Quantum Random-Number Generators: Practical Considerations and Use Cases

REPORT

Dr. Marco Piani
Dr. Michele Mosca
Brian Neill, CISSP

January 2021

evolution

TABLE OF CONTENTS

Quantum Random-Number Generators: Practical Considerations and Use Cases

Executive Summary

Building resilience against future threats is key to ensure that vital parts of the economy and of society can continue to thrive, but it very often does not receive enough attention and focus, be it at the level of individuals, or at that of companies and institutions. The issue is that investments for protection from medium- to long-term risks are somewhat natural to overlook or dismiss, given more pressing present issues, and the costs may not seem immediately justified. It often takes some bad incidents to incentivize the necessary proactive steps.

At present, we are witnessing a proliferation of cyberattacks, and an emerging greater awareness of their negative impact on the economy and society. Furthermore, with the COVID-19 pandemic, we are experiencing the devastating effects of the lack of preparedness for a systemic threat that is known to exist, but whose precise timing and form may be unknown. These two realities, and the deeper societal awareness of these issues, should boost the recognition of the value of designing cyber-resilience into platforms and systems.

A good place to start is the use of more robust and reliable building blocks for information systems, which leads to systems that are more resilient to current and emerging threats.

Random bits that cannot be predicted even partially and with large resources—and that are unknown to adversaries—are one of the most fundamental resources used in cryptography and cybersecurity.

There have been many instances of the failure of Random Number Generators (RNGs), from poor implementation to the suspected compromise of vendors.

Quantum Random-Number Generators (QRNGs), which exploit the built-in unpredictability of quantum mechanics, promise to provide enhanced security for this fundamental building block.

This report is about understanding the features and advantages of QRNGs, and the scenarios in which they may or may not be worth using in practice in the short, medium, or long term. It is intended as a guide for those making design, architecture, and policy decisions that may enable, or dismiss, the use of QRNGs in protecting cyber-systems.

The report is based on relevant literature and, most importantly, on the input of several experts in the fields of RNGs and QRNGs, both from industry and academia, who were asked to express opinions about the features that set QRNGs apart from traditional (also known as "classical") RNGs, especially in terms of providing some advantage of the former over the latter in relevant use-case scenarios.

Features

Summary of findings

The main features that set QRNGs apart from traditional RNGs are the following:

- They rely on quantum features to *generate new, physically-fundamental randomness.* This is in stark contrast to traditional true RNGs, which are based on unknown—but, in principle, knowable—information implicitly pre-existent in a physical device, a kind of information which could even be potentially implanted.
- They use physical processes that, despite being *surprising* because they are based on quantum physics, are nonetheless *simple* compared to the classical physical processes at the basis of classical RNGs, which need to be *complex enough* to avoid predictability.

In addition:

- By relying on basic principles of physics, or relatively weak computational assumptions, a subclass of QRNGs that are so-called *device-independent* permit the certification and validation that the numbers are freshly generated—randomly and unknowingly to adversaries, thus ensuring that the numbers are also private, independently of the specifics of the underlying technological implementation.

The ability to certify that random numbers are also private is a feature unique to that subclass of QRNGs. This feature is conceptually impossible in traditional RNGs, where one must always consider the possibility that the random numbers produced were somewhat implanted or known in advance.

Advantages

Advantages of using devices with these features are the following:

- They reduce or remove the risk that the random data that is being provided by the RNG is known, or partly known, in advance to some adversary, both thanks to the fundamental randomness of quantum processes and because the processes at play, despite being quantum, are relatively simple and resilient against external interference.
- Given the relative simplicity of their underlying quantum processes, they may permit an almost real-time validation or health check that the randomness generation process is functioning properly.
- Device-independent QRNGs can not only allow one to validate randomness, but also to certify that it is private randomness; the produced random numbers are ensured to be unknown to an adversary.
- When used in parallel with other RNGs and combined/composed in a secure manner, they offer a qualitatively different source of randomness that mitigates the risk that the other RNG(s) being used might have been compromised.
- At a more abstract level, including a more secure RNG in platforms and tools may generate greater trust in those tools and platforms among end users.

