1]$ values from the set $\{\, c_i : i = 0, 1, 2, \dots \,\}$ with probabilities $p_i = \Pr(C = c_i) = \Pr(D = d_i)$. For notational convenience in later sections, also set the edge case $c_{-1} \triangleq 0$. Note that the c_i values form a monotonically increasing sequence in i on $[0,1]$.

Then for $y \in [0,1]$,

$$\Pr(C < y) = \sum_{j \,|\, c_j \le y} \Pr\big(C = c_j\big) = \sum_{j=0}^{N_y} \Pr\big(C = c_j\big) = \sum_{j=0}^{N_y} p_j = c_{N_y}, \tag{3.4}$$

where $N_y = \arg\max_i \big\{ c_i \mid c_i \le y \big\}$ is the biggest index of c_i for which $c_i \le y$ still holds. Notice that for a random variable U uniformly distributed on $[0,1]$, $\Pr(U < y) = y$; therefore, (3.4) approximates the behavior of an ideal uniform distribution with an error of $0 \le y - c_{N_y} < c_{N_y+1} - c_{N_y}$.

Furthermore, the approximation error is further upper bounded by the maximum of $(c_{i+1} - c_i)$, giving

$$\begin{aligned} \Pr(U < y) - \Pr(C < y) = y - c_{N_y} < c_{N_y+1} - c_{N_y} \\ \le \max_i (c_{i+1} - c_i) = \max_i \left(\sum_{j=0}^{i+1} p_j - \sum_{j=0}^{i} p_j \right) = \max_i p_i. \end{aligned} \tag{3.5}$$

Similarly,

$$\Pr(x \le C < y) = \Pr(C < y) - \Pr(C < x) = \sum_{j=0}^{N_y} p_j - \sum_{j=0}^{N_x} p_j = c_{N_y} - c_{N_x}, \tag{3.6}$$

where $N_x = \arg\max_i \big\{ c_i \mid c_i \le x \big\}$ is the biggest index of c_i for which $c_i \le x$ still holds, and

$$\begin{aligned} \big|\Pr(x \le U < y) - \Pr(x \le C < y)\big| &= \big|y - c_{N_y} - (x - c_{N_x})\big| \\ &< \max(c_{N_y+1} - c_{N_y}, c_{N_x+1} - c_{N_x}) \le \max_i p_i. \end{aligned} \tag{3.7}$$

The previous equation can also be rewritten using the $H_\infty(D)$ min-entropy giving:

$$\big|\Pr(x \le U < y) - \Pr(x \le C < y)\big| < \max_i p_i = 2^{-H_\infty(D)}. \tag{3.8}$$

3.1.3 Generating close-to-uniform output sequences

Typically, the expected output distribution of random number generators is uniform. Given a D input distribution over $\{0,1\}^{n_e}$ with 2^{n_e} possible values, the goal of post-processing is to create an R output distribution on the

$\{ r_0, r_1 \dots r_l \dots r_{2^{m_e}-1} \}, l \in \mathbb{N}, 0 \leq l < 2^{m_e}$, sample space over $\{0,1\}^{m_e}$ that is statistically ϵ-close to the U uniform distribution over $\{0,1\}^{m_e}$ with 2^{m_e} possible values.

If the D distribution is known, the previously presented idea can be used to transform an input distribution into one approximating the uniform distribution with an upper bound on approximation error using the following scheme:

1. Using the $t(\cdot)$ function presented in Sec. 3.1.2 assign c_i values on $[0,1]$ to all possible d_i outcomes, thus creating a $C = t(D)$ transformed discrete random variable.

2. Assign output values to all c_i such that if $\frac{l}{2^{m_e}} < c_i \leq \frac{l+1}{2^{m_e}}$ then r_l is assigned.

According to (3.6) and (3.7), the following holds for each of the r_l output values:

$$\Pr(R = r_l) = \Pr\left(\frac{l}{2^{m_e}} \leq C < \frac{l+1}{2^{m_e}} \right), \tag{3.9}$$

therefore,

$$d(r_l) = \left| \Pr\left(\frac{l}{2^{m_e}} \leq U < \frac{l+1}{2^{m_e}} \right) - \Pr(R = r_l) \right| < \max_i p_i, \tag{3.10}$$

with $d(r_l)$ noting the absolute deviation in probability of a particular r_l output bit sequence from the ideal case of $\Pr(R = r_l) = 1/2^{m_e}$. This means that the R distribution of the r_l output values approximates a uniform one over $\{0,1\}^{m_e}$ with an upper bound on approximation error given by (3.10).

Upper bound on bit bias

Note that this upper bound on approximation error can also be used to give an upper bound on the bias of a single bit in the created m_e bit long sequence. Let $\mathbf{B} = \{ B_0, B_1, \dots, B_{m_e-1} \}$ be the bits in the m_e bit long sequence with $\mathbf{x} = \{ x_0, x_1, \dots, x_{m_e-1} \}$ bit values and A be the event of $\{B_0 = x_0, \dots, B_{i-1} = x_{i-1}, B_{i+1} = x_{i+1}, \dots, B_{m_e-1} = x_{m_e-1}\}$, where all $\mathbf{x}$ values for the $\mathbf{B}$ bits in the sequence are set except for the ith bit. Then, for any ith bit in the sequence:

$$\begin{aligned} \Pr\big(B_i = x_i \mid A\big) &= \frac{\Pr(\mathbf{B} = \mathbf{x})}{\Pr(A)} = \frac{\Pr(\mathbf{B} = \mathbf{x})}{\Pr(A, B_i = x_i) + \Pr(A, B_i = \overline{x_i})} \\ &= \frac{\Pr(\mathbf{B} = \mathbf{x})}{\Pr(\mathbf{B} = \mathbf{x}) + \Pr(A, B_i = \overline{x_i})}. \end{aligned} \tag{3.11}$$

Looking at a worst-case scenario, this gives

$$\begin{aligned}\left|\Pr\big(B_i = x_i \mid A\big) - \frac{1}{2}\right| &\leq \left|\frac{\frac{1}{2^{m_\mathrm{e}}} + \max_l d(r_l)}{\frac{1}{2^{m_\mathrm{e}}} + \max_l d(r_l) + \frac{1}{2^{m_\mathrm{e}}} - \max_l d(r_l)} - \frac{1}{2}\right| \\ = \left|\frac{\max_l d(r_l)}{2 \cdot \frac{1}{2^{m_\mathrm{e}}}}\right| &= \left|2^{m_\mathrm{e}-1} \max_l d(r_l)\right| < 2^{m_\mathrm{e}-1} \max_i p_i\end{aligned} \tag{3.12}$$

for an upper bound on individual bit bias, since the terms in (3.11) can be upper and lower bounded by $\frac{1}{2^{m_\mathrm{e}}} \pm \max_l d(r_l)$.

Statistical distance from the uniform distribution

According to Sec. 1.4.2 the statistical distance (or the total variation distance) between two discrete probability distributions X and Y is given by

$$\Delta(X, Y) = \frac{1}{2} \sum_v \big|\Pr(X = v) - \Pr(Y = v)\big|. \tag{3.13}$$

An upper bound for the statistical distance of the output of the previously presented scheme from the goal uniform distribution can be written as:

$$\begin{aligned}\Delta(U, R) &= \frac{1}{2} \sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \leq U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &\leq 2^{m_\mathrm{e}-1} \max_l d(r_l) < 2^{m_\mathrm{e}-1} \max_i p_i = 2^{-(H_\infty(D) - m_\mathrm{e} + 1)}\end{aligned} \tag{3.14}$$

according to the upper bound on approximation error presented in (3.10). Notice that the end results of (3.12) and (3.14) are the same, meaning that this expression upper bounds both the individual bit bias and the statistical distance. Adopting the common ϵ notation for the upper bound of statistical distance, the criterion for the output distribution to be ϵ-close to the expected uniform distribution in terms of m_e, ϵ, and $H_\infty(D)$ is then:

$$\log_2 \epsilon < m_\mathrm{e} - H_\infty(D) - 1. \tag{3.15}$$

Note that this means that in the ideal case, the scheme yields better ϵ values for fixed m_e and $H_\infty(D)$ parameters than even universal hash based extractors, such as the one presented in Chapter 4 (See (4.1) for the expression of ϵ.), thus allowing for better bit generation efficiency in theory.

3.1.4 Uniform bit generation scheme for photonic time-of-arrival based random number generators

Ideally, the time differences between photon detection events are exponentially distributed, and when measured with an ideal restartable clock measurement setup,

the D input distribution formed from the measurement results is according to Sec. 2.2.1. Therefore,

$$p_i = p_n = \Pr(D = d_i = n) = \mathrm{e}^{-n\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right), \tag{3.16}$$

allowing the convenient choice of $d_i = n, i = n$ (an indexing scheme where the nth possible measurement outcome is the one with the value of n) since both the i index of d_i and the possible n measurement outcomes are non-negative integers. The $t(\cdot)$ assignment function of c_n is then:

$$t(d_i) = t(n) = c_n = \sum_{j=0}^{n} p_j = \sum_{j=0}^{n} \mathrm{e}^{-j\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right) = 1 - \mathrm{e}^{-(n+1)\lambda\tau}. \tag{3.17}$$

The post-processing scheme can then be defined following the steps presented in Sec. 3.1.3. The upper bound of the scheme's approximation error according to (3.10) is then:

$$\begin{aligned}\left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| &< \max(c_{N_y+1} - c_{N_y}, c_{N_x+1} - c_{N_x}) \\ &\le \max_n p_n = \left(1 - \mathrm{e}^{-\lambda\tau}\right).\end{aligned} \tag{3.18}$$

The value of N_y and N_x can also be calculated using the $t^{-1}(\cdot)$ inverse function, giving

$$N_y \le -\frac{\ln(1-y)}{\lambda\tau} - 1 \Rightarrow N_y = \left\lfloor -\frac{\ln(1-y)}{\lambda\tau} - 1 \right\rfloor \tag{3.19}$$

where $\lfloor\cdot\rfloor$ is the floor function (or greatest integer function). The value for N_x can be calculated similarly. The $c_{N_y+1} - c_{N_y}$ quantity in (3.18) is then:

$$c_{N_y+1} - c_{N_y} = 1 - \mathrm{e}^{-(N_y+2)\lambda\tau} - \left(1 - \mathrm{e}^{-(N_y+1)\lambda\tau}\right) = \mathrm{e}^{-\left\lfloor -\frac{\ln(1-y)}{\lambda\tau}\right\rfloor\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right). \tag{3.20}$$

Notice, that (3.20) is decreasing in N_y and therefore

$$\begin{aligned}&\max(c_{N_y+1} - c_{N_y}, c_{N_x+1} - c_{N_x}) = c_{N_x+1} - c_{N_x} \\ &= \mathrm{e}^{-\left\lfloor -\frac{\ln(1-x)}{\lambda\tau}\right\rfloor\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right) = \mathrm{e}^{\left\lceil \frac{\ln(1-x)}{\lambda\tau}\right\rceil\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right).\end{aligned} \tag{3.21}$$

The upper bound for statistical distance according to (3.14) is

$$\begin{aligned}\Delta(U, R) &< \frac{1}{2}\sum_l \mathrm{e}^{\left\lceil \ln\left(1-\frac{l}{2^{m_\mathrm{e}}}\right)\frac{1}{\lambda\tau}\right\rceil\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right) \\ &\le 2^{m_\mathrm{e}-1}\max_n p_n = 2^{m_\mathrm{e}-1}\cdot\left(1 - \mathrm{e}^{-\lambda\tau}\right) = 2^{-(H_\infty(D)-m_\mathrm{e}+1)},\end{aligned} \tag{3.22}$$

with (3.15) taking the form of

$$\log_2\epsilon < m_\mathrm{e} - H_\infty(D) - 1 = m_\mathrm{e} - \log_2\left(1 - \mathrm{e}^{-\lambda\tau}\right) - 1. \tag{3.23}$$

3.1.5 Bit generation scheme using the joint distribution of multiple independent identically distributed measurement samples

In the following, let the input distribution be the joint probability distribution of multiple i.i.d. samples taken from the same underlying process. Then, the possible $\underline{d}$ outcomes of the input $\underline{D}$ joint distribution are $\underline{q} = (q_0, q_1, \ldots, q_{Q-1})$ Q-tuples, where the elements are the possible outcome combinations of the Q individual samples, with $\underline{p}_i = \Pr(\underline{D} = \underline{d}_i) = \Pr\big(D_0 = q_{0,i}, D_1 = q_{1,i}, \ldots, D_{Q-1} = q_{Q-1,i}\big) = \prod_{\vartheta=0}^{Q-1} p_{q_{\vartheta,i}}$. Also, let us denote the probabilities of individual samples with $p_{q_\vartheta} = \Pr(D_\vartheta = q_\vartheta)$.
Let $t_Q(\cdot)$ be the bijective function that determines the i index assignment of the possible $\underline{d}_i$ Q-tuples meaning that $i = t_Q(\underline{q}) = t_Q((q_{0,i}, q_{1,i}, \ldots, q_{Q-1,i}))$ and $\underline{q} = (q_{0,i}, q_{1,i}, \ldots, q_{Q-1,i}) = t_Q^{-1}(i)$. According to Sec. 3.1.2

$$\underline{c}_i = t(\underline{d}_i) = \sum_{j=0}^{i} \Pr\Big(\underline{D} = \underline{d}_j\Big) = \sum_{j=0}^{i} \underline{p}_j = \sum_{j=0}^{i} \prod_{\vartheta=0}^{Q-1} p_{q_{\vartheta,j}}, \tag{3.24}$$

where $q_{\vartheta,j}$ are the respective elements of the $\underline{q} = t_Q^{-1}(j)$ tuples and $p_{q_{\vartheta,j}}$ are the associated probabilities. According to (3.10)

$$\begin{aligned}\left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| &< \max(c_{N_y+1} - c_{N_y}, c_{N_x+1} - c_{N_x}) \\ &\le \max_i \underline{p}_i = \prod_{\vartheta=0}^{Q-1} \max_i p_{q_{\vartheta,i}} = \left(\max_i p_i\right)^Q,\end{aligned} \tag{3.25}$$

and the upper bound for the statistical distance is:

$$\begin{aligned}\Delta(U,R) = \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| &< 2^{m_\mathrm{e}-1} \max_i \underline{p}_i \\ &= 2^{-(H_\infty(\underline{D}) - m_\mathrm{e} + 1)} = 2^{-(QH_\infty(D) - m_\mathrm{e} + 1)},\end{aligned} \tag{3.26}$$

giving

$$\log_2 \epsilon < m_\mathrm{e} - H_\infty(\underline{D}) - 1 = m_\mathrm{e} - QH_\infty(D) - 1, \tag{3.27}$$

where $H_\infty(D)$ is the min-entropy of an individual sample and $H_\infty(\underline{D})$ is the min-entropy of the joint distribution of multiple samples, respectively. Notice that by increasing Q, ϵ decreases, allowing for creating output distributions that are arbitrarily close to the uniform distribution by using the joint distribution of sufficiently many individual measurement samples as input for the scheme.

3.1.6 Using the joint distribution of multiple photon detection time differences

The time differences between individual photon detection events are exponentially distributed, therefore, the joint distribution of multiple measurement results is:

$$\begin{aligned}\underline{p}_i &= \Pr(\underline{D}=\underline{d}_i) = \Pr\big(D_0 = q_{0,i}, D_1 = q_{1,i}, \ldots, D_{Q-1} = q_{Q-1,i}\big) \\ &= \prod_{\vartheta=0}^{Q-1} p_{q_\vartheta} = \Big(1-\mathrm{e}^{-\lambda\tau}\Big)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_{\vartheta,i}\lambda\tau}.\end{aligned} \tag{3.28}$$

The $\underline{c}_i$ values are:

$$\underline{c}_i = \sum_{j=0}^{i} \Pr\Big(\underline{D}=\underline{d}_j\Big) = \sum_{j=0}^{i} \underline{p}_j = \sum_{j=0}^{i} \left(\Big(1-\mathrm{e}^{-\lambda\tau}\Big)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_{\vartheta,j}\lambda\tau}\right), \tag{3.29}$$

and the upper bound of approximation error is

$$\begin{aligned}\left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R=r_l)\right| &< \max(c_{N_y+1}-c_{N_y}, c_{N_x+1}-c_{N_x}) \\ &\le \max_i \underline{p}_i = \Big(1-\mathrm{e}^{-\lambda\tau}\Big)^Q.\end{aligned} \tag{3.30}$$

Similarly to (3.19) and (3.20) N_y and N_x can be calculated:

$$N_y \le t^{-1}(y) \Rightarrow N_y = \left\lfloor t^{-1}(y) \right\rfloor, \tag{3.31}$$

as well as,

$$c_{N_y+1} - c_{N_y} = \sum_{j=0}^{N_y+1} \underline{p}_j - \sum_{j=0}^{N_y} \underline{p}_j = \underline{p}_{N_y+1} = \Big(1-\mathrm{e}^{-\lambda\tau}\Big)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_{\vartheta,N_y+1}\lambda\tau}. \tag{3.32}$$

Finally, the upper bound for the statistical distance is:

$$\begin{aligned}\Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R=r_l)\right| \\ &< \frac{1}{2}\sum_l \Big|\max(c_{N_x+1}-c_{N_x}, c_{N_y+1}-c_{N_y})\Big| = \frac{1}{2}\sum_l \Big|\max(\underline{p}_{N_y+1}, \underline{p}_{N_x+1})\Big| \\ &\le 2^{m_\mathrm{e}-1} \max_i \underline{p}_i = 2^{m_\mathrm{e}-1}\Big(1-\mathrm{e}^{-\lambda\tau}\Big)^Q = 2^{-(H_\infty(\underline{D})-m_\mathrm{e}+1)} = 2^{-(QH_\infty(D)-m_\mathrm{e}+1)}.\end{aligned} \tag{3.33}$$

3.2 Output characteristics in the presence of general estimation errors

3.2.1 General distributions

For the approximation scheme to work correctly, the distribution of D must be known. However, in most situations, fully characterizing D may not be practically feasible. Therefore, it is worth investigating the case where the probability of the d_i outcomes is not exactly known but with some e_i error, giving $\Pr(D = d_i) = p_i = \hat{p}_i + e_i$, where $\hat{p}_i$ represents our best estimate of the true p_i probability.

Assuming that the $t(\cdot)$ coordinate assignment function assigns c_i coordinate values to d_i outcomes according to the estimated $\hat{p}_i$ probabilities, the assignment rule presented in (3.3) changes to

$$c_i = \sum_{j=0}^{i} \hat{p}_j = \sum_{j=0}^{i} \left(p_j - e_j\right), \tag{3.34}$$

and

$$\Pr(C < y) = \sum_{j=0}^{N_y} p_j = c_{N_y} + \sum_{j=0}^{N_y} e_j. \tag{3.35}$$

The approximation error according to (3.5) is then

$$\begin{aligned} \Pr(U < y) - \Pr(C < y) &= y - \left(c_{N_y} + \sum_{j=0}^{N_y} e_j \right) \\ &< c_{N_y+1} - c_{N_y} + \sum_{j=0}^{N_y} e_j \leq \max_i \hat{p}_i + \sum_{j=0}^{N_y} e_j. \end{aligned} \tag{3.36}$$

Similarly,

$$\begin{aligned} &\Pr(x \leq C < y) = \Pr(C < y) - \Pr(C < x) \\ &= c_{N_y} + \sum_{j=0}^{N_y} e_j - \left(c_{N_x} + \sum_{j=0}^{N_x} e_j \right) = c_{N_y} - c_{N_x} + \sum_{j=N_x+1}^{N_y} e_j, \end{aligned} \tag{3.37}$$

and

$$\begin{aligned} &\left|\Pr(x \leq U < y) - \Pr(x \leq C < y)\right| \\ &= \left| y - \left(c_{N_y} + \sum_{j=0}^{N_y} e_j \right) - \left(x - \left(c_{N_x} + \sum_{j=0}^{N_x} e_j \right) \right) \right| \\ &< \left| c_{N_y+1} - c_{N_y} - (c_{N_x+1} - c_{N_x}) + \sum_{j=N_x+1}^{N_y} e_j \right| \leq \max_i \hat{p}_i + \left| \sum_{j=N_x+1}^{N_y} e_j \right|. \end{aligned} \tag{3.38}$$

Naturally, this can also be written in terms of the estimated $H_\infty(\hat{D}) = -\log_2 \max_i \hat{p}_i$ min-entropy giving:

$$\left|\Pr(x \leq U < y) - \Pr(x \leq C < y)\right| < 2^{-H_\infty(\hat{D})} + \left|\sum_{j=N_x+1}^{N_y} e_j\right|. \tag{3.39}$$

Output bit generation

In the case of output bit sequence assignment (3.10) can be rewritten with errors taken into account as:

$$\begin{aligned} d(r_l) &= \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \leq U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &< \left|c_{N_y+1} - c_{N_y} - (c_{N_x+1} - c_{N_x}) + \sum_{j=N_x+1}^{N_y} e_j\right| \leq \max_i \hat{p}_i + \left|\sum_{j=N_x+1}^{N_y} e_j\right| \end{aligned} \tag{3.40}$$

with the choice of $x = \frac{l}{2^{m_\mathrm{e}}}$ and $y = \frac{l+1}{2^{m_\mathrm{e}}}$. Consequently, the upper bound on bit bias is

$$\left|\Pr(B_i = x_i) - \frac{1}{2}\right| \leq \left|2^{m_\mathrm{e}-1} \max_l d(r_l)\right| < 2^{m_\mathrm{e}-1} \max_i \hat{p}_i + 2^{m_\mathrm{e}-1} \max_l \left|\sum_{j=N_x+1}^{N_y} e_j\right|. \tag{3.41}$$

The statistical distance from the expected uniform output is then

$$\begin{aligned} \Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \leq U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &< \frac{1}{2}\sum_l \left(\max_i \hat{p}_i + \left|\sum_{j=N_x+1}^{N_y} e_j\right|\right) = 2^{m_\mathrm{e}-1} \max_i \hat{p}_i + \frac{1}{2}\sum_l \left|\sum_{j=N_x+1}^{N_y} e_j\right| \\ &\leq 2^{m_\mathrm{e}-1} \max_i \hat{p}_i + 2^{m_\mathrm{e}-1} \max_l \left|\sum_{j=N_x+1}^{N_y} e_j\right| = 2^{-(H_\infty(\hat{D})-m_\mathrm{e}+1)} + 2^{m_\mathrm{e}-1} \max_l \left|\sum_{j=N_x+1}^{N_y} e_j\right|. \end{aligned} \tag{3.42}$$

Alternatively, the sum of the individual $|e_i|$ errors can also be used for an upper

bound:

$$\begin{aligned}\Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R=r_l)\right| \\ &< \frac{1}{2}\sum_l \left(\max_i \hat{p}_i + \left|\sum_{j=N_x+1}^{N_y} e_j\right|\right) = 2^{m_\mathrm{e}-1}\max_i \hat{p}_i + \frac{1}{2}\sum_l \left|\sum_{j=N_x+1}^{N_y} e_j\right| \\ &\le 2^{m_\mathrm{e}-1}\max_i \hat{p}_i + \frac{1}{2}\sum_i |e_i| = 2^{-(H_\infty(\hat{D})-m_\mathrm{e}+1)} + \frac{1}{2}\sum_i |e_i| \\ &\le 2^{-(H_\infty(\hat{D})-m_\mathrm{e}+1)} + \frac{1}{2}\sum_i \left|\max_i e_i\right|.\end{aligned} \tag{3.43}$$

Results of (3.40) and (3.42) show that the quality (distance from the expected uniform) of the output R distribution is heavily influenced by the accuracy of estimation of the input D distribution. In particular, by the $\left|\sum_{j=N_x+1}^{N_y} e_j\right|$ accumulated errors for each (N_x, N_y) interval with $x = \frac{l}{2^{m_\mathrm{e}}}, y = \frac{l+1}{2^{m_\mathrm{e}}}$ corresponding to the r_l output values.

3.2.2 Using single photon arrival times

In the practical scenario of a QRNG based on photon arrival times, the input D distribution is the distribution of the measured DTDs (or in the case of using the algorithm presented in Chapter 2, the distribution of the calculated vDTDs). According to Sec. 3.2.1, the upper bound of the scheme's approximation error in the presence of general errors is given by (3.42) or (3.43), by substituting $H_\infty(\hat{D}) = \log_2\left(1-\mathrm{e}^{-\lambda\tau}\right)$. Notice that in (3.42) beside the min-entropy, the accumulated $\left|\sum_{j=N_x+1}^{N_y} e_j\right|$ errors are the other major contributor to statistical distance, meaning that the mitigation of systematic errors contributing to this quantity (like ones causing e_i errors with the same sign) is important.

3.2.3 Effect of estimation errors when using the joint distribution of multiple samples

Similarly to Sec. 3.2.1 let us investigate the effect of estimation errors when using the joint distribution of multiple samples utilizing the notations introduced in Sec. 3.1.5. The probabilities of the individual samples are then $p_{q_\vartheta} = \Pr(D_\vartheta = q_\vartheta) = \hat{p}_{q_\vartheta} + e_{q_\vartheta}$, with $\hat{p}_{q_\vartheta}$ estimated probability and e_{q_ϑ} estimation error for a particular individual sample in the joint distribution. Using the $\underline{\hat{p}}_i = \prod_{\vartheta=0}^{Q-1} \hat{p}_{q_\vartheta,i}$ estimate for the joint

probability, the actual $\underline{p}_i$ joint probabilities are then:

$$\underline{p}_i = \Pr(\underline{D} = \underline{d}_i) = \hat{\underline{p}}_i + \underline{e}_i = \prod_{\vartheta=0}^{Q-1} p_{q_{\vartheta,i}} = \prod_{\vartheta=0}^{Q-1} \left(\hat{p}_{q_{\vartheta,i}} + e_{q_{\vartheta,i}}\right) \tag{3.44}$$

and

$$\underline{e}_i = \underline{p}_i - \hat{\underline{p}}_i = \prod_{\vartheta=0}^{Q-1} \left(\hat{p}_{q_{\vartheta,i}} + e_{q_{\vartheta,i}}\right) - \prod_{\vartheta=0}^{Q-1} \hat{p}_{q_{\vartheta,i}}, \tag{3.45}$$

with

$$\underline{c}_i = \sum_{j=0}^{i} \hat{\underline{p}}_j = \sum_{j=0}^{i} \left(\underline{p}_j - \underline{e}_j\right). \tag{3.46}$$

The approximation errors and statistical distance can also be calculated by substituting the relevant $\underline{p}_i, \hat{\underline{p}}_i$ and $\underline{e}_i$ into results of Sec. 3.2.1, eventually giving:

$$\begin{aligned} &\left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \leq U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &< \left|c_{N_y+1} - c_{N_y} - (c_{N_x+1} - c_{N_x}) + \sum_{j=N_x+1}^{N_y} \underline{e}_j\right| \leq \max_i \hat{\underline{p}}_i + \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right| \end{aligned} \tag{3.47}$$

and

$$\begin{aligned} \Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \leq U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &< \frac{1}{2}\sum_l \left(\max_i \hat{\underline{p}}_i + \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right|\right) = 2^{m_\mathrm{e}-1} \max_i \hat{\underline{p}}_i + \frac{1}{2}\sum_l \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right| \\ &\leq 2^{m_\mathrm{e}-1} \max_i \hat{\underline{p}}_i + 2^{m_\mathrm{e}-1} \max_l \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right| = 2^{-(QH_\infty(\hat{D})-m_\mathrm{e}+1)} + 2^{m_\mathrm{e}-1} \max_l \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right| \end{aligned} \tag{3.48}$$

or

$$\begin{aligned} \Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \leq U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &< \frac{1}{2}\sum_l \left(\max_i \hat{\underline{p}}_i + \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right|\right) = 2^{m_\mathrm{e}-1} \max_i \hat{\underline{p}}_i + \frac{1}{2}\sum_l \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right| \\ &\leq 2^{m_\mathrm{e}-1} \max_i \hat{\underline{p}}_i + \frac{1}{2}\sum_i |\underline{e}_i| = 2^{m_\mathrm{e}-1} \left(\max_i \hat{p}_i\right)^Q + \frac{1}{2}\sum_i |\underline{e}_i| \\ &= 2^{-(QH_\infty(\hat{D})-m_\mathrm{e}+1)} + \frac{1}{2}\sum_i |\underline{e}_i| \leq 2^{-(QH_\infty(\hat{D})-m_\mathrm{e}+1)} + \frac{1}{2}\sum_i \left|\max_i \underline{e}_i\right|. \end{aligned} \tag{3.49}$$

Additionally, let us investigate the behavior of the $\sum_i |\underline{e}_i|$ error term. For this, start with the simplified general case of the joint distribution of two independent (but not necessarily identically distributed) samples. Then, $\Pr(D_1 = q_1) = p_{q_1} = \hat{p}_{q_1} + e_{q_1}$, where p_{q_1} is the actual probability of the first sample being q_1 and $\hat{p}_{q_1}$ is our best estimate for p_{q_1} with e_{q_1} error. Similarly, for the second sample $\Pr(D_2 = q_2) = p_{q_2} = \hat{p}_{q_2} + e_{q_2}$, where p_{q_2} is the actual probability and $\hat{p}_{q_2}$ is the estimate with e_{q_2} error. Then the joint distribution of the two samples is:

$$\begin{aligned}\Pr(D_1 = q_1, D_2 = q_2) = \underline{p}_{(q_1,q_2)} = p_{q_1}p_{q_2} = (\hat{p}_{q_1} + e_{q_1})(\hat{p}_{q_2} + e_{q_2}) \\ = \hat{p}_{q_1}\hat{p}_{q_2} + \hat{p}_{q_1}e_{q_2} + \hat{p}_{q_2}e_{q_1} + e_{q_1}e_{q_2} = \underline{\hat{p}}_{(q_1,q_2)} + \underline{e}_{(q_1,q_2)}\end{aligned} \tag{3.50}$$

Knowing that, $\underline{\hat{p}}_{(q_1,q_2)} = \hat{p}_{q_1}\hat{p}_{q_2}$,

$$\underline{e}_{(q_1,q_2)} = \underline{p}_{(q_1,q_2)} - \underline{\hat{p}}_{(q_1,q_2)} = \hat{p}_{q_1}e_{q_2} + \hat{p}_{q_2}e_{q_1} + e_{q_1}e_{q_2}. \tag{3.51}$$

The $\sum_{q_1,q_2} \underline{e}_{(q_1,q_2)}$ error term for this joint distribution is then:

$$\begin{aligned}\sum_{q_1,q_2} \underline{e}_{(q_1,q_2)} = \sum_{q_1}\sum_{q_2} \underline{e}_{(q_1,q_2)} = \sum_{q_1}\sum_{q_2} \left(\hat{p}_{q_1}e_{q_2} + \hat{p}_{q_2}e_{q_1} + e_{q_1}e_{q_2}\right) \\ = \sum_{q_1}\hat{p}_{q_1}\sum_{q_2}e_{q_2} + \sum_{q_1}e_{q_1}\sum_{q_2}\hat{p}_{q_2} + \sum_{q_1}e_{q_1}\sum_{q_2}e_{q_2} \\ = \sum_{q_1}e_{q_1} + \sum_{q_2}e_{q_2} + \sum_{q_1}e_{q_1}\sum_{q_2}e_{q_2}.\end{aligned} \tag{3.52}$$

Similarly,

$$\begin{aligned}\sum_{q_1,q_2} \left|\underline{e}_{(q_1,q_2)}\right| = \sum_{q_1}\sum_{q_2} \left(\hat{p}_{q_1}\left|e_{q_2}\right| + \hat{p}_{q_2}\left|e_{q_1}\right| + \left|e_{q_1}\right|\left|e_{q_2}\right|\right) \\ = \sum_{q_1}\left|e_{q_1}\right| + \sum_{q_2}\left|e_{q_2}\right| + \sum_{q_1}\left|e_{q_1}\right|\sum_{q_2}\left|e_{q_2}\right|.\end{aligned} \tag{3.53}$$

Using the results of (3.52) and (3.53), the calculation of these error terms for the joint distribution of multiple samples is also possible, by using a step-by-step approach.[13] Importantly, these results also show that $\sum_{q_1,q_2}\left|\underline{e}_{(q_1,q_2)}\right| \geq \sum_{q_1}\left|e_{q_1}\right|$ and $\sum_{q_1,q_2}\left|\underline{e}_{(q_1,q_2)}\right| \geq \sum_{q_2}\left|e_{q_2}\right|$. This means that by using the joint distribution of multiple samples as input, the error terms present in the upper bound for ϵ increase, and thus, ϵ cannot be made arbitrarily small. Notice, that in case the samples are i.i.d. with the same error terms, then the step-by-step calculation can be simplified according to Sec. A.4, giving

$$\sum_i \underline{e}_i = \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1\right)^Q - 1. \tag{3.54}$$

[13]For example, to calculate results for the joint distribution of three samples, one can first calculate the quantities corresponding to the joint distribution of two samples and then use it paired with the quantities corresponding to a single sample to get results for the case of three samples.

Similarly,

$$\sum_i |\underline{e}_i| = \left(\sum_{q_\vartheta} \left| e_{q_\vartheta} \right| + 1 \right)^Q - 1. \tag{3.55}$$

3.2.4 Minimum of achievable statistical distance

According to the previous section, contrary to the ideal scenario, the statistical distance can only be lowered until a certain value when estimation errors are present. This value can be calculated by finding the minimum of (3.49) in Q for any given m_e by solving

$$\frac{\mathrm{d}}{\mathrm{d}Q} \left(2^{m_\mathrm{e}-1} \left(\max_i \hat{p}_i \right)^Q + \frac{1}{2} \left(\left(\sum_i |e_i| + 1 \right)^Q - 1 \right) \right) = 0.$$

$$2^{m_\mathrm{e}-1} \log \left(\max_i \hat{p}_i \right) \left(\max_i \hat{p}_i \right)^Q + \frac{1}{2} \log \left(\sum_i |e_i| + 1 \right) \left(\sum_i |e_i| + 1 \right)^Q = 0 \tag{3.56}$$

$$2^{m_\mathrm{e}} \log \left(\max_i \hat{p}_i \right) \left(\max_i \hat{p}_i \right)^Q + \log \left(\sum_i |e_i| + 1 \right) \left(\sum_i |e_i| + 1 \right)^Q = 0$$

The solution for equations of the form $a_0 b_0^x + a_1 b_1^x$ (like (3.56)) is

$$x = -\frac{\log \left(-\frac{a_0}{a_1} \right)}{\log b_0 - \log b_1}. \tag{3.57}$$

Since in the case of (3.56) $a_0 = 2^{m_\mathrm{e}} \log \left(\max_i \hat{p}_i \right)$ is always negative and $a_1 = \log \left(\sum_i |e_i| + 1 \right)$ is always positive, with also positive $b_0 = \max_i \hat{p}_i$ and $b_1 = \left(\sum_i |e_i| + 1 \right)$ values, (3.57) has $x = Q \in \mathbb{R}$ solutions. Using (3.57) and (3.56), the Q_opt value corresponding to the minimum achievable statistical distance is then:

$$Q_\mathrm{opt} = -\frac{\log \left(-\frac{2^{m_\mathrm{e}} \log \left(\max_i \hat{p}_i \right)}{\log \left(\sum_i |e_i| + 1 \right)} \right)}{\log \left(\max_i \hat{p}_i \right) - \log \left(\sum_i |e_i| + 1 \right)}. \tag{3.58}$$

Note that since $\log \left(\max_i \hat{p}_i \right) < 0$ and $\log \left(\sum_i |e_i| + 1 \right) \geq 0$, Q_opt is positive. Since in a real scenario, we can only choose integer values for Q, the optimal choice will naturally either be $\lfloor Q_\mathrm{opt} \rfloor$ or $\lceil Q_\mathrm{opt} \rceil$.

3.2.5 Effect of estimation errors when using the distribution of multiple photon arrival times

Similarly to Sec. 3.2.2, the relevant quantities of the case of using the joint distribution of multiple photon arrival times as the input distribution can be calculated using the

general results of Sec. 3.2.3. Note that Sec. 3.2.3 assumes i.i.d. individual samples. Therefore, if the measured individual DTDs cannot be considered i.i.d., additional processing steps before using the post-processing scheme (like the one presented in Chapter 2) might be advisable.

3.3 The effect of input rate estimation errors

The effects of errors in the estimation of the λ photon input rate are of special interest for photonic time-of-arrival based generators, especially considering that the method presented in Chapter 2 offers a way for compensating for most dead time like errors, creating vDTD distributions that follow the expected ideal case (albeit at a slight performance cost). Another important factor to consider is that an error in the estimation of the input rate causes e_i estimation errors mostly with the same sign, making it a major contributor to the accumulated $\left|\sum_{j=N_x+1}^{N_y} e_j\right|$ errors both in the case of using the distribution of single detection events or the joint distribution of multiple detections as input for the post-processing scheme.

3.3.1 Using single photon arrival time differences

To investigate the effect of error in the estimated rate, consider the case of estimating the actual $\lambda = \hat{\lambda} + \lambda_e$ input rate with $\hat{\lambda}$ estimation value and λ_e estimation error. Then,

$$p_n = \Pr(D = d_i = n) = \mathrm{e}^{-n(\hat{\lambda}+\lambda_e)\tau}\left(1 - \mathrm{e}^{-(\hat{\lambda}+\lambda_e)\tau}\right), \tag{3.59}$$

$$\hat{p_n} = \mathrm{e}^{-n\hat{\lambda}\tau}\left(1 - \mathrm{e}^{-\hat{\lambda}\tau}\right), \tag{3.60}$$

while

$$c_n = \sum_{j=0}^{n} \hat{p}_j = \sum_{j=0}^{n} \left(p_j - e_j\right) = 1 - \mathrm{e}^{-(n+1)\hat{\lambda}\tau}. \tag{3.61}$$

and the e_n errors can of course be calculated as $e_n = p_n - \hat{p_n}$. Using these results, the approximation errors can be calculated similarly to Sec. 3.2.1:

$$\begin{aligned}\Pr(C < y) &= \sum_{j=0}^{N_y} p_j = 1 - \mathrm{e}^{-(N_y+1)(\hat{\lambda}+\lambda_e)\tau} \\ &= c_{N_y} + \mathrm{e}^{-(N_y+1)\hat{\lambda}\tau}\left(1 - \mathrm{e}^{-(N_y+1)\lambda_e\tau}\right) = c_{N_y} + \sum_{j=0}^{N_y} e_j.\end{aligned} \tag{3.62}$$

Using (3.37), (3.38) and (3.40), various bounds for the approximation error of random bit generation can be given as:

$$\begin{aligned}
&\left|\Pr\left(\frac{l}{2^{m_e}} \le U < \frac{l+1}{2^{m_e}}\right) - \Pr(R = r_l)\right| < \max(c_{N_y+1} - c_{N_y}, c_{N_x+1} - c_{N_x}) + \left|\sum_{j=N_x+1}^{N_y} e_j\right| \\
&= \mathrm{e}^{\left\lceil \ln\left(1-\frac{l}{2^{m_e}}\right)\frac{1}{\lambda\tau}\right\rceil \lambda\tau}\left(1-\mathrm{e}^{-\lambda\tau}\right) + \left|\sum_{j=N_x+1}^{N_y} e_j\right| \le \max_i \hat{p}_i + \left|\sum_{j=N_x+1}^{N_y} e_j\right| \\
&= 1 - \mathrm{e}^{-\hat{\lambda}\tau} + \left|\left(\sum_{j=0}^{N_y} p_j - \sum_{j=0}^{N_x+1} p_j\right) - \left(\sum_{j=0}^{N_y} \hat{p}_j - \sum_{j=0}^{N_x+1} \hat{p}_j\right)\right| \\
&= 1 - \mathrm{e}^{-\hat{\lambda}\tau} + \left|\mathrm{e}^{-(N_y+1)\hat{\lambda}\tau} - \mathrm{e}^{-(N_y+1)(\hat{\lambda}+\lambda_e)\tau} + \mathrm{e}^{-(N_x+2)(\hat{\lambda}+\lambda_e)\tau} - \mathrm{e}^{-(N_x+2)\hat{\lambda}\tau}\right|,
\end{aligned} \tag{3.63}$$

where $N_x = \left\lfloor -\frac{\ln\left(1-\frac{l}{2^{m_e}}\right)}{\lambda\tau} - 1 \right\rfloor$ and $N_y = \left\lfloor -\frac{\ln\left(1-\frac{l+1}{2^{m_e}}\right)}{\lambda\tau} - 1 \right\rfloor$.