Threat scenarios and use cases

What kinds of threat scenarios and use cases might seriously benefit from these advantages?

- Those protecting *high-value* assets and critical systems:
  This may include confidential information to be shared in encrypted form or authentication for access to strategic applications and databases, at the military, governmental, or enterprise level, particularly in the financial sector. The private nature of the randomness (sometimes referred to as "privacy" in QRNG literature)

provided by device-independent QRNGs may assume particular importance for such assets and systems.
- Those where the ability to ensure the random nature of the numbers generated is paramount to *guarantee fairness* and to *establish or maintain trust*:
  Examples include lotteries or sporting events/competitions, the allocation of scarce resources (in this period of time, this may include newly developed vaccines), the selection of juries in trials, and random checks at strategic facilities (like border or airport controls that should not be biased).

While these are the most serious and compelling cases, a continued increase in performance and a continuous decrease in size and cost could allow QRNGs to be adopted more widely—even at the level of smartphones, for example—adding a layer of security and contributing to an increased level of trust.

From the perspective of enterprise adoption, an observation that was repeatedly made by the surveyed experts is that many applications use hardware security modules (HSMs) to store their most critical information. Thus, assuring that HSMs can easily integrate QRNGs into their platforms and securely combine different sources of randomness is a critical way to deliver (or at least facilitate the delivery of) the additional security features of QRNGs to a wide host of potential applications today and into the future.

Another point repeatedly made by surveyed experts was that, given the growing concerns about cybersecurity and the necessity to depend on systems built out of components that may not be directly and individually trusted, there will be increasing value in trustworthy tools that may be leveraged to enable the use of less-secure infrastructure without compromising security. A general example along these lines is end-to-end encryption, which allows one to retain confidentiality even though the transport tool, e.g., the network, is not trusted. In such circumstances, it is ideal to have a *foundation* for trust which is as independent from assumptions as possible, including assumptions about the trustworthiness of the manufacturer of the underlying technology.

The interest of end users who transparently use randomness, e.g., consumers using cellular phones or online banking, is naturally quite limited. On the other hand, the interest of vendors who may use QRNGs in the products and services they sell is growing, although still limited. Such growth has been more rapid in the telecommunications, defence, and financial sectors, where security concerns have driven the adoption of QRNGs. Nonetheless, the majority of vendors focus on meeting institutional security criteria at a low price, for which standard RNGs often suffice, rather than trying to provide the highest quality of randomness.

There is the expectation that the QRNG take-up will increase considerably in the next 10 years. On top of the conscious adoption in areas where security is of the highest concern, this process may be sped up by the general appeal of the fast-developing quantum technologies and by further improvement in the size, cost, and performance of QRNGs, which may lead to a widespread adoption even in devices like smartphones.

QRNGs are a relatively new emerging tool that should be given proper consideration for the added value and unique properties they may offer, particularly in products and services that protect high-value assets for a long time.

1. Purpose

Quantum Random-Number Generators (QRNGs) bring additional choices to the array of random-number generators already available for real-world deployment. The "right" choice of Random-Number Generator (RNG) depends on several considerations:

- Size of RNG device(s)
- Cost of RNG device(s)
- Rate of random-number generation
- Stability and reliability under given working conditions
- Intended use of the random numbers
- Level of security and secrecy needed
- Trust in the vendors
- Trust in the technical equipment

Technical parameters related to cost, speed, and stability will vary with time, while use-case parameters related to the required level of trust tend to remain firm.

This report explores some of the trust and technical trade-offs offered by QRNGs compared to other options available in the market, most importantly as seen by various subject-matter experts who were surveyed and interviewed. The list of questions asked, as well as the roster of experts who contributed to this report, is provided in Appendix 7.3.

2. Background

While it may not be immediately apparent, randomness is a highly non-trivial concept. What does it mean when a number is random?