Total magnitude of errors

Knowing that the

$$e_n = p_n - \hat{p}_n = \mathrm{e}^{-n(\hat{\lambda}+\lambda_e)\tau}\left(1-\mathrm{e}^{-(\hat{\lambda}+\lambda_e)\tau}\right) - \mathrm{e}^{-n\hat{\lambda}\tau}\left(1-\mathrm{e}^{-\hat{\lambda}\tau}\right) = 0 \tag{3.64}$$

equation has only a singular solution for n in the form of

$$n_{e=0} = -\frac{\ln\left(\frac{1-\mathrm{e}^{-\hat{\lambda}\tau}}{1-\mathrm{e}^{-(\hat{\lambda}+\lambda_e)\tau}}\right)}{\lambda_e\tau}, \tag{3.65}$$

it can be seen that e_n values for $n < n_{e=0}$ have the same sign as λ_e, while e_n values for $n > n_{e=0}$ have a sign opposite of λ_e. Additionally considering that $\sum_n e_n = \sum_n p_n - \sum_n \hat{p}_n = 1 - 1 = 0$, means that

$$\sum_{n=0}^{\lfloor n_{e=0}\rfloor} e_n = -\sum_{n=\lceil n_{e=0}\rceil}^{\infty} e_n, \tag{3.66}$$

and therefore,

$$\begin{aligned}
\sum_n |e_n| &= 2\cdot\left|\sum_{n=0}^{\lfloor n_{e=0}\rfloor} e_n\right| = 2\cdot\left|\left(1-\mathrm{e}^{-(\lfloor n_{e=0}\rfloor+1)(\hat{\lambda}+\lambda_e)\tau}\right) - \left(1-\mathrm{e}^{-(\lfloor n_{e=0}\rfloor+1)\hat{\lambda}\tau}\right)\right| \\
&= 2\cdot\left|\left(1-\mathrm{e}^{-(\lfloor n_{e=0}\rfloor+1)\lambda_e\tau}\right)\mathrm{e}^{-(\lfloor n_{e=0}\rfloor+1)\hat{\lambda}\tau}\right| = \sum_i |e_i|.
\end{aligned} \tag{3.67}$$

Statistical distance

The statistical distance from the expected uniform output according to (3.42) can be written as:

$$\begin{aligned}\Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_e}} \le U < \frac{l+1}{2^{m_e}}\right) - \Pr(R=r_l)\right| \\ &< \frac{1}{2}\sum_l \left(e^{\left\lceil \ln\left(1-\frac{l}{2^{m_e}}\right)\frac{1}{\lambda\tau}\right\rceil \lambda\tau}\left(1-e^{-\lambda\tau}\right) + \left|\sum_{j=N_x+1}^{N_y} e_j\right|\right) \\ &\le \frac{1}{2}\sum_l e^{\left\lceil \ln\left(1-\frac{l}{2^{m_e}}\right)\frac{1}{\lambda\tau}\right\rceil \lambda\tau}\left(1-e^{-\lambda\tau}\right) + \frac{1}{2}\sum_i |e_i| \\ &\le \frac{1}{2}\sum_l \max_i \hat{p}_i + \frac{1}{2}\sum_i |e_i| = 2^{m_e-1}\left(1-e^{-\hat{\lambda}\tau}\right) + \frac{1}{2}\sum_i |e_i|.\end{aligned} \tag{3.68}$$

Notice that while the $\sum_i |e_i|$ error term (calculated in (3.67)) depends on the $\lambda\tau$ product as presented on Fig. 3.1, for the practically interesting regime of smaller $\lambda\tau$ values it is close to constant for the same λ_e/λ ratios.

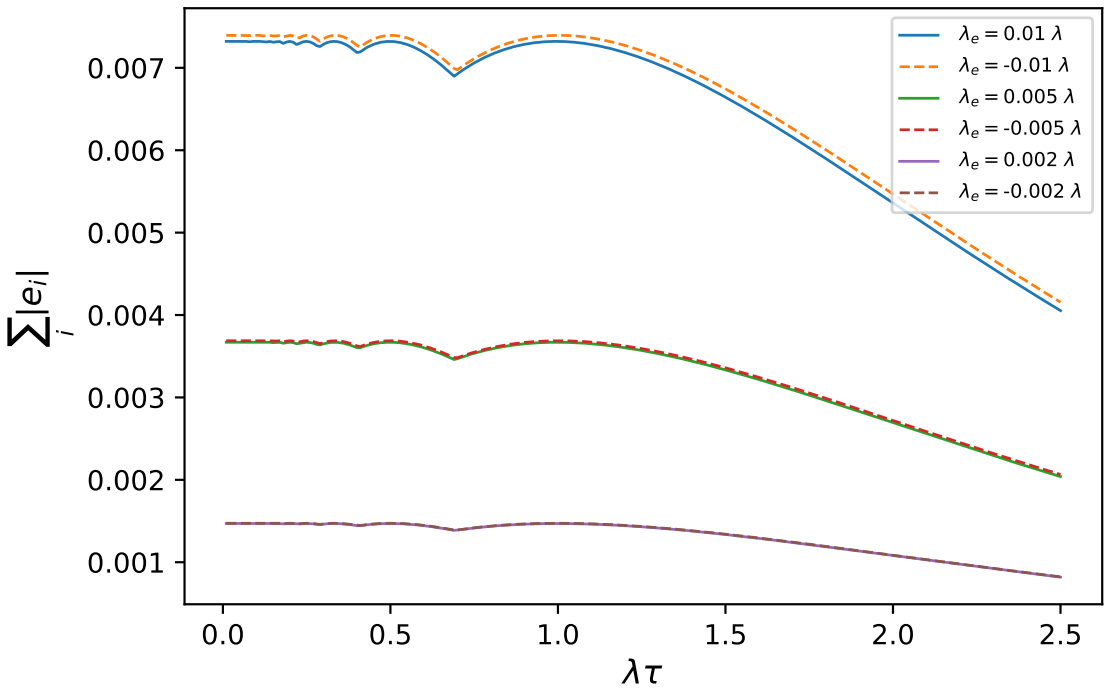

Figure 3.1: Effect of different $\lambda\tau$ parameters on the $\sum_i |e_i|$ error term. The slight "bumps" in the graph correspond to points where the integer part of $n_{e=0}$ decreases by one.

3.3.2 Using the joint distribution of multiple photon arrival time differences

Similarly to the case presented in Sec. 3.3.1, input rate estimation errors can be a major error factor in the case of using the joint distribution of measurement results, meaning that the presence of a $\lambda_e = \lambda - \hat{\lambda}$ rate estimation error of individual measurement samples can majorly contribute to the accumulated $\left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right|$ errors. The notable quantities in this case are: According to (3.59) and (3.60),

$$p_{q_\vartheta} = \Pr(D = q_\vartheta) = \mathrm{e}^{-q_\vartheta(\hat{\lambda}+\lambda_e)\tau}\left(1 - \mathrm{e}^{-(\hat{\lambda}+\lambda_e)\tau}\right), \tag{3.69}$$

$$\hat{p}_{q_\vartheta} = \mathrm{e}^{-q_\vartheta\hat{\lambda}\tau}\left(1 - \mathrm{e}^{-\hat{\lambda}\tau}\right), \tag{3.70}$$

giving

$$\underline{p}_i = \prod_{\vartheta=0}^{Q-1} p_{q_\vartheta} = \left(1 - \mathrm{e}^{-(\hat{\lambda}+\lambda_e)\tau}\right)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_\vartheta(\hat{\lambda}+\lambda_e)\tau}, \tag{3.71}$$

$$\underline{\hat{p}}_i = \prod_{\vartheta=0}^{Q-1} \hat{p}_{q_\vartheta} = \left(1 - \mathrm{e}^{-\hat{\lambda}\tau}\right)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_\vartheta\hat{\lambda}\tau}, \tag{3.72}$$

$$\underline{e}_i = \underline{p}_i - \underline{\hat{p}}_i, \tag{3.73}$$

and

$$\underline{c}_i = t(\underline{d}_i) = \sum_{j=0}^{i} \underline{\hat{p}}_j = \sum_{j=0}^{i} \left(\left(1 - \mathrm{e}^{-\hat{\lambda}\tau}\right)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_{\vartheta,j}\hat{\lambda}\tau}\right) = \sum_{j=0}^{i} \left(\underline{p}_j - \underline{e}_j\right). \tag{3.74}$$

The upper bound of approximation error is the same as in (3.47):

$$\begin{aligned} &\left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R = r_l)\right| \\ &< \left|c_{N_y+1} - c_{N_y} - (c_{N_x+1} - c_{N_x}) + \sum_{j=N_x+1}^{N_y} \underline{e}_j\right| \le \max_i \underline{\hat{p}}_i + \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right|, \end{aligned} \tag{3.75}$$

while N_y and N_x can be calculated similarly to (3.31) and (3.32) with the only difference of using the estimated probabilities in $t^{-1}(\cdot)$, giving

$$N_y \le t^{-1}(y) \Rightarrow N_y = \left\lfloor t^{-1}(y) \right\rfloor, \tag{3.76}$$

as well as,

$$c_{N_y+1} - c_{N_y} = \sum_{j=0}^{N_y+1} \underline{\hat{p}}_j - \sum_{j=0}^{N_y} \underline{\hat{p}}_j = \underline{\hat{p}}_{N_y+1} = \left(1 - \mathrm{e}^{-\hat{\lambda}\tau}\right)^Q \prod_{\vartheta=0}^{Q-1} \mathrm{e}^{-q_{\vartheta,N_y+1}\hat{\lambda}\tau}. \tag{3.77}$$

The upper bound for the statistical distance is:

$$\begin{aligned}
\Delta(U,R) &= \frac{1}{2}\sum_l \left|\Pr\left(\frac{l}{2^{m_\mathrm{e}}} \le U < \frac{l+1}{2^{m_\mathrm{e}}}\right) - \Pr(R=r_l)\right| \\
&< \frac{1}{2}\sum_l \left|\max(\underline{p}_{N_y+1}, \underline{p}_{N_x+1})\right| + \frac{1}{2}\sum_l \left|\sum_{j=N_x+1}^{N_y} \underline{e}_j\right| = \frac{1}{2}\sum_l \left|\max(\underline{p}_{N_y+1}, \underline{p}_{N_x+1})\right| + \frac{1}{2}\sum_i |\underline{e}_i| \\
&\le 2^{m_\mathrm{e}-1} \max_i \hat{\underline{p}}_i + \frac{1}{2}\sum_i |\underline{e}_i| = 2^{m_\mathrm{e}-1}\left(1-\mathrm{e}^{-\hat{\lambda\tau}}\right)^Q + \frac{1}{2}\sum_i |\underline{e}_i| \\
&= 2^{m_\mathrm{e}-1}\left(1-\mathrm{e}^{-\hat{\lambda\tau}}\right)^Q + \frac{1}{2}\left(\left(\sum_i |e_i| + 1\right)^Q - 1\right).
\end{aligned} \tag{3.78}$$

Note that according to (3.53) the $\sum_i |\underline{e}_i|$ error term grows when using the joint distribution of multiple DTDs compared to the single sample case. Due to this, using the joint distribution of more DTDs is only beneficial up to a certain point, from where the magnitude of the $\Delta(U,R)$ statistical distance becomes dominated by the error term.

Using the general results of Sec. 3.2.4, the Q_opt value corresponding to the minimal achievable statistical distance can be calculated as:

$$Q_\mathrm{opt} = -\frac{\log\left(-\frac{2^{m_\mathrm{e}}\log\left(1-\mathrm{e}^{-\hat{\lambda\tau}}\right)}{\log\left(\sum_i |e_i|+1\right)}\right)}{\log\left(1-\mathrm{e}^{-\hat{\lambda\tau}}\right) - \log\left(\sum_i |e_i|+1\right)}. \tag{3.79}$$

Naturally, the optimal integer choice for Q will either be $\lfloor Q_\mathrm{opt} \rfloor$ or $\lceil Q_\mathrm{opt} \rceil$.

3.4 A practical bit generation scheme

Since exponentially distributed random variables can take infinitely many values, in practice, it may be beneficial to limit the possible number of outcomes. This way, the number of possible d_i input outcomes is limited and may be easier to manage, which also means a limited number of $i = t_Q(\underline{q})$ indexes, too. Consider the following modified distribution with N_O finite number of outcomes for the individual measurement samples:

$$p_i = \begin{cases} \mathrm{e}^{-i\lambda\tau}\left(1-\mathrm{e}^{-\lambda\tau}\right) = \mathrm{e}^{-n\lambda\tau}\left(1-\mathrm{e}^{-\lambda\tau}\right) = p_n & \text{if } n < N_O - 1, \\ 1 - \sum_{j=0}^{N_O-2} \mathrm{e}^{-j\lambda\tau}\left(1-\mathrm{e}^{-\lambda\tau}\right) = \mathrm{e}^{-\lambda(N_O-1)\tau} = p_{N_O-1} & \text{if } n \ge N_O - 1. \end{cases} \tag{3.80}$$

Notice, that in the case of $\hat{\lambda}$ estimated rate with λ_e rate estimation error, the error terms corresponding to the modified part of $n \geq N_O - 1$ and the original exponential $n < N_O - 1$ part have the same sign if

$$N_O \geq n_{e=0} = -\frac{\ln\left(\frac{1-\mathrm{e}^{-\hat{\lambda}\tau}}{1-\mathrm{e}^{-(\hat{\lambda}+\lambda_e)\tau}}\right)}{\lambda_e \tau}, \tag{3.81}$$

therefore (3.67) holds if this criterion for N_O is satisfied.
For i index assignment, use the following rule:

$$i = t_Q(\overline{q}) = N_O^0 q_0 + N_O^1 q_1 + \ldots + N_O^{Q-1} q_{Q-1} = \sum_{\vartheta=0}^{Q-1} N_O^\vartheta q_\vartheta. \tag{3.82}$$

Then, the individual q_ϑ outcomes can be calculated from i as

$$q_\vartheta = \left\lfloor \left(i \mod N_O^{\vartheta+1} \right) / N_O^\vartheta \right\rfloor. \tag{3.83}$$

Using this and the result of Sec. A.5 the $\underline{c_i}$ values are:

$$c_i = \sum_{j=0}^{i} \underline{p}_j = \sum_{j=0}^{i} \prod_{\vartheta=0}^{Q-1} p_{q_{\vartheta,j}} = \sum_{\vartheta=0}^{Q-1} \left(\prod_{k=\vartheta+1}^{Q-1} p_{q_k} \sum_{j=0}^{q_\vartheta - 1} p_j \right) + \prod_{k=0}^{Q-1} p_{q_k}. \tag{3.84}$$

Additionally, the $\sum_{j=0}^{q_\vartheta-1} p_j$ term can be further simplified as:

$$\sum_{j=0}^{q_\vartheta-1} p_j = \begin{cases} \left(1-\mathrm{e}^{-\lambda\tau}\right) \frac{1-\mathrm{e}^{-q_\vartheta\lambda\tau}}{1-\mathrm{e}^{-\lambda\tau}} = 1-\mathrm{e}^{-q_\vartheta\lambda\tau} & \text{if } q_\vartheta < N_O - 1, \\ 1 & \text{if } q_\vartheta \geq N_O - 1. \end{cases} \tag{3.85}$$

A practical scheme for bit generation can then be constructed according to the steps of Sec. 3.1.3. Notice that since the target output is an m_e-bit long binary string, after calculating c_i, the r_l output value assignment can be realized by simply taking the most significant m_e bits of the binary representation of c_i.
Note that the results of Sec. 3.3.1 and Sec. 3.3.2 for upper bounds on estimation error and statistical distance hold for this practical bit generation scheme as long as $p_0 = (1-\mathrm{e}^{-\lambda\tau}) \geq \mathrm{e}^{-\lambda(N_O-1)\tau} = p_{N_O-1}$, giving the criterion of:

$$N_O \geq 1 - \frac{\ln\left(1-\mathrm{e}^{-\lambda\tau}\right)}{\lambda\tau}. \tag{3.86}$$

3.5 Experimental results

To verify correctness and investigate applicability in a real-world scenario, I implemented the practical bit generation scheme presented in the previous section. I used the overestimated unbinned dataset of Sec. 2.5.2 as input and generated output bits with the scheme presented in Sec. 3.4, then tested the quality of this output with the statistical test suites presented in Sec. 1.5.2.

3.5.1 Input parameters

The input dataset contains $1.556 \cdot 10^9$ vDTDs with an average $\lambda\tau = 2.6388 \cdot 10^{-4}$ value. Unfortunately, the input vDTD sequence exhibited a slow, small drift in this $\lambda\tau$ value; therefore, the previously presented average cannot be considered constant. To account for this, I used estimated $\lambda\tau$ values of smaller subsequences corresponding to smaller time segments, where the effect of the previously observed drift is negligible. Each subsequence consisted of 10 million samples, with estimated $\lambda\tau$ values ranging from $2.6244 \cdot 10^{-4}$ to $2.6608 \cdot 10^{-4}$. I calculated λ_e estimation errors by calculating the half-width of the 99% confidence interval for the average of $\lambda\tau$ based on 10^7 samples, giving:

$$\lambda_\mathrm{e} = \frac{\sqrt{2} \cdot \mathrm{Erf}^{-1}(0.99)}{\sqrt{2 \cdot 10^7}} \cdot \lambda\tau = 8.145 \cdot 10^{-4} \cdot \lambda\tau.$$

Parameters for the bit generation scheme

For the choice of $m_\mathrm{e} = 10$, $Q = 2$ is optimal according to Sec. 3.2.4, resulting in a $\Delta(U, R) < 2^{-10.62}$ bound for the output sequence. I chose this particular value for m_e as it offers a good output bit generation efficiency of $m_\mathrm{e}/Q = 5$ output bits generated per input sample while offering a $\Delta(U, R)$ that is not far from the minimum achievable $\Delta(U, R)_\mathrm{min} < 2^{-10.79}$ (at $m_\mathrm{e} = 1$, $Q = 1$). The criteria of (3.81) states that $N_O \geq 3756.238$, while according to (3.86) $N_O \geq 30938.487$, therefore, my choice of $N_O = 2^{16} = 65536$ satisfies both of these. In addition to the statistical distance, a bound for bit bias can also be calculated, which is:

$$d(B_i) = \left| \Pr(B_i = x_i) - \frac{1}{2} \right| < 2^{m_\mathrm{e}-1} \max_i \hat{p}_i + 2^{m_\mathrm{e}-1} \max_l \left| \sum_{j=N_x+1}^{N_y} e_j \right| = 2^{-8.45}. \quad (3.87)$$

Note that these bounds for the bit bias and statistical distance are forgiving compared to what is usually recommended for cryptographic applications, and due to this, well-designed statistical testing tools are likely to detect such large deviations from uniformity. On the other hand, actual output data may approximate uniformity better with better statistical properties than what the calculated bounds suggest, as the bounds correspond to safe overestimations of worst-case scenarios.
Using these parameters, I generated $7.78 \cdot 10^9$ output bits from $1.556 \cdot 10^9$ input samples, resulting in a total bit generation efficiency of 5 bits/sample.

3.5.2 Statistical testing results

I used the suites presented in Sec 1.5.2 with default parameters for statistical testing (Same parameter set as in Sec. 2.5.3). While I only present a summary of the results

here, the exact outputs of the statistical testing suites used are available in a github repository at `https://github.com/S0lymi/dissertaion_results`. The results indicate that the generated bit sequence is close to uniform but with small deviations. From the NIST STS, the generated sequence failed on 3 of the statistical tests from the 188 total, namely the bit frequency and two of the cumulative sum test cases. It also failed one test from TestU01's Rabbit battery. The fact that the output passed all other tests from the NIST STS, the Alphabit, and SmallCrush batteries and showed good statistical properties when evaluated with ENT supports its closeness to uniformity. Due to the limited length of the tested output compared to the input length requirements of TestU01's Crush battery and Dieharder, these suites could not be run.[14]

3.5.3 Pairing with other post-processing techniques

Besides improving the measurement setup used, combining the scheme with other post-processing techniques can also offer a way to improve the quality of the output bitstream. For a simple example, I utilized the simple XOR extractor [119] to improve output quality. In this case, the bound on bit bias presented previously can be used to give a bound on the bit bias of the output bitstream of the extractor, giving:

$$d_{xor}(B_i) = 2 \cdot d(B_i)^2 = 2^{-15.9}. \tag{3.88}$$

Thus, the quality of the output sequence can be majorly improved at the cost of bit generation efficiency dropping to 2.5 output bits generated per input sample since the XOR extractor halves the number of output bits.[15] The quality improvement is supported by the fact that the output bits of this scenario successfully passed all applicable statistical tests (The Dieharder, SmallCrush, and Crush batteries could not be run due to limited input length).

3.6 Summary of the results

I proposed a post-processing scheme inspired by the continuous probability integral transform for quantum random number generators. The scheme can transform an input measurement distribution into an output distribution that is ideally arbitrarily close to uniform, with advantageous properties regarding potentially achievable

[14] In the case of Dieharder the suite can complete with rewounding the file multiple times, thus potentially skewing the results.

[15] This output efficiency is still more than five times higher than what is achievable with the simple method presented in [80] and used in Sec. 2.5.3.

bit generation efficiency. I have given bounds for the deviation of the output from uniformity both in the ideal well-characterized scenario and with characterization errors present. I showed that in the presence of errors, the output's deviation from uniformity can no longer be made arbitrarily small, and I deduced a formula for finding the optimal parameters for minimizing statistical distance. Using my general theoretical results, I thoroughly investigated the particular case of using the scheme with photonic time-of-arrival based QRNGs and defined a simple practical bit generation method that I then used to generate output bit sequences from previous experimental measurement data. I further verified my theoretical results with statistical testing done on this output. Results presented in the chapter form the basis of Thesis II.