The attribute of being random applies more correctly to a sequence of numbers—without loss of generality, assuming just the bit values 0 and 1—rather than to individual numbers. Randomness is strictly related to lack (and ideally, impossibility) of predictability. A simple test can be used to see if a sequence of numbers is random or not: compress it using zip compression on a PC. If you can compress a file of data and it shrinks in size, it means that the compression tool found a recurring pattern in the data, removed redundant information, and plans to add it back in later. There is predictability in the data, which therefore is not random.[1]

Figure 1. The output of a random-number generator can be thought of as sequences of bits.

In this sense, it might be better to speak of the randomness of a *source*. Ideally, one would like to have access to a source that produces random-bit strings, where the values of the bits can be described by independent and identically distributed (in short, *i.i.d.*) random variables: the value of each bit is independent of past or future bit values, and it is 0 or 1 with the same probabilities as the other bits. The best scenario would be one where each bit is *unbiased*—that is, equally likely to be a 0 or a 1.

[1] The impossibility of compression, independently of the algorithm used for compression, can actually be considered the defining trait of randomness, according to the approach to complexity by Gregory Chaitin and Andrey Kolmogorov; a formal discussion of these concepts is beyond the scope of this report.

It is nearly impossible to establish whether a source of bit strings is actually random. As long as an i.i.d. source is not constant, that is, as long as it does not produce exclusively 0s or 1s, any output bit string of whatever fixed length can be generated, including those that contain only 1s or only 0s. Indeed, in the case where the i.i.d. source is unbiased, any string of the same length is equally probable. For example, the strings 00000, 11111, and 01001 are all equally probable.

Here, then, is the challenge: Suppose our source produces a specific string. How can we be at least *confident* that the source is *actually* random if any string is equally likely? The answer is that there are other properties of the string that we can analyze. For example, we can try to identify patterns or global properties, like the weight of the string, that is, how many 1s it contains. While all strings may be equally likely, their weight is not. If the source is really i.i.d., then we expect that, for long enough strings, with overwhelming probability, we will observe a string that is *typical*, that is, in the case of an unbiased source, that the number of 0s and 1s in it will be about the same. As an example of a pattern, imagine a source that alternates 0s and 1s. Considered individually, the bits may appear identically distributed and unbiased, but they are not independent: knowledge of the value of one bit and of the *rule* allows one to reconstruct all other bits in the ordered string. The string is actually highly compressible.

Exemplar bit string produced	String probability assuming 0 and 1 equally likely and bits are independent	Fraction of zeros	Displays Pattern?	Is the source random?
0 0 0 0 0 0 0 0 0 0	2^{-10}	100%	Yes	It is reasonable to suspect it is not; it might well be that the source outputs only 0s
0 1 0 1 0 1 0 1 0 1	2^{-10}	50%	Yes	It is reasonable to suspect it is not; it might well be that the source output simply switches between 0 and 1
0 0 1 0 0 0 0 0 0 1	2^{-10}	80%	No	Potentially yes, but it appears but it appears to be biased; it may indicate that 0 and 1 are not equally likely – that is, the assumption of 0 and 1 being actually likely may not hold
1 0 1 1 0 1 0 0 1 0	2^{-10}	50%	No	Potentially yes, and it appears also unbiased; 0 and 1 may really be equally likely

Figure 2. Assuming that 0 and 1 are produced with the same probability and that each bit is produced independently, all strings of the same length are equally probable and should contain approximately the same number of 0s and 1s, appearing without following any pattern. One can never be completely certain that the source is i.i.d., but one can validate those assumptions by checking various properties of the strings produced by the source.

The validation, or rather, corroboration of random sources, is typically performed with standardized tests that look for signs that the strings it produces exhibit some kind of pattern, going from an *excess* of 0s or 1s (that is, a bias) to correlations between various locations of the string. See more details in Section 2.5.