While the proposed scheme has advantageous properties in terms of potential efficiency and computational complexity, its sensitivity to systematic errors can be a drawback. Still, it might offer a valuable post-processing tool, especially for physical setups in well-controlled environments. Additionally, being able to give well-defined upper bounds for the deviations of the output also enables easy integration with other post-processing techniques.

Since the used bit generation and index assignment scheme also play an important part in the resulting output deviation, investigating ways to tailor bit generation to mitigate specific characteristic errors presents a potential future research direction that may further expand useability. Alternatively, one can aim to exploit the scheme's error sensitivity; for example, pairing it with common statistical testing practices may present an interesting future research case for increasing the flexibility of the current testing tools regarding possible input distributions.

Chapter 4

Efficiency and quality improvement of time-of-arrival quantum random number generators with hashing

In this chapter, I present a post-processing scheme utilizing min-entropy estimation and hashing for photonic time-of-arrival based QRNGs. The main idea of my method is that knowing the min-entropy of the QRNG output allows for the parametrization of universal hash functions as secure randomness extractors. Utilizing the mathematical model presented in Sec. 2.2, I first calculate a lower bound for the min-entropy of DTDs measured with a continuously running measurement clock in the presence of detector dead time. Then, I show ways to handle the practical case of not exactly known input photon rate and detector dead time. I also consider the potential effects of additive noise and give a corrected secure lower bound for min-entropy. I then parametrize and implement a Toeplitz hash based extractor using my previous results and use it to generate random bits from measurement data collected with a purposely non-ideal physical measurement setup. This way, I show an example case of using the theoretical results in a practical scenario while also experimentally validating their correctness.

4.1 Hash functions as randomness extractors

A $(k_\mathrm{e}, \epsilon, n_\mathrm{e}, d_\mathrm{e}, m_\mathrm{e})$ randomness extractor is an $\mathrm{Ext} : \{0,1\}^{n_\mathrm{e}} \times \{0,1\}^{d_\mathrm{e}} \mapsto \{0,1\}^{m_\mathrm{e}}$ function that can take the n_e bit long output of a weak entropy source, coupled with a d_e bit long uniform seed as input and create an m_e bit long output sequence with a distribution that is ϵ-close statistically to the goal uniform distribution, provided that the input probability distribution D on $\{0,1\}^{n_\mathrm{e}}$ has at least $H_\infty(D) \geq k_\mathrm{e}$ bits of

min-entropy. As presented in Sec. 3.1.3, the goal of post-processing is to create close to uniformly distributed m_e bit-long output sequences. Since the binary representations of the physical measurement results can also be considered as outputs from a weak entropy source, randomness extractors can be good candidates for post-processing solutions for quantum random number generators. Note that extractors require a uniformly random U_{d_e} seed of d_e bits to function. The reusability of this seed is crucial when working with RNGs since otherwise, the randomness needed for constant reseeding of the extractor may exceed the randomness output of the RNG. An extractor is called strong if concatenating the seed with the extractor's output yields a distribution that is still ϵ-close to uniform, which also means that the seed is reusable; therefore, strong extractors are desired when working with RNGs. Since universal hash functions are proven to be strong extractors by the Leftover Hash Lemma [120], I chose the popular Topelitz hash to serve as a basis for the post-processing method I present in this chapter.

4.1.1 Parameters of the Toeplitz hash based extractor

A Toeplitz hash based randomness extractor operates by multiplying the n_e bit long input with a random binary Toeplitz matrix of size $n_e \times m_e$ to produce an m_e bit long output at each operation step. Additionally, the k_e extractable entropy contained in the n_e long input has the following relationship with the m_e output length:

$$m_e = k_e - 2\log\epsilon. \tag{4.1}$$

Since I obtain the n_e input bits by simply concatenating multiple measurement records,

$$n_e = [\text{number of records}] \cdot [\text{length of a record}], \tag{4.2}$$

while

$$k_e \leq [\text{number of records}] \cdot H_\infty(D), \tag{4.3}$$

where $H_\infty(D)$ is the min-entropy of a single measurement record.
This means that for a given goal ϵ, after choosing the number of records to concatenate, all other extractor parameters can be straightforwardly calculated if $H_\infty(D)$ is known. Notice that if $H_\infty(D)$ is not exactly known, but a $\underline{H_\infty(D)}$ lower bound for the min-entropy is available if an extractor is parametrized according to this lower bound, the resulting $\underline{\epsilon}$ value is an upper bound on the original goal ϵ. Therefore, the lower bound of min-entropy can also be used to meet output quality goals in the form of $\Delta \leq \epsilon_{\text{goal}}$. Due to this, in the following, my main goal is to derive a $\underline{H_\infty(D)}$ lower bound for the min-entropy of time-of-arrival QRNGs that then can be used for extractor parametrization.

4.2 The lower bound for min-entropy in ToA QRNGs

The min-entropy of individual measurement results can be calculated based on the mathematical model of QRNG operation with a continuous clock presented in Sec. 2.2.

4.2.1 Ideal case

In the ideal case of no dead time, the conditional distribution of the DTDs, which are the measurement results, can be calculated from (2.4), giving:

$$\begin{aligned} p_n &= \Pr\big(D = n \mid \gamma = y\big) \\ &= \begin{cases} \Pr(y + T < \tau) & \text{if } n = 0, \\ \Pr\big(n\tau \le y + T < (n+1)\tau\big) & \text{if } n > 0, \end{cases} \\ &= \begin{cases} 1 - \mathrm{e}^{-\lambda(\tau - y)} & \text{if } n = 0, \\ \big(1 - \mathrm{e}^{-\lambda\tau}\big)\,\mathrm{e}^{-\lambda(n\tau - y)} & \text{if } n > 0. \end{cases} \end{aligned} \tag{4.4}$$

To calculate worst-case min-entropy, maximize p_n:

$$\begin{aligned} &\max_{n,y}\big(\Pr\big(D = n \mid \gamma = y\big)\big) \\ &\quad= \max_{n,y} \begin{cases} 1 - \mathrm{e}^{-\lambda(\tau - y)} & \text{if } n = 0, \\ \mathrm{e}^{\lambda y}\left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda n\tau} & \text{if } n > 0, \end{cases} \\ &\quad= \max_{n,y} \begin{cases} \left(1 - \mathrm{e}^{-\lambda\tau}\right) & \text{if } n = 0, y \to 0, \\ \mathrm{e}^{\lambda\tau}\left(1 - \mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda n\tau} & \text{if } n > 0, y \to \tau, \end{cases} \\ &\quad= \max_{n,y} \begin{cases} 1 - \mathrm{e}^{-\lambda\tau} & \text{if } n = 0, y \to 0, \\ 1 - \mathrm{e}^{-\lambda\tau} & \text{if } n = 1, y \to \tau, \end{cases} \\ &\quad= 1 - \mathrm{e}^{-\lambda\tau}, \end{aligned} \tag{4.5}$$

so then the min-entropy is:

$$H_\infty(D) = -\log_2\left(\max_{n,y} p_n\right) = -\log_2\left(1 - \mathrm{e}^{-\lambda\tau}\right). \tag{4.6}$$

4.2.2 Case of nonzero dead time

Similarly to the previous case, the min-entropy of DTDs when dead time is present can be calculated from (2.14), where

$$
\begin{aligned}
p_n &= \Pr\big(D = n \mid \gamma = y\big) \\
&= \begin{cases}
0 \text{ if } n < k, \\
\Pr(y + T + \delta < \tau) \text{ if } n = k, \\
\Pr\left((n-k)\tau \le y + T + \delta < (n-k+1)\tau\right) \text{ if } n = k+1, \\
\Pr\left((n-k)\tau \le y + T + \delta < (n-k+1)\tau\right) \text{ if } n > k+1,
\end{cases} \\
&= \begin{cases}
0 \text{ if } n < k, \\
\begin{cases}
1 - \mathrm{e}^{-\lambda(\tau - y - \delta)} & \text{if } y < \tau - \delta, n = k, \\
0 & \text{if } y \ge \tau - \delta, n = k,
\end{cases} \\
\begin{cases}
\mathrm{e}^{-\lambda(\tau - y - \delta)}\left(1 - \mathrm{e}^{-\lambda\tau}\right) & \text{if } y < \tau - \delta, n = k+1, \\
1 - \mathrm{e}^{-\lambda(2\tau - y - \delta)} & \text{if } y \ge \tau - \delta, n = k+1,
\end{cases} \\
\left(\mathrm{e}^{-\lambda((n-k)\tau - y - \delta)}\right)\left(1 - \mathrm{e}^{-\lambda\tau}\right) \text{ if } n > k+1.
\end{cases}
\end{aligned} \tag{4.7}
$$

To calculate the worst case min-entropy, I maximize p_n just like before, only now maximizing according to possible ζ dead time values too:

$$
H_\infty(D) = -\log_2 \max_{n,y,\zeta}(\Pr\big(D = n \mid \gamma = y\big)) = -\log_2\left(1 - \mathrm{e}^{-\lambda\tau}\right). \tag{4.8}
$$

Note that this result is the same as in the case without dead time (ideal case), due to the maximization reaching its maximum at the zero fractional dead time point since the achievable maximum for p_n is the probability corresponding to the case of detection in the first window when detector sensitivity (previous detection or end of dead time) starts at the beginning of the window. Also, note that this resulting worst-case min-entropy is the same as the min-entropy of DTDs measured with an ideal restartable clock with no dead time [113]. Additionally, note that in this calculation of $H_\infty(D)$ ζ is not restricted in any way as in (4.8) p_n is maximized over all possible ζ. Due to this, $H_\infty(D) = -\log_2\left(1 - \mathrm{e}^{-\lambda\tau}\right) \le H_\infty(D \mid \zeta = Z)$ will hold for any possible Z distribution of ζ.

4.2.3 Accounting for estimation errors of the input photon rate

Both the dead time of the detection system and other physical imperfections can change the value of the detected λ_d photon rate. Since in (4.8) $H_\infty(D)$ is a function of the actual true λ input photon rate, the potential difference of λ_d from λ must be handled. The effect of a known constant ζ dead time on the detected photon rate is

given in (2.12), from which the actual true λ input photon rate is:

$$\lambda = \frac{\lambda_{\mathrm{d}}}{1 - \lambda_{\mathrm{d}}\zeta}. \tag{4.9}$$

In a practical scenario, however, the exact value of the ζ dead time might not be known, or its value might not be constant. Other imperfections may also influence the actual true λ rate. In the following, I present a way to give a lower bound for $H_\infty(D)$ even in the presence of such imperfections, by showing that λ and therefore $H_\infty(D)$ is monotonic in both ζ and λ_{d}.

Monotonicity of the worst-case min-entropy in the photon rate

The $H_\infty(D)$ worst-case min-entropy is monotonically decreasing in λ, since

$$\frac{\partial H_\infty(D)}{\partial \lambda} = -\frac{1}{\log(2)} \frac{\tau}{1 - \mathrm{e}^{-\lambda\tau}} < 0 \tag{4.10}$$

for all $\lambda > 0, \tau > 0$. This also means that if we can give a $\lambda_{\max}$ upper bound for the actual true input λ photon rate, a $H_\infty(D)_{\mathrm{L}}$ lower bound for the min-entropy can be calculated such that

$$H_\infty(D)_{\mathrm{L}} = -\log_2\left(1 - \mathrm{e}^{-\lambda_{\max}\tau}\right). \tag{4.11}$$

Note that $H_\infty(D)$ is similarly monotonic in τ, so possible uncertainty in τ can also be addressed in a similar manner by giving a $\tau_{\max}$ upper bound.[16]

Monotonicity in dead time

Notice that (4.9) is monotonically increasing in both λ_{d} and ζ for the physically relevant possible values[17] of $0 < \lambda_{\mathrm{d}} < 1/\zeta$, $0 < \zeta$. Due to this, an upper bound of λ_{d} also upper bounds λ, thus lower bounding $H_\infty(D)$. Similarly, if an upper bound for the ζ dead time is known, it can be used to give a safe lower bound on the min-entropy. Using (4.11) and (4.9) with $\lambda_{\mathrm{d,max}}$ and $\zeta_{\max}$ upper bound for λ_{d} and ζ, the following $H_\infty(D)_{\mathrm{L}}$ lower bound can be calculated for the min-entropy:

$$H_\infty(D)_{\mathrm{L}} = -\log_2\left(1 - \mathrm{e}^{-\frac{\lambda_{\mathrm{d,max}}\tau}{1-\lambda_{\mathrm{d,max}}\zeta_{\max}}}\right). \tag{4.12}$$

A $p_{\max}$ upper bound for the probability of the most likely measurement result can also be given as

$$p_{\max} = 1 - \mathrm{e}^{-\frac{\lambda_{\mathrm{d,max}}\tau}{1-\lambda_{\mathrm{d,max}}\zeta_{\max}}}. \tag{4.13}$$

[16] For most physical setups, though, τ is assumed to be a known constant, as this parameter is typically known with a much better error margin than λ.

[17] Note, that λ_{d} is maximized in $1/\zeta$, so the nominator in (4.9) always stays positive.

4.2.4 Accounting for additive noise

Although generally coherent light sources based on stimulated emission, like lasers, are assumed to be Poissonian photon sources due to the underlying physical working principle, in reality, photons from spontaneous emission (e.g., thermal effects) may also have a small superpoissonian contribution to the output distribution of the source (though this effect has been shown to be vanishing with increasing attenuation [107]). This unwanted process can be modeled by introducing additional photon counts mixed with the ideal exponential statistics. This idea of modeling the source as an idealized exponential mixed with various additive error sources can also be used to model various afterpulsing effects of the detector system or even a potential attacker.

Assume that our count statistic is made up of photons coming from an underlying true exponential source with C_{exp} number of independent detection events for a given time period, responsible for the majority of the total photon counts, and a smaller at most C_{noise} amount of photon counts coming from noise processes or even potential attackers. This essentially means a limit on noise/attacker intensity while assuming that an attacker is not capable of influencing photons from the trusted exponential photon source.[18] Additionally, denote the probability of the most probable measurement outcome from the trusted ideal exponential source with $p_{\max}$ The goal is to give a worst-case lower estimate for min-entropy. For this, I propose the following:

Theorem 3. *There exists a DTD series in a $C_{total} = C_{exp} + C_{noise}$ record long $\mathbb{D} = \{ D_1, D_2, \ldots, D_{C_{total}} \}$ joint ideal exponential and noise photon DTD series for which*

$$\underline{H_\infty(\mathbb{D})} = -(C_{exp} - C_{noise}) \log_2 p'_{max}$$
$$= \begin{cases} -(C_{exp} - C_{noise}) \log_2 \left(\frac{p_{max} C_{exp}}{C_{exp} - C_{noise}} \right) & \text{if } p_{max} C_{exp} < C_{exp} - C_{noise}, \\ 0 & \text{if } p_{max} C_{exp} \geq C_{exp} - C_{noise}, \end{cases} \tag{4.14}$$

is a lower bound in min-entropy, where p_{max} is the probability of the most likely outcome of the ideal exponential distribution.

Proof. Let $S_0, S_1, ..., S_i, ..., S_{C_{\text{exp}}}$ be the arrival times of photons from the trusted ideal exponential source, with $D_1^{\text{id}}, D_2^{\text{id}}, ..., D_i^{\text{id}}, ..., D_{C_{\text{exp}}}^{\text{id}}$ measured DTDs between them and note arrival times of noise/attacker photons with $N_1, ..., N_i, ..., N_{C_{\text{noise}}}$. Since the noise/attacker photons are allowed to have any distribution and use

[18] This also means that quantum operations, like entangling additional photons with photons from the trusted source, are not allowed either.

any strategy, even allowing dependence on other photon detections, let us not consider the min-entropy contribution of intervals where noise counts are involved (see Fig. 4.1), only the entropy contribution of intervals from the sub-series $\{D_j\}_{j\in J}$, where $J = \{j \mid \nexists N_i : S_{j-1} < N_i < S_j\}$ (the intervals not affected by noise counts). This leaves a total of $C_{\text{exp}} - C_{\text{noise}}$ intervals of the ideal source considered as contributing to min-entropy from the whole $\mathbb{D}$ series.

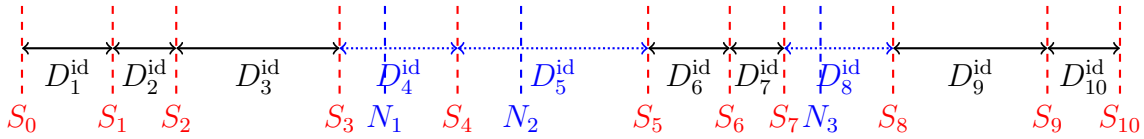

Figure 4.1: Example of handling additive noise. Times noted with S_i are counts from the assumed underlying ideal distribution, with $\{D_i^{\text{id}}$ intervals between them. After introducing N_i noise counts, we only consider the entropy contribution of intervals not affected by the noise, which are $\{D_1^{\text{id}}, D_2^{\text{id}}, D_3^{\text{id}}, D_6^{\text{id}}, D_7^{\text{id}}, D_9^{\text{id}}, D_{10}^{\text{id}}\}$ in this pictured example case.

Photons from noise/attacker can also have a distorting effect on the overall measured distribution and, therefore, the distribution of the remaining considered series. From the point of min-entropy, this means the possible change of the original p_{max} probability to a new p'_{max} value (change in the probability of the most frequent result). Assuming a worst-case scenario, additional noise counts can have the following effect:[19]

- Noise is positioned so that all original counts corresponding to p_{max} (originally most likely outcome of the ideal source) are kept in the considered sub-series, only DTDs with other values are dropped, thus increasing the probability of the most likely outcome to $p'_{\text{max}} = \frac{p_{\text{max}} C_{\text{exp}}}{C_{\text{exp}} - C_{\text{noise}}}$ and reducing min-entropy.

This way, (4.14) is a lower bound in min-entropy for the considered sub-series. Therefore, it is a valid lower bound for the whole series, too. □

4.2.5 Framework for extractor parameter selection

To give a combined min-entropy lower bound for a case containing all the previously investigated noise effects, the following steps can be used:

1. Select upper bounds $\lambda_{\text{d,max}}$ and ζ_{max} for λ_{d} and ζ and calculate p_{max} according to (4.13).

[19] While the actual physical feasibility of this worst-case scenario may at times be questionable, I still consider it to give a safe lower estimate.

2. Determine C_{exp} ideal exponential count number, and C_{noise} maximum contained noise count number for the $\mathbb{D}$ input DTD series corresponding to the n_{e} bit long extractor input.

3. Use (4.14) to calculate the final $\underline{H_{\infty}(\mathbb{D})}$ for the input $\mathbb{D}$ DTD series.

4. Calculate the rest of the extractor parameters according to Sec. 4.1.1.

Note that in the third step, a $p_{\max}$ upper bound of the probability of the most likely outcome of the intervals corresponding to C_{exp} is used. The reasoning presented in Sec. 4.2.4 is still valid as potential dependence between intervals due to non-ideal effects during measurement of photons of the ideal source is already accounted for in $p_{\max}$ (see Sec. 4.2), therefore, the ability to sum interval min-entropies in (4.14) remains.
Also note that counts from additive noise sources raise the experimentally detected λ_{d}, but this does not lead to any additional security weaknesses and can be handled as presented in Sec. 4.2.3.
While the min-entropy is determined by the physical setup, the other defining extractor parameter, n_{e}, can be chosen freely. Generally, choosing n_{e} to be larger is advantageous, as the scheme becomes more robust against bursty noise, as well as providing a better output ratio for a given ϵ according to (4.1). This comes at a cost of increased computational need, however.

4.3 Experimental results

4.3.1 Parameter selection

To acquire real-world measurement results, I used the same physical setup as the one presented in Sec. 2.5.1. I collected and processed $2 \cdot 10^{10}$ intervals to experimentally verify the validity of the presented framework. During data acquisition, the measured detection rate was around $\lambda_{\text{d}} = 1.3 \cdot 10^6\,\text{cps}$ (counts per second) and between $\lambda_{\text{d,min}} = 1.08 \cdot 10^6\,\text{cps}$ and $\lambda_{\text{d,max}} = 1.37 \cdot 10^6\,\text{cps}$ at all times. I chose not to try mitigating this fluctuation as my goal was to show robustness. I also did not use any of the available protective covers made for severely limiting counts from the environment, leading to an average $\lambda_{\text{n}} = 20000\,\text{cps}$ noise rate at the detector. According to the results of Sec. 2.5, I chose an overly safe upper bound for the detector dead time of $\tau_{\text{d}} = 1000\tau$. In practice, afterpulsing effects are often neutralized by the longer detector dead times compared to them, which is also the case for our hardware, showing negligible afterpulsing probability. To present an example of handling this

effect when calculating extractor parameters, I assume a maximum probability for afterpulsing of $P_{\text{after}} = 10^{-4}$ nonetheless. Due to the quality of laser sources, another quantity often considered experimentally negligible is the number of photons created in the source, not via stimulated emission. Similarly to the previous case of afterpulsing, this effect could be considered negligible in our setup, but I still assume an exemplary maximum probability for it to be $P_{\text{nonstim}} = 10^{-6}$. Naturally, after choosing a concrete n_{e}, other noise factors can also be considered (such as the potential for bursts in intensity of the noise/attacker photons), which can further influence the final choice of C_{noise} and C_{exp}.
Using the steps of Sec. 4.2.5, relevant measurement and extractor parameters in my case are the following:

- Using $\lambda_{\text{d,max}} = 1.37 \cdot 10^6$ cps and $\tau_{\text{d,max}} = 1000\tau = 250$ ns, gives an upper bound of $\lambda = 2.08 \cdot 10^6$ cps and $p_{\text{max}} = 5.2 \cdot 10^{-4}$ according to Sec. 4.2.3.

- To underestimate the rate at which the ideal exponentially distributed photons corresponding C_{exp} arrive, I calculate $\lambda_{\text{id}} = \lambda_{\text{min}}(1 - P_{\text{nonstim}})(1 - P_{\text{after}}) - \lambda_{\text{n}}$. Similarly, to overestimate the average rate of noise photons corresponding to C_{noise}, I calculate $\lambda_{\text{noise}} = \lambda_{\text{n}} + P_{\text{nonstim}}\lambda_{\text{max}} + P_{\text{after}}\lambda_{\text{max}}$.

- I chose $n_{\text{e}} = 4096$ bits, meaning that in each processed block of the extractor, I process 256 16-bit records at once. Using the average λ_{id} and λ_{noise} input photon rates gives an average of $C_{\text{exp, avg}} = 251$ and $C_{\text{noise, avg}} = 5$. To further account for potential bursty noise, I chose $C_{\text{noise}} = 4 \cdot C_{\text{noise, avg}} = 20$ for the actual upper bound for noise effects, giving $C_{\text{exp}} = 236$. The resulting lower bound for min-entropy is $\underline{H_\infty(\mathbb{D})} = k_{\text{e}} = 2328.327$ bits and the choice of $m_{\text{e}} = 2200$ bits gives $\epsilon \leq 2^{-64}$. To fully parametrize the extractor, a $d_{\text{e}} = n_{\text{e}} + m_{\text{e}} - 1 = 6295$ bit long random seed is needed, which can come from a different trusted source or can even be a "baked in" string due to its reusability. For this purpose, I used the random data collected in Sec. 2.5. This way, 8.594 output bits are created from the available 9.095 bits of min-entropy per measurement record.

While theoretically, the maximum achievable average output bit generation speed is $8.594 \cdot \lambda_{\text{d}} = 11.172$ Mbps, currently, the actual setup is computationally bottlenecked by the limited capabilities of the implementation of the Toeplitz hashing algorithm. As the main goal of the experiment was to demonstrate proof of concept, the software implementation of Toeplitz hashing was not heavily optimized and utilized only the CPU of the computer running it. This practically limited processing speeds to around 10^5 processed measurement samples per second. Fortunately, there are precedents in

the literature for implementations utilizing either an FPGA [121] or GPU [122] that report calculation capabilities of multiple hundreds of Mbps output, presenting a solution for this problem. Nonetheless, the post-processing scheme's main drawback of being computationally expensive is apparent.

4.3.2 Statistical testing results

Similarly to the previous chapter, I used the suites presented in Sec 1.5.2 with default parameters for statistical testing (Same parameter set as in Sec. 2.5.3), presenting here only a summary of the results, while the full outputs of the suites used are available in a github repository at `https://github.com/S0lymi/dissertaion_results`.

From the NIST STS, the output sequence first failed on 1 of the statistical tests from the 188 total. Since even totally random sequences are expected to fail statistical tests sometimes, I rerun the suite on 10 different disjunct parts of output to determine if this failure was due to some consistent error in the output or simply one of the expected random occurrences of failure. I found that 5 of the 10 datasets had 1 failing analysis case, while the remaining passed all the test cases. Importantly, the 5 failing test cases corresponded to differing statistical tests each time, meaning that the tested data exhibited the effect of expected random behavior and not some systematic deviation. In the case of Dieharder, I observed similar results: Initially, 2 of the many test cases reported a "WEAK" assessment (Dieharder's way to signal additional investigation is advised). Due to this, I rerun the battery with the *-m 2* flag to run the individual tests with more input data and found that the assessments changed to "PASSED", signaling that the data can indeed be considered random. This notion is further supported by the fact that the output data successfully passed all the other test cases in TestU01's Alphabit, Rabbit, and SmallCrush batteries,[20] while showing statistical properties when examined with ENT that are consistent with the expected uniform output distribution.

4.3.3 Effect of additive noise on achievable output rates

From Sec. 4.2.4, we can see that the achievable output efficiency and, consequently, the final output rate are heavily influenced by the magnitude of additive noise. Figure 4.2 shows the effect of changing λ_{noise}, while all other physical parameters stay fixed with values calculated in Sec. 4.3.1 and m_{e} is chosen so that $\epsilon < 2^{-64}$ for all data points.

[20]The Crush battery could not complete due to the finite input length.

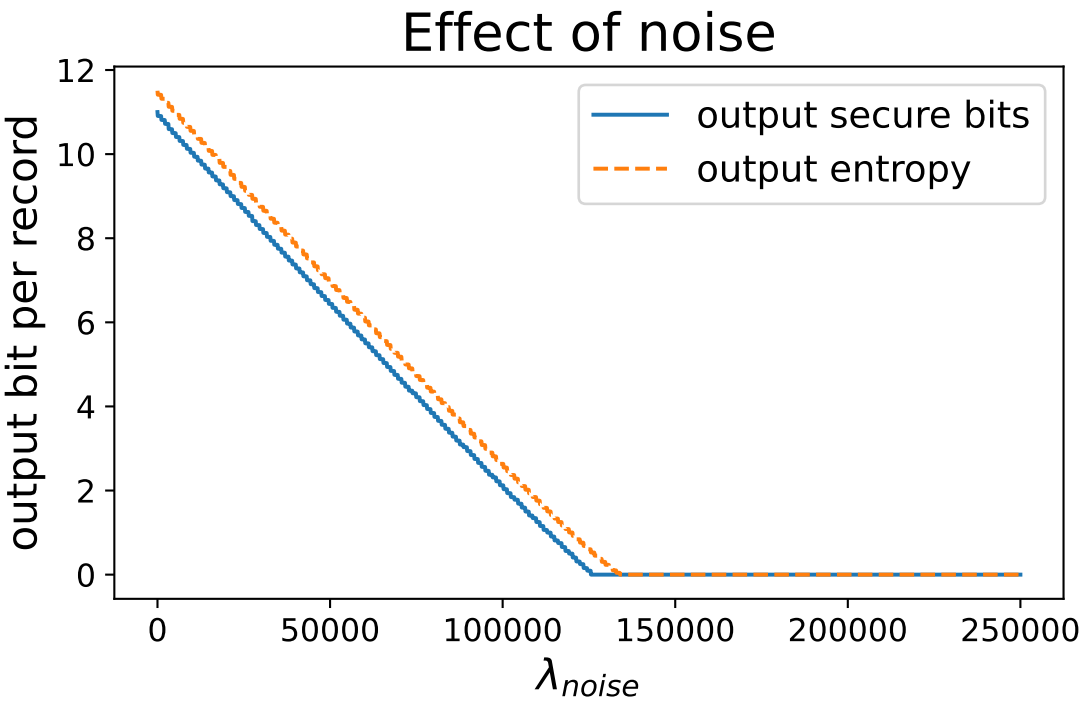

Figure 4.2: Effect of additive noise on achievable output bit and entropy rates

The lines for the m_e output bits and k_e output entropy decrease in a staircase with an almost fixed distance between them. This is the result of the quantified nature of possible m_e and C_{noise} parameters (both must be an integer), while the distance is determined by the actual ϵ value first satisfying the requirement of $\epsilon < 2^{-64}$ according to (4.1). Data shows that without noise, the experimental setup would have an efficiency of 11.09375 output bits per measurement record for a hypothetical bit generation speed of around 14.4 Mbps. On the other hand, for any λ_{noise} values above 125496, the system can no longer guarantee the quality requirement of $\epsilon < 2^{-64}$ for any m_e, while for $\lambda_{noise} \geq 133932$ even the k_e lower bound for secure extractable min-entropy drops to 0. Still, these values show that the system can tolerate relatively high noise levels (reaching around 0.1 in peak $\lambda_{noise}/\lambda_d$ ratio) and still provide secure output, albeit at the cost of heavily reduced output efficiency.

4.4 Summary of the results

Using the mathematical model presented in Sec. 2.2, I proposed a post-processing scheme for QRNGs based on photonic time-of-arrival that utilizes min-entropy estimation for parametrizing universal hash functions. The main idea is that once a lower bound for the min-entropy of a distribution is known, universal hash

functions can be used for randomness extraction due to the leftover hash lemma. I presented a lower bound for min-entropy in the non-ideal case of continuously running measurement clock with dead time. Then, I showed ways to account for the typical distortion effects of fluctuating input rate, unknown dead time, or even additive noise sources with an upper limit on intensity. I also showed, through a practical example, how these results can be used to determine the parameters of a Toeplitz hashing algorithm for generating secure, good-quality output random bitstring with an intentionally non-ideal measurement setup. The quality of the output is also verified by statistical testing tools. Results presented in the chapter form the basis of Thesis III.

While the presented post-processing scheme is capable of handling some errors and non-idealities, this capability comes at the cost of reduced output bitrate. Due to this, if feasible, better characterization of potential error sources is often worthwhile even when using this scheme, as it may be possible to find tighter lower bounds for the min-entropy than what is presented in the chapter, making better and stricter characterization of noise sources a possible direction for future development. This may especially be true for well-characterized setups in a controlled, trusted environment, where tighter assumptions can lead to different, tighter bounds on entropy.

An additional practically important thing to note regarding the presented scheme is that it is computationally expensive to calculate the output bitstrings. This means that either finding more efficient ways of realizing the presented Toeplitz hash based algorithm or utilizing other less computationally demanding universal hash algorithms for randomness extraction is a possible direction for future development that can greatly boost practical applicability.

Chapter 5

Summary of Theses

Thesis I. Correlation avoidance in single photon detecting quantum random number generators by dead time overestimation

I proposed an algorithm to avoid the otherwise present correlations between consecutive measurement samples when using a continuously running clock with a photonic time-of-arrival based QRNG. The algorithm produces an output distribution that is identical to the ideal case of zero starting phase and no dead time, thus preserving the memoryless property of the underlying physical process. I also calculated and then evaluated, and experimentally verified the main performance measures of the algorithm, showing a comprehensive overview of the capabilities, benefits, costs, and limits of using it.

I.1 I developed an algorithm to eliminate the unwanted stochastic effects, like unwanted correlations between consecutive samples, that are present in time-of-arrival QRNGs mainly due to using a non-ideal continuously running measurement clock or a detector with nonzero dead time. The algorithm exploits the observation that these non-idealities affect only the distribution of measurement samples with small values and operates by selective elimination of the affected problematic values while accordingly modifying the rest of the measurement results. This way, the algorithm creates independent and exponentially distributed output values from the originally correlated measurement samples, at the cost of reduced output sample rates.

I.2 To quantify the performance cost of using the algorithm, I deduced a closed-form expression of the reduction in output sample rate in the form of (Expected number of output samples)/(unit time).

I.3 I gave bounds for the algorithm's previously calculated output sample rate

in the more general cases of non-constant dead time. For this, I first showed that the output rate is monotonic in the dead time and used this monotonicity to give bounds on the output rate.

I.4 I also investigated the output rate reduction caused by the algorithm compared to the case of not using it, as well as the achievable maximum output rate and entropy of the output. To validate my theoretical results, I used experimental data from a physical time-of-arrival QRNG setup. My results showed a clear contrast between unprocessed output data and output data processed with the algorithm. I showed this difference in output quality both by utilizing general statistical testing tools and some more specific statistical evaluation methods I developed specially for this measurement scenario.

Related own publications: **J2**, **J4**, **C6**

Thesis II. Post-processing for random number generators inspired by the probability integral transform

I developed a method inspired by the continuous probability integral transform to transform the discretely distributed measurement statistics of a QRNG device to a distribution that is close to uniform with an upper bound on the statistical distance between the transformed and ideal uniform distribution. Since the main goal of QRNGs is to generate uniformly distributed output bitstrings, the developed method can be used for post-processing with QRNG setups. I also showed the practical effects of potential errors and proposed a simple bit generation scheme, which I used to experimentally validate my theoretical results.

II.1 I developed a method inspired by the probability integral transform to create a close-to-uniform distribution from a known RNG output distribution and gave bounds on the bias of the samples and statistical distance of the output distribution from the uniform distribution. Assuming independent measurement samples, I showed that by using the joint distribution of multiple samples, the bias and statistical distance can be made arbitrarily low in the ideal case of perfectly characterized input sample distributions.

II.2 I calculated the bounds for bias and statistical distance for the non-ideal case where the distribution of the used measurement samples is only known with some estimation error, both for the case of using singular input samples and the case of using the joint distribution of multiple input samples. I also gave bounds for bias

and statistical distance for the practically interesting case of erroneously estimated photon input rate for time-of-arrival QRNGS.

II.3 I showed that in the case of estimation errors, the bias and statistical distance cannot be made arbitrarily low and gave a formula for finding the optimal parameters for producing the output distribution with the lowest possible statistical distance from the goal uniform distribution.

II.4 I proposed a practical bit generation scheme, which I used to experimentally investigate the applicability of my previous theoretical results, utilizing experimental data gathered with a real-world time-of-arrival QRNG setup.

Related own publications: **C7**

Thesis III. Efficiency and quality improvement of time-of-arrival quantum random number generators with universal hashing

I gave various lower bounds, assuming various possible ideal and non-ideal scenarios, for the min-entropy of measurement results of non-ideal time-of-arrival QRNGs. Since universal hash functions can be used as randomness extractors if a lower bound for min-entropy is known, these bounds can be used to securely parametrize a post-processing scheme for random bit generation. I utilized this to show the validity of my results experimentally by implementing and parametrizing a Toeplitz hash based bit generation scheme paired with measurement results from a physical time-of-arrival QRNG setup and statistically testing the quality of the output.

III.1 I gave a lower bound for min-entropy in non-ideal time-of-arrival QRNGs using a non-restartable measurement clock with nonzero dead time that can be used to securely parametrize hash-based randomness extractors. This lower bound is $H_\infty(D) = -\log_2\left(1 - \mathrm{e}^{-\lambda\tau}\right)$, where $H_\infty(D)$ is the lower bound for min-entropy, λ is the input photon rate and τ is the resolution of the measurement clock.

III.2 I showed that the previously calculated lower bound is monotonic in both the input rate and dead time. I provided a formula for calculating a corrected lower bound for the min-entropy in practically interesting cases where the exact input photon rate or dead time is unknown, with only some upper bound known for the values.

III.3 I provided a method to account for the effects of possible additive noise sources, allowing safe parameterization of hash-based extractors even in non-ideal cases where environmental noise sources or even a limited capability malicious

attacker may be present.

III.4 I collected measurement samples utilizing a real-world time-of-arrival QRNG setup and implemented and parametrized a Toeplitz hash based extractor for bit generation, both to present an example of using my previous results in practical scenarios and to experimentally validate the correctness of the derived formulae.

Related own publications: **J3**, **C4**

List of Publications

This is a full list of my publications as of writing this manuscript; not all of them are related to my theses. Thesis related publications are directly mentioned in Chapter and referenced when necessary in other chapters.

Journal Papers

J1 Balázs Solymos and László Bacsárdi. "Real-time Processing System for a Quantum Random Number Generator". In: *Infocommunications journal* 12.1 (2020), pp. 53–59. ISSN: 2061-2079. DOI: 10.36244/icj.2020.1.8. URL: http://dx.doi.org/10.36244/icj.2020.1.8

J2 Ágoston Schranz, Balázs Solymos, and Miklós Telek. "Stochastic performance analysis of a time-of-arrival quantum random number generator". In: *IET Quantum Communication* (Dec. 2023). ISSN: 2632-8925. DOI: 10.1049/qtc2.12080. URL: http://dx.doi.org/10.1049/qtc2.12080

J3 Balázs Solymos and László Bacsárdi. "Secure post-processing for non-ideal photon arrival time based quantum random number generator". In: *Infocommunications journal* 16.1 (2024), pp. 12–19. ISSN: 2061-2079. DOI: 10.36244/icj.2024.1.2. URL: http://dx.doi.org/10.36244/icj.2024.1.2

J4 Balázs Solymos, Ágoston Schranz, and Miklós Telek. "Correlation avoidance in single-photon detecting quantum random number generators by dead time overestimation". In: *EPJ Quantum Technology* 11.1 (Sept. 2024). ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-024-00272-8. URL: http://dx.doi.org/10.1140/epjqt/s40507-024-00272-8

Conference Papers

C1 Balázs Solymos and László Bacsárdi. "Affordable statistical testing for quantum random number generators in space applications". In: *Selected papers of the 6th International Conference on Research, Technology and Education of Space (H-SPACE 2020)*. 2020. URL: https://m2.mtmt.hu/api/publication/31833498

C2 Balázs Solymos and László Bacsárdi. "Statisztikai eszközök űrbéli kvantum véletlenszámgenerátorokhoz". In: *Magyar Űrkutatási Fórum 2021 - Az előadások összefoglalói*. 2021, pp. 43–43. URL: https://m2.mtmt.hu/api/publication/32516853

C3 Balázs Solymos and László Bacsárdi. "Statistical testing tools for Quantum Random Number Generators onboard CubeSats". In: *IAF Space Communications and Navigation Symposium 2021 at the 72nd International Astronautical Congress, IAC 2021*. Dubai, UAE, Oct. 2021, pp. 1–6. URL: https://m2.mtmt.hu/api/publication/32519888

C4 Balázs Solymos and László Bacsárdi. "Efficiency improvement of photon arrival time based quantum random number generator with hashing". In: *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. Vol. 78. IEEE, May 2023, pp. 000053–000058. DOI: 10.1109/saci58269.2023.10158613. URL: http://dx.doi.org/10.1109/saci58269.2023.10158613

C5 Balázs Solymos, Tamás Nepusz, and László Bacsárdi. "Routing the quantum internet in large LEO satellite constellations". In: *2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI)*. Vol. 37. IEEE, Nov. 2023, pp. 000397–000402. DOI: 10.1109/cinti59972.2023.10382070. URL: http://dx.doi.org/10.1109/cinti59972.2023.10382070

C6 Ágoston Schranz, Balázs Solymos, and Miklós Telek. "Experimental Time-of-Arrival Quantum Random Number Generation with Dead Time Overestimation". In: *2024 7th International Balkan Conference on Communications and Networking (BalkanCom)*. IEEE, June 2024. DOI: 10.1109/balkancom61808.2024.10557176. URL: http://dx.doi.org/10.1109/balkancom61808.2024.10557176

C7 Balázs Solymos and László Bacsárdi. "Probability integral based post-processing for photonic quantum random number generators". In: *2025 IEEE Cybernetics & Informatics (K&I)*. IEEE, 2025. (submitted)

Appendix A

A.1 Distribution of continuous clock phases

As presented in Ref. [108], we can further investigate the distribution of γ_i. More precisely, we assume $\gamma_0 = x$ and compute the distribution of γ_1. For γ_1 we have

$$\gamma_1 = \tau \left\langle \frac{\gamma_0 + T}{\tau} \right\rangle, \tag{A.1}$$

where $\langle a \rangle = a - \lfloor a \rfloor$ is the fractional part of a, and the subscript of T_1 is suppressed for notational convenience. Based on this relation (for $0 \leq x, y < \tau$), the conditional cumulative distribution function (CDF) of γ_1 is

$$\begin{aligned}
&F_{\gamma_1|\gamma_0=x}(y) = \Pr\big(\gamma_1 < y \mid \gamma_0 = x\big) \\
&= \Pr(x + T < y) + \sum_{i=1}^{\infty} \Pr(i\tau \leq x + T < i\tau + y) \\
&= \Pr\big(T < \max(y - x, 0)\big) + \sum_{i=1}^{\infty} \Pr(i\tau - x \leq T < i\tau + y - x) \\
&= \chi_{\{y>x\}} \left(1 - \mathrm{e}^{-\lambda(y-x)}\right) \\
&\qquad + \sum_{i=1}^{\infty} \left[\left(1 - \mathrm{e}^{-\lambda(i\tau+y-x)}\right) - \left(1 - \mathrm{e}^{-\lambda(i\tau-x)}\right) \right] \\
&= \chi_{\{y>x\}} \left(1 - \mathrm{e}^{-\lambda(y-x)}\right) + \sum_{i=1}^{\infty} \left(\mathrm{e}^{-\lambda(i\tau-x)} - \mathrm{e}^{-\lambda(i\tau+y-x)}\right) \\
&= \chi_{\{y>x\}} \left(1 - \mathrm{e}^{-\lambda(y-x)}\right) + \mathrm{e}^{\lambda x} \left(1 - \mathrm{e}^{-\lambda y}\right) \sum_{i=1}^{\infty} \mathrm{e}^{-\lambda i\tau} \\
&= \chi_{\{y>x\}} \left(1 - \mathrm{e}^{-\lambda(y-x)}\right) + \mathrm{e}^{\lambda x} \left(1 - \mathrm{e}^{-\lambda y}\right) \frac{\mathrm{e}^{-\lambda\tau}}{1 - \mathrm{e}^{-\lambda\tau}},
\end{aligned} \tag{A.2}$$

where χ_A is the indicator of A, and we used the CDF of the exponential distribution with parameter λ, $\Pr(X < x) = 1 - \mathrm{e}^{-\lambda x}$. The conditional probability density function

(PDF) of γ_1 (for $0 \leq x, y < \tau$) is

$$f_{\gamma_1|\gamma_0=x}(y) = \frac{\mathrm{d}}{\mathrm{d}y} F_{\gamma_1|\gamma_0=x}(y) \tag{A.3}$$
$$= \chi_{\{y>x\}} \lambda \mathrm{e}^{-\lambda(y-x)} + \lambda \mathrm{e}^{-\lambda(y-x)} \frac{\mathrm{e}^{-\lambda\tau}}{1-\mathrm{e}^{-\lambda\tau}}.$$

First, note that (A.3) depends on x, which means that the consecutive phase variables are *dependent*. Assuming that the distribution of γ_0 is known, the evolution of the random process $\gamma_0, \gamma_1, \ldots$ can be computed based on (A.3). The CDF and the PDF of the stationary phase at a photon arrival are defined as $F_\gamma(x) = \lim_{n\to\infty} \Pr(\gamma_n < x)$ and $f_\gamma(x) = \frac{\mathrm{d}}{\mathrm{d}x} F_\gamma(x)$.