One important measure of randomness is entropy. Roughly speaking, it is the amount of information (measured in bits) necessary to describe a certain string among the set of all possible strings, or, equivalently, the amount of information gained when it is communicated which string was actually generated among the many possibilities. Roughly speaking, bit strings that are not random have limited entropy and can be described with fewer bits than the bit string itself. In the compression example above, this means that a file with high entropy will not compress and will remain the same size, whereas a file with low entropy will compress to a smaller file size.

Sources of true randomness are also known as *entropy sources* (see Section 2.4).

2.1 Random for Whom?

A string of bits produced by a box may pass a large set of randomness tests; yet, one can question whether the randomness is only *apparent*, and one can wonder about other agents who could know the sequence other than the agents authorized to access the box.

For example, if the string is the result of some deterministic physical process or of some deterministic algorithm, then, in principle, somebody could reconstruct or run the same deterministic process and acquire information about the string, all the way down to reconstructing the exact string. An example of these circumstances is someone determining (with better than just guessing chances) where the ball will end in a roulette wheel, based on the initial physical conditions (how fast the wheel is rotating, where the ball is dropped, and so on).

> WARNING: Do NOT use beacon-generated values as cryptographic secret keys!

Figure 3. Warning for users of the "Interoperable Randomness Beacons" NIST Project webpage (retrieved September 2020).

An extreme case of randomness that may be known to many is the one where a long string of bits is simply stored in the source and accessed bit by bit. The content of the box could be public, like in the case of a set of random numbers collected for reference or use in a book or in some digital format. Tables of random numbers with good statistical properties, that is, close to i.i.d., used to be employed relatively widely [1]. Nowadays, there are public services providing sequences of random numbers to the general public, acting thus as *randomness beacons* [2]. Such random numbers have high quality and can have several uses, but they should not be used when the randomness needs to be private (see Figure 3). Private beacons could still be used and shared within a specific organization or a restricted set of authorized users, who may then have access to shared private random numbers.

Another scenario where the property of being private is violated is one where the box can be maliciously influenced externally, or the bits are implanted. In such a case, an agent could not only know the bits, but even control them, and thus, indirectly, whatever process or action that may depend on the use of the output bits. This could, in principle, still be done in such a way that the output string of bits passes a randomness and/or validation test.

2.2 The Importance of Randomness

Randomness finds applications in disparate fields, particularly science, information technology, secure communication, and data handling.

Some common examples of where randomness is used include:

- Information security, particularly in cryptography, to prevent eavesdroppers from guessing cryptographic keys and/or parameters used to protect data
- Secure multi-party computation, in which several parties collaborate to process information while keeping their respective inputs private, e.g., blockchains
- Financial systems, particularly when there is a concern and need to prevent a real financial transaction from being recorded and *replayed* later, like with a chip-and-pin credit card
- Statistical sampling to remove any chance of unconscious bias during scientific experiments or social studies involving polling
- Fault-tolerance testing of IT systems to simulate random failures
- Computer simulations and sampling to provide numerical estimates of quantities that cannot be calculated exactly
- Gaming or gambling, be it at the level of casinos, online gaming or lotteries
- Legal processes, including the selection of jurors from a jury pool

2.3 Quantum Mechanics

While the laws of classical physics describe well the behaviour of physical systems on a macroscopic, everyday scale, quantum mechanics is our best framework to describe the world at the fundamental level, particularly when it comes to atoms and elementary particles.

When quantum mechanics was developed in the first half of the 20th century, it constituted a deep conceptual departure from classical mechanics. In particular, it introduced the notion of quantization of energy (so that one speaks of the discrete levels of energy of, for example, an atom), and, most importantly, it showed that there are fundamental limitations on what we can simultaneously know and predict about properties like the position or the velocity of a physical system— this is famously known as the Heisenberg uncertainty principle.

There are three fundamental quantum features that are relevant in the context of randomness generation (see Appendix 7.1 for more details):

- Quantum mechanics allows systems to be in any *superposition* of two or more classically distinct and distinguishable states.
- The result of most measurements that can be performed on a quantum system is inherently random.
- Quantum systems composed of many subsystems may exhibit a property called *entanglement*, which affects the joint behaviour of those subsystems in ways that are inexplicable within the realm of classical physics.