Theorem 4. *The stationary phase at a photon arrival is uniformly distributed in* $[0, \tau)$.

Proof. The distribution of the stationary phase is associated with the eigenfunction of an operator composed from the characteristic function. Namely, $f_\gamma(\cdot)$ is the solution of

$$f_\gamma(y) = \int_{x=0}^{\tau} f_\gamma(x) f_{\gamma_1|\gamma_0=x}(y)\, \mathrm{d}x, \tag{A.4}$$

with normalization condition $\int_{y=0}^{\tau} f_\gamma(y)\mathrm{d}y = 1$. The solution of this integral equation is $f_\gamma(y) = \frac{1}{\tau}$ for $y \in [0, \tau)$, since

$$f_{\gamma_1|\gamma_0\equiv\text{uniform}}(y) = \int_{x=0}^{\tau} \frac{1}{\tau} f_{\gamma_1|\gamma_0=x}(y)\, \mathrm{d}x$$
$$= \int_{x=0}^{y} \frac{\lambda}{\tau} \mathrm{e}^{-\lambda(y-x)}\, \mathrm{d}x + \int_{x=0}^{\tau} \frac{\lambda}{\tau} \mathrm{e}^{-\lambda(y-x)} \frac{\mathrm{e}^{-\lambda\tau}}{1-\mathrm{e}^{-\lambda\tau}} \mathrm{d}x$$
$$= \frac{\lambda}{\tau} \frac{1-\mathrm{e}^{-\lambda y}}{\lambda} + \frac{\lambda}{\tau} \frac{\mathrm{e}^{-\lambda y}(\mathrm{e}^{\lambda\tau}-1)}{\lambda} \frac{\mathrm{e}^{-\lambda\tau}}{1-\mathrm{e}^{-\lambda\tau}} = \frac{1}{\tau}. \tag{A.5}$$

□

That is, if γ_0 is uniformly distributed on the τ grid, then every consecutive γ_i has a uniform marginal distribution.

A.2 Calculation of correlation coefficients of DTDs

Without loss of generality, set $i = 1$ and $i + 1 = 2$ and compute the correlation ρ_{D_1,D_2} based on

$$\rho_{D_1,D_2} = \frac{\mathbb{E}(D_1 D_2) - \mathbb{E}(D_1)\,\mathbb{E}(D_2)}{\sqrt{\left(\mathbb{E}\left(D_1^2\right) - \mathbb{E}(D_1)^2\right)\left(\mathbb{E}\left(D_2^2\right) - \mathbb{E}(D_2)^2\right)}}. \tag{A.6}$$

A.2.1 Case of no dead time

According to (2.5), for $n_1 > 0$ and $n_2 > 0$, we have

$$\begin{aligned}\Pr\left(D_2 = n_2, D_1 = n_1 \mid \gamma_0 = x_0\right) &= \int_{x_2=0}^{\tau}\int_{x_1=0}^{\tau} f_{n_2}\left(x_2, x_1\right) \cdot f_{n_1}\left(x_1, x_0\right) \mathrm{d}x_1 \mathrm{d}x_2 \\ &= \int_{x_2=0}^{\tau}\int_{x_1=0}^{\tau} \lambda \mathrm{e}^{-\lambda(x_2+n_2\tau-x_1)} \lambda \mathrm{e}^{-\lambda(x_1+n_1\tau-x_0)} \mathrm{d}x_1 \mathrm{d}x_2 \\ &= \lambda\tau\left(1-\mathrm{e}^{-\lambda\tau}\right)\mathrm{e}^{-\lambda(n_1\tau+n_2\tau-x_0)}.\end{aligned} \tag{A.7}$$

Furthermore, using the uniform distribution of γ_0, the expectation of the product $D_1 D_2$ becomes

$$\begin{aligned}\mathbb{E}(D_1 D_2) &= \int_0^{\tau} \frac{1}{\tau}\,\mathbb{E}(D_1 D_2 \mid \gamma_0 = x)\mathrm{d}x \\ &= \int_0^{\tau} \frac{1}{\tau} \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} ij \Pr\big(D_2 = i, D_1 = j \mid \gamma_0 = x\big)\mathrm{d}x = \frac{\mathrm{e}^{-\lambda\tau}}{\left(1-\mathrm{e}^{-\lambda\tau}\right)^2}.\end{aligned} \tag{A.8}$$

The DTDs' expected values $\mathbb{E}(D_1) = \mathbb{E}(D_2)$ and second moments $\mathbb{E}\left(D_1^2\right) = \mathbb{E}\left(D_2^2\right)$ can be calculated using Ref. [108, Eq. (12)], yielding

$$\mathbb{E}(D_1) = \mathbb{E}(D_2) = \sum_{n=1}^{\infty} n \cdot \Pr(D_1 = n) = \frac{\left(1-\mathrm{e}^{-\lambda\tau}\right)^2}{\lambda\tau \mathrm{e}^{-\lambda\tau}} \sum_{n=1}^{\infty} n \cdot \mathrm{e}^{-\lambda\tau n} = \frac{1}{\lambda\tau} \tag{A.9}$$

and

$$\mathbb{E}\left(D_1^2\right) = \mathbb{E}\left(D_2^2\right) = \sum_{n=1}^{\infty} n^2 \cdot \Pr(D_1 = n) = \frac{\left(1-\mathrm{e}^{-\lambda\tau}\right)^2}{\lambda\tau \mathrm{e}^{-\lambda\tau}} \sum_{n=1}^{\infty} n^2 \cdot \mathrm{e}^{-\lambda\tau n} = \frac{\left(1+\mathrm{e}^{-\lambda\tau}\right)}{\lambda\tau\left(1-\mathrm{e}^{-\lambda\tau}\right)}. \tag{A.10}$$

Finally, the correlation between D_1 and D_2, purely a function of the product $\lambda\tau$, is

$$\rho_{D_1,D_2} = \frac{\frac{\mathrm{e}^{-\lambda\tau}}{\left(1-\mathrm{e}^{-\lambda\tau}\right)^2} - \frac{1}{(\lambda\tau)^2}}{\frac{\left(1+\mathrm{e}^{-\lambda\tau}\right)}{\lambda\tau\left(1-\mathrm{e}^{-\lambda\tau}\right)} - \frac{1}{(\lambda\tau)^2}} = \frac{(\lambda\tau)^2\mathrm{e}^{-\lambda\tau} - \left(1-\mathrm{e}^{-\lambda\tau}\right)^2}{\lambda\tau\left(1-\mathrm{e}^{-2\lambda\tau}\right) - \left(1-\mathrm{e}^{-\lambda\tau}\right)^2}. \tag{A.11}$$

The correlation tends to zero as $(\lambda\tau) \to 0$ or $(\lambda\tau) \to \infty$, its value is negative in between (see Fig. 2.2). It is monotonically decreasing until obtaining its minimum of -0.2233 around $\lambda\tau = 3.5749$. Thus, increasing $\lambda\tau$ from zero increases the magnitude of correlations between successive DTDs,[21] and the resulting sequence of random variables will always contain systematic correlations.

[21] This statement is valid until the global minimum is reached at $\lambda\tau = 3.5749$; however, values of $\lambda\tau > 1$ are impractical. They represent a domain in which, on average, more than one photon arrives within a clock period. This practically means a good-quality SPD with high photon rate tolerance connected to a low-resolution TDC, which is outside the domain relevant for most QRNG setups.

A.2.2 Nonzero dead time case

Along the lines of the dead time free case, the distribution of D_1 and the joint distribution of D_1 and D_2 can be calculated from the conditional density function, that is

$$\begin{aligned} f_n(x,y) &= \frac{\mathrm{d}}{\mathrm{d}y}F_n(x,y) \\ &= \begin{cases} \chi_{\{x+\delta<y\}}\lambda \mathrm{e}^{-\lambda(y-x-\delta)} & \text{if } n=k, \\ \chi_{\{x+\delta<\tau\}}\lambda \mathrm{e}^{-\lambda(y-x-\delta+\tau)} + \chi_{\{\tau<x+\delta<\tau+y\}}\lambda \mathrm{e}^{-\lambda(y-x-\delta+\tau)} & \text{if } n=k+1, \\ \lambda \mathrm{e}^{-\lambda(y+(n-k)\tau-\delta-x)} & \text{if } n>k+1, \end{cases} \end{aligned} \tag{A.12}$$

for $n \geq k$.

Utilizing the uniform distribution of γ_0, as

$$p_{n_1} \triangleq \Pr\left(D_1 = n_1\right) = \frac{1}{\tau}\int_{x_0=0}^{\tau}\int_{x_1=0}^{\tau} f_{n_1}\left(x_1, x_0\right)\mathrm{d}x_1\mathrm{d}x_0, \tag{A.13}$$

$$\begin{aligned} p_{n_1,n_2} &\triangleq \Pr\left(D_2 = n_2, D_1 = n_1\right) \\ &= \frac{1}{\tau}\int_{x_0=0}^{\tau}\int_{x_1=0}^{\tau}\int_{x_2=0}^{\tau} f_{n_2}\left(x_2, x_1\right)\cdot f_{n_1}\left(x_1, x_0\right)\mathrm{d}x_2\mathrm{d}x_1\mathrm{d}x_0. \end{aligned} \tag{A.14}$$

The distributions allow us to calculate the expected values $\mathbb{E}\left(D_1-k\right)$, $\mathbb{E}\left(\left(D_1-k\right)^2\right)$ and $\mathbb{E}\left(\left(D_1-k\right)\left(D_2-k\right)\right)$, along with the correlation $\rho_{D_1,D_2} = \rho_{D_1-k,D_2-k}$:

$$\mathbb{E}\left(D_1-k\right) = \sum_{n_1=1}^{\infty} n_1 p_{n_1} = \frac{1+\lambda\delta}{\lambda\tau}, \tag{A.15}$$

$$\mathbb{E}\left(\left(D_1-k\right)^2\right) = \sum_{n_1=1}^{\infty} n_1^2 p_{n_1} = \frac{1+\lambda\delta+\mathrm{e}^{-\lambda\tau}(2\mathrm{e}^{\lambda\delta}-1-\lambda\delta)}{\lambda\tau\left(1-\mathrm{e}^{-\lambda\tau}\right)}, \tag{A.16}$$

$$\mathbb{E}\left(\left(D_1-k\right)\left(D_2-k\right)\right) = \sum_{n_1=1}^{\infty} n_1 \sum_{n_2=1}^{\infty} n_2 p_{n_1,n_2}, \tag{A.17}$$

$$\rho_{D_1,D_2} = \mathrm{corr}(D_1,D_2) = \frac{\mathbb{E}\left(\left(D_1-k\right)\left(D_2-k\right)\right) - \mathbb{E}^2\left(D_1-k\right)}{\mathbb{E}\left(\left(D_1-k\right)^2\right) - \mathbb{E}^2\left(D_1-k\right)}, \tag{A.18}$$

where closed-form expressions are provided for the former two and computed the latter two numerically.

A.3 Computation and estimation of further performance indices of the presented overestimation method

A.3.1 Computation of general performance indices

The analysis approach of this section allows the computation of more detailed performance indices of Algorithm 1.

To compute the distribution of Θ_1 based on 2.14, introduce $\hat{\Theta}(z, x_0) = \mathbb{E}\left(z^{\Theta_1} \mid \gamma_0 = x_0\right)$ (the z-transform of Θ_1), $F_{\mathrm{d}}(z, x_0, x_1) = \sum_{i=0}^{m} z^i f_i(x_0, x_1)$ (describing the dropped arrivals), and $F_{\mathrm{a}}(z, x_0, x_1) = \sum_{i=m+1}^{\infty} z^i f_i(x_0, x_1)$ (describing the accepted arrivals). Based on these functions, $\hat{\Theta}(z, x_0)$ can be obtained as

$$\begin{aligned}\hat{\Theta}(z, x_0) = &\\ &\int_{x_1} F_{\mathrm{a}}(z, x_0, x_1)\mathrm{d}x_1 + \int_{x_1}\int_{x_2} F_{\mathrm{d}}(z, x_0, x_1)F_{\mathrm{a}}(z, x_1, x_2)\mathrm{d}x_2\mathrm{d}x_1 + \dots \\ &= \sum_{i=1}^{\infty}\int_{x_1}\dots\int_{x_i} F_{\mathrm{d}}(z, x_0, x_1)\dots F_{\mathrm{d}}(z, x_{i-2}, x_{i-1}) \\ &\qquad F_{\mathrm{a}}(z, x_{i-1}, x_i)\mathrm{d}x_i\dots\mathrm{d}x_1 \end{aligned} \quad \text{(A.19)}$$

The CDF of initial phase distribution after an accepted photon arrival is provided in the second term of (2.20). Its density function (obtained by a derivation according to the function parameter) is

$$f_{\mathrm{init}}(x) = \frac{\lambda \mathrm{e}^{-\lambda x}}{1 - \mathrm{e}^{-\lambda\tau}}, \quad \text{(A.20)}$$

for $0 \le x \le \tau$. The distribution of Θ_1 is obtained in z-transform domain as

$$\hat{\Theta}(z) = \mathbb{E}\left(z^{\Theta_1}\right) = \int_x f_{\mathrm{init}}(x)\hat{\Theta}(z, x)\mathrm{d}x. \quad \text{(A.21)}$$

Note that the mean "time" between observations, which was computed directly in Sec. 2.3.1, is

$$\mathbb{E}(\Theta) = \frac{\mathrm{d}}{\mathrm{d}z}\hat{\Theta}(z)\bigg|_{z=1}.$$

Unfortunately, the infinite number of integrals in (A.19) makes the numerical analysis of $\hat{\Theta}(z)$ computationally challenging but can be efficiently approximated using the following Erlangization approach.

A.3.2 Approximation based on an Erlang clock

Following the pattern of [108, eq. (50)], map $f_n(x_0, x_1)$ into matrices of size $\hat{N} \times \hat{N}$

$$\begin{aligned}\{\mathbf{A}_n\}_{ij} &= \Pr\big(J_1 = j, D_1 = n \mid J_0 = i\big) \\ &= \begin{cases} \Pr\Big(\Omega = n\hat{N} + j - i - L\Big) & \text{if } n\hat{N} + j \geq i + L, \\ 0 & \text{otherwise,} \end{cases} \\ &= \begin{cases} q(1-q)^{n\hat{N}+j-i-L} & \text{if } n\hat{N} + j \geq i + L, \\ 0 & \text{otherwise,} \end{cases}\end{aligned}$$

where $\hat{N}$ is the order of the Erlang clock, $q = \frac{\lambda\tau}{\lambda\tau+\hat{N}}$ and the discretized version of dead time is $L = \lfloor \hat{N}\zeta/\tau \rfloor$ (integer). Furthermore, $J_i \in \{1, \ldots, \hat{N}\}$ denotes the phase of the grid process at S_i, while Ω denotes the number of phase changes.

To compute the number of intervals associated with a dropped and an accepted photon arrival, introduce $\mathbf{A}_{\mathrm{d}}(z) = \sum_{n=0}^{m} \mathbf{A}_n z^n$ and $\mathbf{A}_{\mathrm{a}}(z) = \sum_{n=m+1}^{\infty} \mathbf{A}_n z^n$.

The Erlang clock based approximate of $\hat{\Theta}(z, x_0)$ is obtained by considering that an accepted photon arrival is preceded by an arbitrary number of dropped photon arrivals, thus

$$\mathbf{\Theta}(z) = \sum_{i=0}^{\infty} \mathbf{A}_{\mathrm{d}}^i(z)\mathbf{A}_{\mathrm{a}}(z) = \big(\mathbf{I} - \mathbf{A}_{\mathrm{d}}(z)\big)^{-1}\mathbf{A}_{\mathrm{a}}(z), \tag{A.22}$$

from which the distribution of Θ can be obtained by inverse z-transform and its kth factorial moment as

$$f_k = \mathbb{E}\left(\Theta(\Theta-1)\ldots(\Theta-k+1)\right) = \frac{\mathrm{d}^k}{\mathrm{d}z^k}\, v_{\mathrm{init}}\mathbf{\Theta}(z)\mathbb{1}\bigg|_{z=1}, \tag{A.23}$$

where $v_{\mathrm{init}} = \frac{\hat{v}}{\hat{v}\mathbb{1}}$ and $\{\hat{v}\}_i = q(1-q)^{i-1}$, is the discretized version of (A.20). E.g., the squared coefficient of variation (SCV) of Θ can be obtained from the factorial moments as

$$C_\Theta^2 = \frac{E(\Theta^2) - E(\Theta)^2}{E(\Theta)^2} = \frac{f_2 + f_1 - f_1^2}{f_1^2}. \tag{A.24}$$

Fig. A.1 shows example estimation and simulation results (same simulation parameters as in 2.4) for this example case. The approximation already has a decent accuracy with relative errors on the order of 10^{-2} for $\hat{N} = 100$ and 10^{-3} for $\hat{N} = 1000$ (depicted on Fig. A.1).

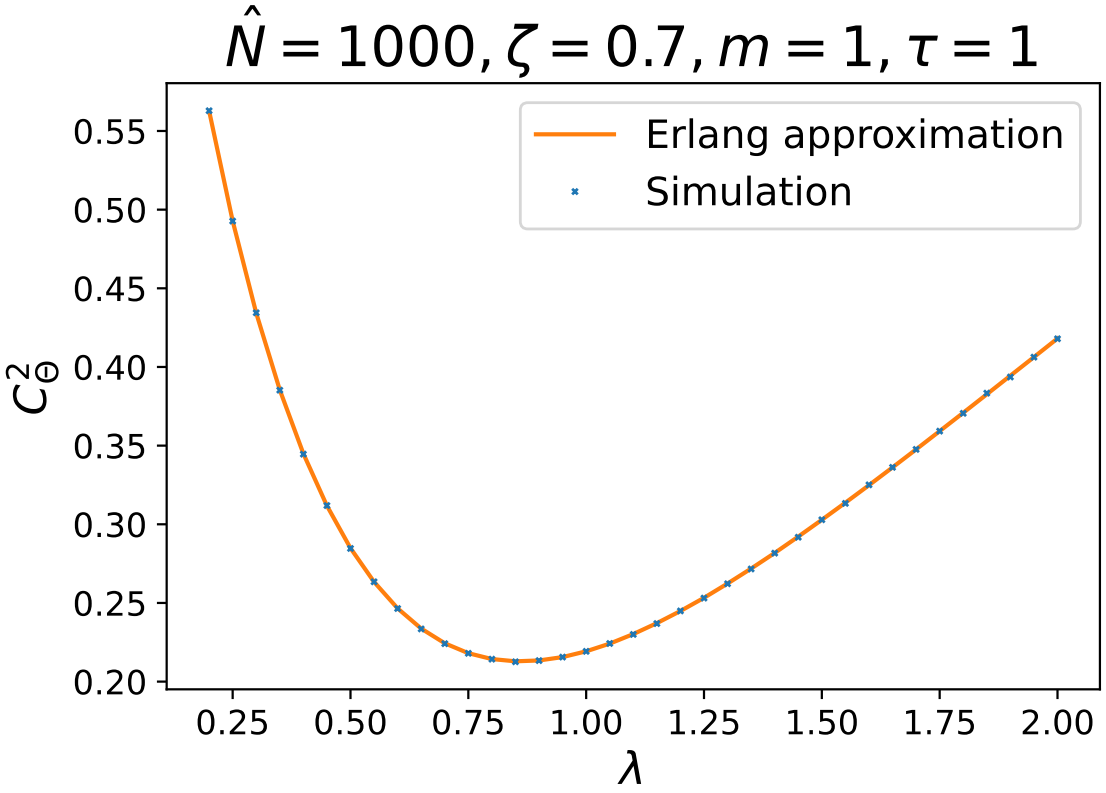

Figure A.1: Simulated and approximated (with $\hat{N} = 1000$ Erlang phases) results for C_Θ^2 for different λ input rates with fixed $\zeta = 0.7$ dead time.