2.3.1 Quantum computation

Quantum computers are devices that harness the phenomena of quantum mechanics to process information in a profoundly different way than present-day computers. Conventional computers process binary *bits* of information—ones and zeros—while quantum computers process quantum bits, or *qubits*, that can be in a quantum *superposition* of states: not just one or zero *simultaneously*. Controlling such *qubits* can lead to a very large—in some cases, exponential—increase in computing power [3]. Quantum computation is inherently more *fragile* than standard computation. While there is presently a race to build full-fledged quantum computers, this should be seen more like a marathon than a sprint, due to the scientific and technological challenges involved.

2.3.1.1 Quantum computers as a threat

Because of their efficiency, quantum computers, when built, will pose a threat to widely adopted public-key cryptographic schemes like RSA, which rely on the difficulty of certain mathematical problems like factorization. Furthermore, the efficiency of quantum search is such to also suggest increasing the amount of key used, and hence *randomness* consumed, in symmetric cryptography in order to maintain the same level of security.

An important point is that, while the field of quantum computation research is currently not able to build a quantum computer at the scale needed to break classical cryptography, the need to shield today's encrypted data from future quantum attacks may be considered immediate given constraints on the time for which information must remain secure [4].

2.3.1.2 Quantum computation and certifiable randomness

One interesting fact about quantum computers is that, to achieve an advantage with respect to classical computation, they need to realize a relatively high level of superposition and entanglement during their operation. In turn, this means that they can be used to produce quantum states that, once measured, yield a classical output that can be certified as random [5].

2.4 Randomness Generation

The following is a summary of the tools and methods that can be used to generate random numbers. Figure 4 describes the different categories of RNGs available today. QRNGs fall under the category of true physics-based RNGs, exploiting quantum features to generate randomness.