A.4 Error terms of the joint distribution of Q i.i.d. samples

Theorem 5. *The $\sum_i \underline{e}_i$ error term of the joint distribution of Q i.i.d. samples can be calculated from the $\sum_{q_\vartheta} e_{q_\vartheta}$ error term of a single sample as:*

$$\sum_i \underline{e}_i = \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1 \right)^Q - 1. \tag{A.25}$$

Proof. For the case of $Q = 1$,

$$\sum_i \underline{e}_i = \sum_{q_\vartheta} e_{q_\vartheta} = \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1 \right)^Q - 1 = \sum_{q_\vartheta} e_{q_\vartheta} + 1 - 1 = \sum_{q_\vartheta} e_{q_\vartheta}. \tag{A.26}$$

Using induction and (3.52) to calculate the case of $Q+1$ samples from the case of Q

samples:

$$\sum_i \underline{e}_i = \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1\right)^Q - 1 + \sum_{q_\vartheta} e_{q_\vartheta} + \sum_{q_\vartheta} e_{q_\vartheta} \left(\left(\sum_{q_\vartheta} e_{q_\vartheta} + 1\right)^Q - 1\right)$$
$$= \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1\right)^Q + \sum_{q_\vartheta} e_{q_\vartheta} \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1\right)^Q - 1 = \left(\sum_{q_\vartheta} e_{q_\vartheta} + 1\right)^{Q+1} - 1 \tag{A.27}$$

□

A.5 Calculation of c_i with the presented practical bit generation scheme

The

$$c_i = \sum_{j=0}^{i} \underline{p}_j = \sum_{j=0}^{i} \prod_{\vartheta=0}^{Q-1} p_{q_\vartheta,j} \tag{A.28}$$

can be separated into smaller sums using the index assignment rules presented in (3.82) and (3.83):

$$c_i = \sum_{j=0}^{i} \underline{p}_j = \sum_{j=0}^{t_Q((N_O-1,N_O-1,\ldots,N_O-1,q_{Q-1}-1))} + \sum_{j=t_Q((0,0,\ldots,0,q_{Q-1}))}^{t_Q((N_O-1,N_O-1,\ldots,q_{Q-2}-1,q_{Q-1}))}$$
$$+ \ldots + \sum_{j=t_Q((0,0,\ldots,q_{Q-2},q_{Q-1}))}^{t_Q((N_O-1,q_1-1,\ldots,q_{Q-2}-1,q_{Q-1}))} + \sum_{j=t_Q((0,q_1,\ldots,q_{Q-2},q_{Q-1}))}^{t_Q((q_0-1,q_1,\ldots,q_{Q-2}-1,q_{Q-1}))} + \underline{p}_{t_Q((q_0,q_1,\ldots,q_{Q-2},q_{Q-1}))}. \tag{A.29}$$

Using $\underline{p}_j = \prod_{\vartheta=0}^{Q-1} p_{q_\vartheta,j}$ and the fact that $\sum_{j=0}^{N_O-1} p_j = 1$, we can rewrite (A.29) as:

$$c_i = \sum_{j=0}^{i} \underline{p}_j = \sum_{j=0}^{q_{Q-1}-1} p_j + p_{q_{Q-1}} \sum_{j=0}^{q_{Q-2}-1} p_j + \ldots + \prod_{k=2}^{Q-1} p_{q_k} \sum_{j=0}^{q_1-1} p_j + \prod_{k=1}^{Q-1} p_{q_k} \sum_{j=0}^{q_0-1} p_j + \prod_{k=0}^{Q-1} p_{q_k}$$
$$= \sum_{\vartheta=0}^{Q-1} \left(\prod_{k=\vartheta+1}^{Q-1} p_{q_k} \sum_{j=0}^{q_\vartheta-1} p_j \right) + \prod_{k=0}^{Q-1} p_{q_k}. \tag{A.30}$$

Bibliography

[1] Y. Dodis et al. "On the (Im)possibility of Cryptography with Imperfect Randomness". In: *45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE. DOI: 10.1109/focs.2004.44. URL: http://dx.doi.org/10.1109/focs.2004.44.

[2] Ian Goldberg and David Wagner. "Randomness and the Netscape browser". In: *Dr Dobb's Journal-Software Tools for the Professional Programmer* 21.1 (1996), pp. 66–71.

[3] Nadia Heninger et al. "Mining your Ps and Qs: Detection of widespread weak keys in network devices". In: *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*. 2012, pp. 205–220. DOI: 10.5555/2362793.2362828.

[4] Frank Miller. *Telegraphic code to insure privacy and secrecy in the transmission of telegrams*. CM Cornwell, 1882.

[5] GS Vernam. "Secret signaling system". 1310719. July 1919. URL: https://patents.google.com/patent/US1310719A/en.

[6] Steven M. Bellovin. "Frank Miller: Inventor of the One-Time Pad". In: *Cryptologia* 35.3 (July 2011), pp. 203–222. ISSN: 1558-1586. DOI: 10.1080/01611194.2011.583711. URL: http://dx.doi.org/10.1080/01611194.2011.583711.

[7] Claude E Shannon. "Communication theory of secrecy systems". In: *The Bell System Technical Journal* 28.4 (1949), pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[8] Whitfield Diffie and Martin Hellman. "New Directions in Cryptography (1976)". In: *Ideas That Created the Future*. The MIT Press, Feb. 2021, pp. 421–440. ISBN: 9780262363174. DOI: 10.7551/mitpress/12274.003.0044. URL: http://dx.doi.org/10.7551/mitpress/12274.003.0044.

[9] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126. DOI: 10.1145/359340.359342.

[10] Laszlo Gyongyosi and Sandor Imre. "A Survey on quantum computing technology". In: *Computer Science Review* 31 (Feb. 2019), pp. 51–71. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2018.11.002. URL: http://dx.doi.org/10.1016/j.cosrev.2018.11.002.

[11] Prateek Singh et al. "A Survey on Available Tools and Technologies Enabling Quantum Computing". In: *IEEE Access* 12 (2024), pp. 57974–57991. ISSN: 2169-3536. DOI: 10.1109/access.2024.3388005. URL: http://dx.doi.org/10.1109/access.2024.3388005.

[12] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. IEEE. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[13] Duc-Thuan Dam et al. "A Survey of Post-Quantum Cryptography: Start of a New Race". In: *Cryptography* 7.3 (Aug. 2023), p. 40. ISSN: 2410-387X. DOI: 10.3390/cryptography7030040. URL: http://dx.doi.org/10.3390/cryptography7030040.

[14] László Gyöngyösi, Laszlo Bacsardi, and Sandor Imre. "A survey on quantum key distribution". In: *Infocommunications Journal* 11.2 (2019), pp. 14–21. DOI: 10.36244/ICJ.2019.2.2. URL: https://doi.org/10.36244/ICJ.2019.2.2.

[15] Víctor Zapatero, Álvaro Navarrete, and Marcos Curty. "Implementation Security in Quantum Key Distribution". In: *Advanced Quantum Technologies* (Jan. 2024). ISSN: 2511-9044. DOI: 10.1002/qute.202300380. URL: http://dx.doi.org/10.1002/qute.202300380.

[16] W. K. Wootters and W. H. Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (Oct. 1982), pp. 802–803. ISSN: 1476-4687. DOI: 10.1038/299802a0. URL: http://dx.doi.org/10.1038/299802a0.

[17] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (Dec. 2014), pp. 7–11. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.05.025. URL: http://dx.doi.org/10.1016/j.tcs.2014.05.025.

[18] Artur K. Ekert. "Quantum cryptography based on Bell's theorem". In: *Physical Review Letters* 67.6 (Aug. 1991), pp. 661–663. ISSN: 0031-9007. DOI: 10.1103/physrevlett.67.661. URL: http://dx.doi.org/10.1103/physrevlett.67.661.

[19] Víctor Zapatero et al. "Advances in device-independent quantum key distribution". In: *npj Quantum Information* 9.1 (Feb. 2023). ISSN: 2056-6387. DOI: 10.1038/s41534-023-00684-x. URL: http://dx.doi.org/10.1038/s41534-023-00684-x.

[20] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods*. Wiley, Sept. 2008. ISBN: 9783527626212. DOI: 10.1002/9783527626212. URL: http://dx.doi.org/10.1002/9783527626212.

[21] Joseph F. Fitzsimons. "Private quantum computation: an introduction to blind quantum computing and related protocols". In: *npj Quantum Information* 3.1 (June 2017). ISSN: 2056-6387. DOI: 10.1038/s41534-017-0025-3. URL: http://dx.doi.org/10.1038/s41534-017-0025-3.

[22] Laszlo Gyongyosi and Sandor Imre. "Advances in the quantum internet". In: *Communications of the ACM* 65.8 (July 2022), pp. 52–63. ISSN: 1557-7317. DOI: 10.1145/3524455. URL: http://dx.doi.org/10.1145/3524455.

[23] Jack Moshman. "The Generation of Pseudo-Random Numbers on a Decimal Calculator". In: *Journal of the ACM* 1.2 (Apr. 1954), pp. 88–91. ISSN: 1557-735X. DOI: 10.1145/320772.320775. URL: http://dx.doi.org/10.1145/320772.320775.

[24] U.V. Vazirani and V.V. Vazirani. "Efficient And Secure Pseudo-Random Number Generation". In: *25th Annual Symposium on Foundations of Computer Science, 1984.* IEEE. DOI: 10.1109/sfcs.1984.715948. URL: http://dx.doi.org/10.1109/sfcs.1984.715948.

[25] L. Blum, M. Blum, and M. Shub. "A Simple Unpredictable Pseudo-Random Number Generator". In: *SIAM Journal on Computing* 15.2 (May 1986), pp. 364–383. ISSN: 1095-7111. DOI: 10.1137/0215025. URL: http://dx.doi.org/10.1137/0215025.

[26] Makoto Matsumoto and Takuji Nishimura. "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator". In: *ACM Transactions on Modeling and Computer Simulation* 8.1 (Jan. 1998), pp. 3–30. ISSN: 1558-1195. DOI: 10.1145/272991.272995. URL: http://dx.doi.org/10.1145/272991.272995.

[27] Stefan Kutschera et al. "MRNG: Accessing Cosmic Radiation as an Entropy Source for a Non-Deterministic Random Number Generator". In: *Entropy* 25.6 (May 2023), p. 854. ISSN: 1099-4300. DOI: 10.3390/e25060854. URL: http://dx.doi.org/10.3390/e25060854.

[28] Davide G. Marangon, Giuseppe Vallone, and Paolo Villoresi. "Random bits, true and unbiased, from atmospheric turbulence". In: *Scientific Reports* 4.1 (June 2014). ISSN: 2045-2322. DOI: 10.1038/srep05490. URL: http://dx.doi.org/10.1038/srep05490.

[29] Mike Hamburg, Paul Kocher, and Mark E Marson. "Analysis of Intel's Ivy Bridge digital random number generator". In: *Online: http://www. cryptography. com/public/pdf/Intel_ TRN G_ Report_ 20120312. pdf* (2012).

[30] T. Stojanovski, J. Pihl, and L. Kocarev. "Chaos-based random number generators. Part II: practical realization". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 48.3 (Mar. 2001), pp. 382–385. ISSN: 1057-7122. DOI: 10.1109/81.915396. URL: http://dx.doi.org/10.1109/81.915396.

[31] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin. "Quantum random number generators". In: *Reviews of Modern Physics* 89.1 (Feb. 2017). ISSN: 1539-0756. DOI: 10.1103/revmodphys.89.015004. URL: http://dx.doi.org/10.1103/revmodphys.89.015004.

[32] Vaisakh Mannalatha, Sandeep Mishra, and Anirban Pathak. "A comprehensive review of quantum random number generators: concepts, classification and the origin of randomness". In: *Quantum Information Processing* 22.12 (Dec. 2023). ISSN: 1573-1332. DOI: 10.1007/s11128-023-04175-y. URL: http://dx.doi.org/10.1007/s11128-023-04175-y.

[33] Masatugu Isida and Hiroji Ikeda. "Random number generator". In: *Annals of the Institute of Statistical Mathematics* 8.2 (Dec. 1956), pp. 119–126. ISSN: 1572-9052. DOI: 10.1007/bf02863577. URL: http://dx.doi.org/10.1007/bf02863577.

[34] JB Manelis. "Generating random noise with radioactive sources". In: *Electronics (US)* 34.36 (1961).

[35] Francisco Orts et al. "A quantum circuit to generate random numbers within a specific interval". In: *EPJ Quantum Technology* 10.1 (May 2023). ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-023-00174-1. URL: http://dx.doi.org/10.1140/epjqt/s40507-023-00174-1.

[36] Yuanhao Li et al. "Quantum random number generator using a cloud superconducting quantum computer based on source-independent protocol". In: *Scientific Reports* 11.1 (Dec. 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-021-03286-9. URL: http://dx.doi.org/10.1038/s41598-021-03286-9.

[37] Vaishnavi kumar and Padmapriya Pravinkumar. "Quantum random number generator on IBM QX". In: *Journal of Cryptographic Engineering* (Dec. 2023). ISSN: 2190-8516. DOI: 10.1007/s13389-023-00341-1. URL: http://dx.doi.org/10.1007/s13389-023-00341-1.

[38] Ramin Salehi, Mohammad Razaghi, and Bashir Fotouhi. "Hybrid Hadamard and controlled-Hadamard based quantum random number generators in IBM QX". In: *Physica Scripta* 97.6 (May 2022), p. 065101. ISSN: 1402-4896. DOI: 10.1088/1402-4896/ac698b. URL: http://dx.doi.org/10.1088/1402-4896/ac698b.

[39] Si Qi Ng et al. "240 Gbps Quantum Random Number Generator with Photonic Integrated Chip". In: *CLEO 2023*. CLEO_AT. Optica Publishing Group, 2023. DOI: 10.1364/cleo_at.2023.am4n.6. URL: http://dx.doi.org/10.1364/cleo%5C_at.2023.am4n.6.

[40] Ramón Bernardo-Gavito et al. "Extracting random numbers from quantum tunnelling through a single diode". In: *Scientific Reports* 7.1 (Dec. 2017). ISSN: 2045-2322. DOI: 10.1038/s41598-017-18161-9. URL: http://dx.doi.org/10.1038/s41598-017-18161-9.

[41] Thomas Roger et al. "Real-time interferometric quantum random number generation on chip". In: *Journal of the Optical Society of America B* 36.3 (Mar. 2019), B137. ISSN: 1520-8540. DOI: 10.1364/josab.36.00b137. URL: http://dx.doi.org/10.1364/josab.36.00b137.

[42] Bing Bai et al. "18.8 Gbps real-time quantum random number generator with a photonic integrated chip". In: *Applied Physics Letters* 118.26 (June 2021). ISSN: 1077-3118. DOI: 10.1063/5.0056027. URL: http://dx.doi.org/10.1063/5.0056027.

[43] Davide G. Marangon et al. "A fast and robust quantum random number generator with a self-contained integrated photonic randomness core". In: *Nature Electronics* 7.5 (May 2024), pp. 396–404. ISSN: 2520-1131. DOI: 10.1038/s41928-024-01140-0. URL: http://dx.doi.org/10.1038/s41928-024-01140-0.

[44] *ID Quantique Quantis QRNG chip*. https://www.idquantique.com/random-number-generation/products/quantis-qrng-chip/. 2024. (Last accessed 2024/11/14).

[45] *Cryptalabs QRNG hardware*. https://cryptalabs.com/quantum-random-number-generator/. 2024. (Last accessed 2024/11/14).

[46] *QuantumCtek QRNG*. http://www.quantum-info.com/English/product/pfour/liangzisuijishuyuan/2019/0731/579.html. 2024. (Last accessed 2024/03/05).

[47] *Quside QRNGs*. https://quside.com/. 2024. (Last accessed 2024/11/14).

[48] *Toshiba creates Quantum Random Number Generation ready for mass manufacture*. https://www.toshiba.eu/quantum/news/toshiba-creates-quantum-random-number-generation-ready-for-mass-manufacture/. 2024. (Last accessed 2024/11/14).

[49] *Comscire QRNG*. https://comscire.com/random-number-generator-selection-guide/. 2024. (Last accessed 2024/11/14).

[50] S. Pironio et al. "Random numbers certified by Bell's theorem". In: *Nature* 464.7291 (Apr. 2010), pp. 1021–1024. ISSN: 1476-4687. DOI: 10.1038/nature09008. URL: http://dx.doi.org/10.1038/nature09008.

[51] Yang Liu et al. "Device-independent quantum random-number generation". In: *Nature* 562.7728 (Sept. 2018), pp. 548–551. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0559-3. URL: http://dx.doi.org/10.1038/s41586-018-0559-3.

[52] Lynden K. Shalm et al. "Device-independent randomness expansion with entangled photons". In: *Nature Physics* 17.4 (Jan. 2021), pp. 452–456. ISSN: 1745-2481. DOI: 10.1038/s41567-020-01153-4. URL: http://dx.doi.org/10.1038/s41567-020-01153-4.

[53] Wen-Zhao Liu et al. "Device-independent randomness expansion against quantum side information". In: *Nature Physics* 17.4 (Feb. 2021), pp. 448–451. ISSN: 1745-2481. DOI: 10.1038/s41567-020-01147-2. URL: http://dx.doi.org/10.1038/s41567-020-01147-2.

[54] Rutvij Bhavsar, Sammy Ragy, and Roger Colbeck. "Improved device-independent randomness expansion rates using two sided randomness". In: *New Journal of Physics* 25.9 (Sept. 2023), p. 093035. ISSN: 1367-2630. DOI: 10.1088/1367-2630/acf393. URL: http://dx.doi.org/10.1088/1367-2630/acf393.

[55] Martin Plesch and Matej Pivoluska. "Device-independent randomness amplification with a single device". In: *Physics Letters A* 378.40 (Aug. 2014), pp. 2938–2944. ISSN: 0375-9601. DOI: 10.1016/j.physleta.2014.08.007. URL: http://dx.doi.org/10.1016/j.physleta.2014.08.007.

[56] Jan Bouda et al. "Device-independent randomness extraction from an arbitrarily weak min-entropy source". In: *Physical Review A* 90.3 (Sept. 2014). ISSN: 1094-1622. DOI: 10.1103/physreva.90.032313. URL: http://dx.doi.org/10.1103/physreva.90.032313.

[57] Max Kessler and Rotem Arnon-Friedman. "Device-Independent Randomness Amplification and Privatization". In: *IEEE Journal on Selected Areas in Information Theory* 1.2 (Aug. 2020), pp. 568–584. ISSN: 2641-8770. DOI: 10.1109/jsait.2020.3012498. URL: http://dx.doi.org/10.1109/jsait.2020.3012498.

[58] Matej Pivoluska et al. "Semi-device-independent random number generation with flexible assumptions". In: *npj Quantum Information* 7.1 (Mar. 2021). ISSN: 2056-6387. DOI: 10.1038/s41534-021-00387-1. URL: http://dx.doi.org/10.1038/s41534-021-00387-1.

[59] Thibault Michel et al. "Real-Time Source-Independent Quantum Random-Number Generator with Squeezed States". In: *Physical Review Applied* 12.3 (Sept. 2019). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.12.034017. URL: http://dx.doi.org/10.1103/physrevapplied.12.034017.

[60] Yu-Huai Li et al. "Quantum random number generation with uncharacterized laser and sunlight". In: *npj Quantum Information* 5.1 (Nov. 2019). ISSN: 2056-6387. DOI: 10.1038/s41534-019-0208-1. URL: http://dx.doi.org/10.1038/s41534-019-0208-1.

[61] David Drahi et al. "Certified Quantum Random Numbers from Untrusted Light". In: *Physical Review X* 10.4 (Dec. 2020). ISSN: 2160-3308. DOI: 10.1103/physrevx.10.041048. URL: http://dx.doi.org/10.1103/physrevx.10.041048.

[62] You-Qi Nie et al. "Experimental measurement-device-independent quantum random-number generation". In: *Physical Review A* 94.6 (Dec. 2016). ISSN: 2469-9934. DOI: 10.1103/physreva.94.060301. URL: http://dx.doi.org/10.1103/physreva.94.060301.

[63] Tao Wu et al. "Measurement-Device-Independent Quantum Random-Number Generator With Source Flaws". In: *IEEE Photonics Journal* 15.6 (Dec. 2023), pp. 1–5. ISSN: 1943-0647. DOI: 10.1109/jphot.2023.3331547. URL: http://dx.doi.org/10.1109/jphot.2023.3331547.

[64] You-Qi Nie et al. "Measurement-device-independent quantum random number generation over 23 Mbps with imperfect single-photon sources". In: *Quantum Science and Technology* 9.2 (Apr. 2024), p. 025024. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ad34f4. URL: http://dx.doi.org/10.1088/2058-9565/ad34f4.

[65] Marco Avesani et al. "Semi-Device-Independent Heterodyne-Based Quantum Random-Number Generator". In: *Physical Review Applied* 15.3 (Mar. 2021). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.15.034034. URL: http://dx.doi.org/10.1103/physrevapplied.15.034034.

[66] Hamid Tebyanian et al. "Semi-device-independent randomness from d-outcome continuous-variable detection". In: *Phys. Rev. A* 104 (6 Dec. 2021), p. 062424. DOI: 10.1103/PhysRevA.104.062424. URL: https://link.aps.org/doi/10.1103/PhysRevA.104.062424.

[67] Ammar Alkassar, Thomas Nicolay, and Markus Rohe. "Obtaining True-Random Binary Numbers from a Weak Radioactive Source". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005, pp. 634–646. ISBN: 9783540320449. DOI: 10.1007/11424826_67. URL: http://dx.doi.org/10.1007/11424826_67.

[68] Rajesh Duggirala, Amit Lal, and Shankar Radhakrishnan. "Radioisotope Decay Rate Based Counting Clock". In: *Radioisotope Thin-Film Powered Microsystems*. Springer New York, 2010, pp. 127–170. ISBN: 9781441967633. DOI: 10.1007/978-1-4419-6763-3_7. URL: http://dx.doi.org/10.1007/978-1-4419-6763-3_7.

[69] Daniel Ruschen et al. "Generation of true random numbers based on radioactive decay". In: *power* 3 (2017), p. 3V.

[70] Kanin Aungskunsiri et al. "Random number generation from a quantum tunneling diode". In: *Applied Physics Letters* 119.7 (Aug. 2021). ISSN: 1077-3118. DOI: 10.1063/5.0055955. URL: http://dx.doi.org/10.1063/5.0055955.

[71] Scott A Wilber and Luis Araujo. "The ComScire® CryptoStrong™ Random Number Generator." In: (2019). URL: https://comscire.com/files/whitepaper/CryptoStrong_Whitepaper.pdf.

[72] G. E. Katsoprinakis et al. "Quantum random number generator based on spin noise". In: *Physical Review A* 77.5 (May 2008). ISSN: 1094-1622. DOI: 10.1103/physreva.77.054101. URL: http://dx.doi.org/10.1103/physreva.77.054101.

[73] Thomas Jennewein et al. "A fast and compact quantum random number generator". In: *Review of Scientific Instruments* 71.4 (Apr. 2000), pp. 1675–1680. ISSN: 1089-7623. DOI: 10.1063/1.1150518. URL: http://dx.doi.org/10.1063/1.1150518.

[74] André Stefanov et al. "Optical quantum random number generator". In: *Journal of Modern Optics* 47.4 (Mar. 2000), pp. 595–598. DOI: 10.1080/09500340008233380. URL: https://doi.org/10.1080%2F09500340008233380.

[75] Osung Kwon, Young-Wook Cho, and Yoon-Ho Kim. "Quantum random number generator using photon-number path entanglement". In: *Applied Optics* 48.9 (Mar. 2009), p. 1774. ISSN: 1539-4522. DOI: 10.1364/ao.48.001774. URL: http://dx.doi.org/10.1364/ao.48.001774.

[76] Harald Fürst et al. "High speed optical quantum random number generation". In: *Optics express* 18.12 (2010), pp. 13029–13037. DOI: 10.1364/OE.18.013029. URL: https://doi.org/10.1364/OE.18.013029.

[77] Simone Tisa et al. "High-speed quantum random number generation using CMOS photon counting detectors". In: *IEEE Journal of Selected Topics in Quantum Electronics* 21.3 (2015), pp. 23–29. DOI: 10.1109/JSTQE.2014.2375132. URL: https://doi.org/10.1109/JSTQE.2014.2375132.

[78] Gaëtan Gras et al. "Quantum Entropy Model of an Integrated Quantum-Random-Number-Generator Chip". In: *Physical Review Applied* 15.5 (May 2021). DOI: 10.1103/physrevapplied.15.054048. URL: https://doi.org/10.1103%2Fphysrevapplied.15.054048.

[79] Bruno Sanguinetti et al. "Quantum Random Number Generation on a Mobile Phone". In: *Physical Review X* 4.3 (Sept. 2014). ISSN: 2160-3308. DOI: 10.1103/physrevx.4.031056. URL: http://dx.doi.org/10.1103/physrevx.4.031056.

[80] Mario Stipčević and B Medved Rogina. "Quantum random number generator based on photonic emission in semiconductors". In: *Review of scientific instruments* 78.4 (2007), p. 045104. DOI: 10.1063/1.2720728. URL: https://doi.org/10.1063/1.2720728.

[81] James F Dynes et al. "A high speed, postprocessing free, quantum random number generator". In: *Applied physics letters* 93.3 (2008), p. 031109. DOI: 10.1063/1.2961000. URL: https://doi.org/10.1063/1.2961000.

[82] Michael Wahl et al. "An ultrafast quantum random number generator with provably bounded output bias based on photon arrival time measurements". In: *Applied Physics Letters* 98.17 (2011), p. 171105. DOI: 10.1063/1.3578456. URL: https://doi.org/10.1063/1.3578456.

[83] Michael Wahl et al. "Addendum:"An ultrafast quantum random number generator with provably bounded output bias based on photon arrival time measurements"[Appl. Phys. Lett. 98, 171105 (2011)]". In: *Applied Physics Letters* 101.15 (2012), p. 171105. DOI: 10.1063/1.4758702. URL: https://doi.org/10.1063/1.4758702.

[84] Andrea Stanco et al. "Certification of the efficient random number generation technique based on single-photon detector arrays and time-to-digital converters". In: *IET Quantum Communication* 2.3 (2021), pp. 74–79. DOI: 10.1049/qtc2.12018. URL: https://doi.org/10.1049/qtc2.12018.

[85] Nicola Massari et al. "A monolithic SPAD-based random number generator for cryptographic application". In: *ESSCIRC 2022- IEEE 48th European Solid State Circuits Conference (ESSCIRC)*. IEEE, Sept. 2022. DOI: 10.1109/esscirc55480.2022.9911498. URL: https://doi.org/10.1109%2Fesscirc55480.2022.9911498.

[86] T. Symul, S. M. Assad, and P. K. Lam. "Real time demonstration of high bitrate quantum random number generation with coherent laser light". In: *Applied Physics Letters* 98.23 (June 2011). ISSN: 1077-3118. DOI: 10.1063/1.3597793. URL: http://dx.doi.org/10.1063/1.3597793.

[87] Qiang Zhou et al. "Practical quantum random-number generation based on sampling vacuum fluctuations". In: *Quantum Engineering* 1.1 (Mar. 2019), e8. ISSN: 2577-0470. DOI: 10.1002/que2.8. URL: http://dx.doi.org/10.1002/que2.8.

[88] Cédric Bruynsteen et al. "100-Gbit/s Integrated Quantum Random Number Generator Based on Vacuum Fluctuations". In: *PRX Quantum* 4.1 (Mar. 2023). ISSN: 2691-3399. DOI: 10.1103/prxquantum.4.010330. URL: http://dx.doi.org/10.1103/prxquantum.4.010330.

[89] Bing Qi et al. "High-speed quantum random number generation by measuring phase noise of a single-mode laser". In: *Optics Letters* 35.3 (Jan. 2010), p. 312. ISSN: 1539-4794. DOI: 10.1364/ol.35.000312. URL: http://dx.doi.org/10.1364/ol.35.000312.

[90] J.-R. Álvarez et al. "Random number generation by coherent detection of quantum phase noise". In: *Optics Express* 28.4 (Feb. 2020), p. 5538. ISSN: 1094-4087. DOI: 10.1364/oe.383196. URL: http://dx.doi.org/10.1364/oe.383196.

[91] Philip J. Bustard et al. "Quantum random bit generation using stimulated Raman scattering". In: *Optics Express* 19.25 (Nov. 2011), p. 25173. DOI: 10.1364/oe.19.025173. URL: https://doi.org/10.1364%2Foe.19.025173.

[92] M. J. Collins et al. "Random number generation from spontaneous Raman scattering". In: *Applied Physics Letters* 107.14 (Oct. 2015). ISSN: 1077-3118. DOI: 10.1063/1.4931779. URL: http://dx.doi.org/10.1063/1.4931779.

[93] Frédéric Monet, Jean-Sébastien Boisvert, and Raman Kashyap. "A simple high-speed random number generator with minimal post-processing using a random Raman fiber laser". In: *Scientific Reports* 11.1 (June 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-021-92668-0. URL: http://dx.doi.org/10.1038/s41598-021-92668-0.

[94] Alfréd Rényi. "On measures of entropy and information". In: *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*. Vol. 4. University of California Press. 1961, pp. 547–562.

[95] Robert Konig, Renato Renner, and Christian Schaffner. "The Operational Meaning of Min- and Max-Entropy". In: *IEEE Transactions on Information Theory* 55.9 (Sept. 2009), pp. 4337–4347. DOI: 10.1109/tit.2009.2025545.

[96] Emil Simion. "The relevance of statistical tests in cryptography". In: *IEEE Security & Privacy* 13.1 (2015), pp. 66–70. DOI: 10.1109/MSP.2015.16.

[97] Ronald L. Wasserstein and Nicole A. Lazar. "The ASA Statement on p-Values: Context, Process, and Purpose". In: *The American Statistician* 70.2 (Apr. 2016), pp. 129–133. ISSN: 1537-2731. DOI: 10.1080/00031305.2016.1154108. URL: http://dx.doi.org/10.1080/00031305.2016.1154108.

[98] N. Smirnov. "Table for Estimating the Goodness of Fit of Empirical Distributions". In: *The Annals of Mathematical Statistics* 19.2 (June 1948), pp. 279–281. ISSN: 0003-4851. DOI: 10.1214/aoms/1177730256. URL: http://dx.doi.org/10.1214/aoms/1177730256.

[99] *NIST SP 800-22: Documentation and Software*. https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software. 2024. (Last accessed 2024/11/14).

[100] *Improved version of the NIST Statistical Test Suite (STS)*. https://github.com/arcetri/sts. 2024. (Last accessed 2024/11/14).

[101] Katarzyna Anna Kowalska, Davide Fogliano, and Jose Garcia Coello. "On the revision of NIST 800-22 Test Suites". In: *Cryptology ePrint Archive* (2022).

[102] *dieharder by Robert G. Brown, Duke University Physics Department, Durham, NC 27708-0305 Copyright Robert G. Brown, 2019*. https://webhome.phy.duke.edu/~rgb/General/dieharder.php. 2024. (Last accessed 2024/11/14).

[103] G. Marsaglia. *"The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness". Florida State University. 1995. Archived from the original on 2016-01-25.* https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/. (Last accessed 2024/11/14).

[104] Pierre L'Ecuyer and Richard Simard. "TestU01: A C Library for Empirical Testing of Random Number Generators". In: *ACM Transactions on Mathematical Software* 33.4 (Aug. 2007), pp. 1–40. DOI: 10.1145/1268776.1268777. URL: https://doi.org/10.1145/1268776.1268777.

[105] *ENT: A Pseudorandom Number Sequence Test Program*. https://www.fourmilab.ch/random/. 2024. (Last accessed 2024/11/14).

[106] Roy J Glauber. "Coherent and incoherent states of the radiation field". In: *Physical Review* 131.6 (Sept. 1963), pp. 2766–2788. DOI: 10.1103/PhysRev.131.2766. URL: https://doi.org/10.1103/PhysRev.131.2766.

[107] M. C. Teich and B. E. A. Saleh. "Effects of random deletion and additive noise on bunched and antibunched photon-counting statistics". In: *Optics Letters* 7.8 (Aug. 1982), p. 365. DOI: 10.1364/ol.7.000365. URL: https://doi.org/10.1364/ol.7.000365.

[108] Ágoston Schranz, Balázs Solymos, and Miklós Telek. "Stochastic performance analysis of a time-of-arrival quantum random number generator". In: *IET Quantum Communication* (Dec. 2023). ISSN: 2632-8925. DOI: 10.1049/qtc2.12080. URL: http://dx.doi.org/10.1049/qtc2.12080.

[109] Catherine Forbes et al. *Statistical Distributions*. Wiley, Nov. 2010. ISBN: 9780470627242. DOI: 10.1002/9780470627242. URL: http://dx.doi.org/10.1002/9780470627242.

[110] Ágoston Schranz and Eszter Udvary. "Mathematical analysis of a quantum random number generator based on the time difference between photon detections". In: *Optical Engineering* 59.4 (2020), p. 044104. DOI: 10.1117/1.OE.59.4.044104. URL: https://doi.org/10.1117/1.OE.59.4.044104.

[111] William R. Leo. *Techniques for Nuclear and Particle Physics Experiments: A How-to Approach*. Springer Berlin Heidelberg, 1994. ISBN: 9783642579202. DOI: 10.1007/978-3-642-57920-2. URL: http://dx.doi.org/10.1007/978-3-642-57920-2.

[112] Jörg W. Müller. "Generalized dead times". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 301.3 (Mar. 1991), pp. 543–551. ISSN: 0168-9002. DOI: 10.1016/0168-9002(91)90021-h. URL: http://dx.doi.org/10.1016/0168-9002(91)90021-h.

[113] Ágoston Schranz. "Optical solutions for quantum key distribution transmitters". PhD thesis. Budapest University of Technology and Economics, 2021. URL: http://hdl.handle.net/10890/16991.

[114] Balázs Solymos and László Bacsárdi. "Secure post-processing for non-ideal photon arrival time based quantum random number generator". In: *Infocommunications journal* 16.1 (2024), pp. 12–19. ISSN: 2061-2079. DOI: 10.36244/icj.2024.1.2. URL: http://dx.doi.org/10.36244/icj.2024.1.2.

[115] Olive Jean Dunn. "Multiple Comparisons among Means". In: *Journal of the American Statistical Association* 56.293 (1961), pp. 52–64. DOI: 10.1080/01621459.1961.10482090. URL: https://doi.org/10.1080/01621459.1961.10482090.

[116] Rupert G. Miller. *Simultaneous Statistical Inference*. Springer New York, 1981. ISBN: 9781461381228. DOI: 10.1007/978-1-4613-8122-8. URL: http://dx.doi.org/10.1007/978-1-4613-8122-8.

[117] Ágoston Schranz, Balázs Solymos, and Miklós Telek. "Experimental Time-of-Arrival Quantum Random Number Generation with Dead Time Overestimation". In: *2024 7th International Balkan Conference on Communications and Networking (BalkanCom)*. IEEE, June 2024. DOI: 10.1109/balkancom61808.2024.10557176. URL: http://dx.doi.org/10.1109/balkancom61808.2024.10557176.

[118] F. N. David and N. L. Johnson. "The Probability Integral Transformation When Parameters are Estimated from the Sample". In: *Biometrika* 35.1/2 (May 1948), p. 182. ISSN: 0006-3444. DOI: 10.2307/2332638. URL: http://dx.doi.org/10.2307/2332638.

[119] Robert B Davies. *Exclusive OR (XOR) and hardware random number generators*. https://www.robertnz.net/pdf/xor2.pdf. 2002. (Last accessed 2024/11/14).

[120] R. Impagliazzo, L. A. Levin, and M. Luby. "Pseudo-random generation from one-way functions". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing - STOC '89*. STOC '89. ACM Press, 1989, pp. 12–24. DOI: 10.1145/73007.73009. URL: http://dx.doi.org/10.1145/73007.73009.

[121] Xiaoguang Zhang et al. "FPGA implementation of Toeplitz hashing extractor for real time post-processing of raw random numbers". In: *2016 IEEE-NPSS Real Time Conference (RT)*. IEEE, June 2016. DOI: 10.1109/rtc.2016.7543094. URL: https://doi.org/10.1109/rtc.2016.7543094.

[122] Maurício J. Ferreira, Nuno A. Silva, and Nelson J. Muga. "Efficient Randomness Extraction in Quantum Random Number Generators". In: *Anais do II Workshop de Comunicação e Computação Quântica (WQuantum 2022)*. Sociedade Brasileira de Computação, May 2022. DOI: 10.5753/wquantum.2022.223591. URL: https://doi.org/10.5753/wquantum.2022.223591.

[123] Balázs Solymos and László Bacsárdi. "Real-time Processing System for a Quantum Random Number Generator". In: *Infocommunications journal* 12.1 (2020), pp. 53–59. ISSN: 2061-2079. DOI: 10.36244/icj.2020.1.8. URL: http://dx.doi.org/10.36244/icj.2020.1.8.

[124] Balázs Solymos, Ágoston Schranz, and Miklós Telek. "Correlation avoidance in single-photon detecting quantum random number generators by dead time overestimation". In: *EPJ Quantum Technology* 11.1 (Sept. 2024). ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-024-00272-8. URL: http://dx.doi.org/10.1140/epjqt/s40507-024-00272-8.

[125] Balázs Solymos and László Bacsárdi. "Affordable statistical testing for quantum random number generators in space applications". In: *Selected papers of the 6th International Conference on Research, Technology and Education of Space (H-SPACE 2020)*. 2020. URL: https://m2.mtmt.hu/api/publication/31833498.

[126] Balázs Solymos and László Bacsárdi. "Statisztikai eszközök űrbéli kvantum véletlenszámgenerátorokhoz". In: *Magyar Űrkutatási Fórum 2021 - Az előadások összefoglalói*. 2021, pp. 43–43. URL: https://m2.mtmt.hu/api/publication/32516853.

[127] Balázs Solymos and László Bacsárdi. "Statistical testing tools for Quantum Random Number Generators onboard CubeSats". In: *IAF Space Communications and Navigation Symposium 2021 at the 72nd International Astronautical Congress, IAC 2021*. Dubai, UAE, Oct. 2021, pp. 1–6. URL: https://m2.mtmt.hu/api/publication/32519888.

[128] Balázs Solymos and László Bacsárdi. "Efficiency improvement of photon arrival time based quantum random number generator with hashing". In: *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. Vol. 78. IEEE, May 2023, pp. 000053–000058. DOI: 10.1109/saci58269.2023.10158613. URL: http://dx.doi.org/10.1109/saci58269.2023.10158613.

[129] Balázs Solymos, Tamás Nepusz, and László Bacsárdi. "Routing the quantum internet in large LEO satellite constellations". In: *2023 IEEE 23rd International Symposium on Computational Intelligence and Informatics (CINTI)*. Vol. 37. IEEE, Nov. 2023, pp. 000397–000402. DOI: 10.1109/cinti59972.2023.10382070. URL: http://dx.doi.org/10.1109/cinti59972.2023.10382070.

[130] Balázs Solymos and László Bacsárdi. "Probability integral based post-processing for photonic quantum random number generators". In: *2025 IEEE Cybernetics & Informatics (K&I)*. IEEE, 2025. (submitted).

List of Figures

List of Tables

List of Abbreviations

BB84	Bennett–Brassard 1984 (protocol)
BS	Beam splitter
CDF	Cumulative distribution function
CPU	Central processing unit
CW	Continuous wave
DTD	Discretized time differences
FFT	Fast Fourier transform
FPGA	Field-programmable gate array
FWHM	Full width at half maximum
GPU	Graphics processing unit
MSE	Mean square error
NIST	National Institute of Standards and Technology
NIST STS	National Institute of Standards and Technology Statistical Test Suite
OTP	One-time pad (protocol)
PDF	Probability density function
PMF	Probability mass function
PMT	Photomultiplier tube
PPP	Poisson point process
PRNG	Pseudorandom number generator
QKD	Quantum key distribution
QRNG	Quantum random number generator
RNG	Random number generator

RSA	Rivest–Shamir–Adleman (cryptosystem)
SCV	Squared coefficient of variation
SPD	Single photon detector
TDC	Time-to-digital converter
ToA	Time-of-arrival
TRNG	True random number generator
uC	Microcontroller
vDTD	virtual DTD (discretized dead time)
VOA	Variable optical attenuator
XOR	Exclusive or