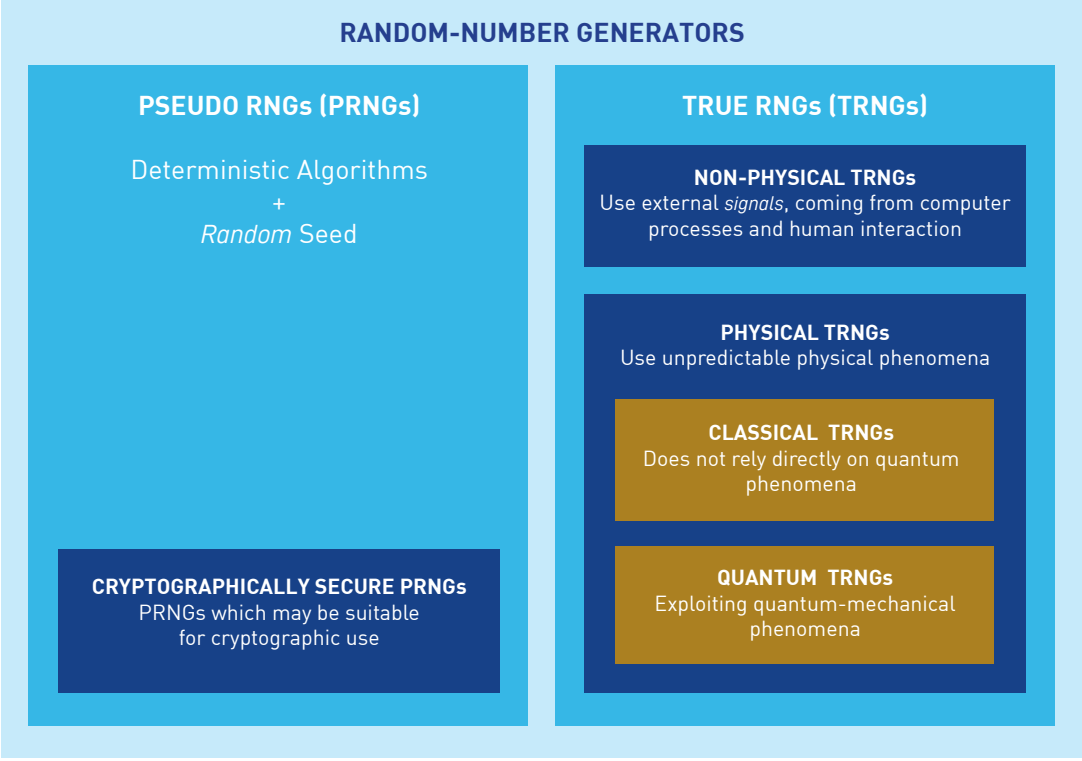

Figure 4. Typologies of random-number generators. The major division is between PseudoRandom-Number Generators (PRNGs), which are used to expand an initial, finite random seed though a deterministic process, and True Random Number Generators (TRNGs), which continuously tap into some entropy source. The properties of such entropy source are what then differentiates different kinds of TRNGs; the source might be whatever non-dedicated external signal for 'non-physical' TRNGs or be some physical system whose purpose is specifically to serve as an entropy source. In turn, the nature of the random physical process or system—either classical or quantum—distinguishes classical TRNGs from quantum TRNGs.

2.4.1 Pseudorandom-number generators

A pseudorandom-number generator (PRNG) is a computer program or function that expands a short string into an arbitrarily long string that looks like random data. A PRNG can be used to efficiently convert a small amount of true randomness into a much larger amount of *effectively random* bits, meaning that it would be difficult for anyone to tell the difference between the output of the PRNG and a string of truly random bits.

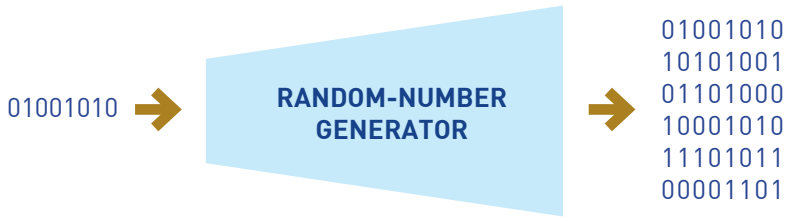

Figure 5. Randomness expansion via a deterministic process. A fairly or completely random string gets deterministically expanded into a longer string that may present some properties typical of randomness, like a correct "typical" weight, but, roughly speaking, each output bit cannot be as random as the input bits. Randomness is being "diluted", as the output longer string can at most be as random (technically, be characterized at most by the same entropy) as the input string, given that a deterministic process cannot increase randomness.

There are many ways of generating pseudorandom numbers, and various statistical tests are available for measuring the strength of a proposed scheme [6]. Some applications may not require all statistical tests to succeed, so suitable PRNGs can be selected depending on the application. The expected performance of a PRNG under a given statistical test is usually hard to determine analytically and often must rely on experimental evidence.

The modern notion for a *perfect* PRNG is based on computational complexity: a PRNG that generates uniform outputs is called *perfect—and only if*—it passes every reasonable statistical test.[2] The class of PRNGs desired in cryptography, also called cryptographically secure PRNGs (CSPRNGs), requires that the PRNG run efficiently and with a negligible bias. In applications such as reciprocal authentication, session key generation, and stream ciphers, the requirement is not just that the sequence of numbers be uniformly random, but that the successive members of the sequence be unpredictable. A question that is still open is whether actual CSPRNGs exist; interestingly, such a question is connected to profound open problems in mathematics and complexity theory, like the famous P vs NP problem. The CSPRNGs constructed so far are only *believed* to be secure generators (with large experimental evidence); a conclusive mathematical proof does not exist.

Famously, one proposed CSPRNG that was standardized, called Dual_EC_DRBG (Dual Elliptic Curve Deterministic Random Bit Generator), was known to be susceptible to having a backdoor that allowed the possessor of the backdoor parameters to predict the outputs of the RNG. Revelations behind its standardization and wide use led to public backlash and distrust of standards bodies [7].

Nonetheless, PRNGs believed to be CSPRNGs are widely used in cryptographic applications for emulating the one-time pad, in what is often called a *computationally secure* realization of the one-time pad. One-time-pad cryptography works by encrypting a plaintext into a cyphertext as follows: One independent random bit is added to each bit of the plaintext

[2] The tests are required to be efficiently run, only tests where the runtime does not grow too fast as function of the size of the string are checked for randomness.

through a XOR addition—that is, through standard addition, but with the less familiar rule of 1+1 = 0. Due to the randomness, the original bit is perfectly masqueraded and can only be decrypted by a receiver who also knows the value of the random bit. Roughly speaking, the one-time pad is the only encryption scheme that is entirely and provably secure, but it comes at the cost of using a key that is as long as the message.

Most of the currently deployed symmetric stream ciphers adopt CSPRNGs for encrypting messages by expanding a small key to the length of the message. Here, the concept of *perfect security* is replaced with the concept of *semantic security*: roughly speaking, an adversary must not be able to compute any information about the plaintext from its ciphertext in a feasible amount of time.

The initial seed for (pseudo)random number generators is typically created by a True Random- Number Generator that uses some entropy source and is then expanded deterministically. In practice, the entropy source is collected from various places on the computer, such as keystroke timing patterns, disk electrical activity, mouse movements, and instantaneous values of the system clock. If not collected properly, the seed becomes the weakest part of a PRNG scheme because if someone can correctly guess the seed that was used, they will know all of the random numbers that were produced by the PRNG. The seed needs to be random, unpredictable, and large enough so that someone cannot guess it using trial-and-error, brute-force methods.

An example of source of entropy bytes for the seed is the `/dev/random` system call on a Linux machine. The `/dev/random` routine gathers environmental noise from various measurements within a Linux computer that could contain entropy. Such entropy can be expanded into pseudorandom numbers using a PRNG.

PRNGs need to be implemented carefully, including the choice of the source of entropy to seed the PRNG. There are plenty of historical examples in which vulnerabilities were discovered that allowed to reconstruct past and future pseudorandom numbers [8–10].

2.4.2 Physical True Random-Number Generators

Despite being *physical* in some sense, disk operation timings and mouse movements are not consistently good randomness sources—and, certainly, they are not designed to be. TRNGs that are considered strictly physical use as an entropy source some physical process that is deemed to produce numbers that are not predictable in any reasonable way and attempt to harvest such entropy as efficiently as possible.

Physical entropy sources do not typically produce a stream of bits that are immediately unbiased and uncorrelated, so the raw stream of bits goes through a process of entropy extraction. Such extraction ideally produces a stream of bits that can be considered fully random and will pass statistical randomness tests.

The entropy extraction reduces the number of bits while increasing their randomness and/or unpredictability. In general, it also requires consuming some initial independent randomness. Depending on how random the original entropy source itself is, the rate of extraction may be very low. A good entropy source is one that produces bits that are highly random from the start or that at least produces enough raw bits with enough entropy that, even accounting for the entropy extraction, a good entropy source ends up with a high rate of high-quality random bits.

One important note, from both a conceptual and a practical point of view, is that, when considering TRNGs, we may be able to check the validity and proper functioning of the physical process that acts as entropy source separately from (or in conjunction with)

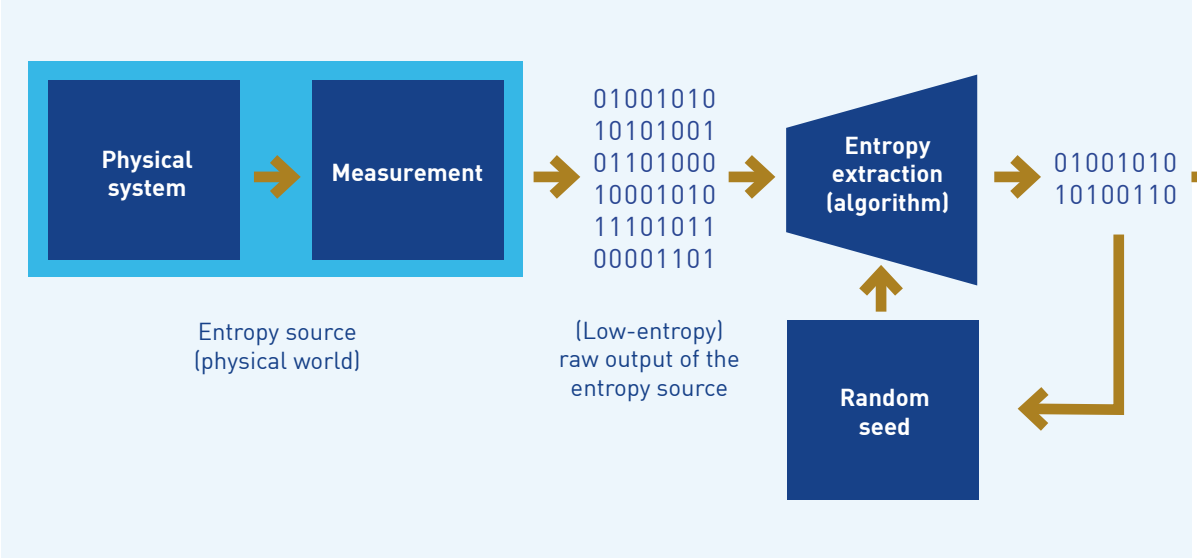

Figure 6. Structure of a True Random-Number generator based on the extraction of randomness from some physical process. Notice the general need for an initial small seed of i.i.d. randomness (which can then be replenished with part of the output of the extraction algorithm).

testing the randomness of the final high-entropy output of the RNG. This check adds to the confidence that:

- the final output really has the properties of a random string, which is relevant as the statistical tests have their own limitations, especially at the time of utilization, and that
- the entropy source has not been compromised in any way, particularly by any malicious agent.

2.4.2.1 Classical/Traditional

For traditional physical TRNGs, the nature of the entropy source is classical; the underlying physical system is *complicated* enough (like a chaotic system) that a computer cannot simulate how it works in order to predict its exact behaviour.

Coin tossing, dice rolling, and roulette turning are all familiar processes that allow the generation of random numbers. Unfortunately, such processes are slow in the generation of random numbers, are not very stable or steady, cannot really be run continuously, and cannot be easily embedded in relevant systems like computers.

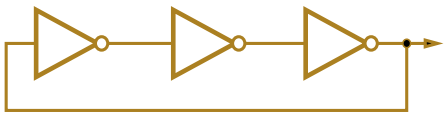

Figure 7. Scheme of a ring oscillator: An odd number of NOT gates is linked sequentially, and the final output is fed back to the first gate. Given a non-zero gate delay, the output oscillates between 0 and 1 with a period that has a random component.

More modern solutions exist. One such example is given by ring oscillators, composed of an odd number of NOT gates disposed sequentially, whose final output is fed back as input. The odd number of gates is such that the input signal gets flipped, leading to an oscillation, but the delays in the operation of the gates are such that the oscillation period contains a random component, which can be used as entropy source. Such ring oscillators can be directly embedded in IT equipment.

2.4.2.2 Quantum

Quantum mechanics offers true fundamental randomness that is not based only on a lack of detailed information. The goal in quantum random-number generators is to utilize set-ups where this fundamental randomness arises in ways that are easy to describe and to quantify.

A traditional example of a quantum randomness source is radioactive decay [11]. Each atom of a radioactive material has some probability of decaying in each time interval, but the exact time of decay is random in the sense that it fundamentally cannot be predicted. Decay timing can be observed through a Geiger-Müller detector and used as a source of entropy, but such a source has several drawbacks, starting from the need to handle radioactive material.

More recent quantum random-number generators use quantum properties of light, for example, where measurement results are similarly unpredictable in principle and the overall core set-up of source and measurement is simple and well understood. An example is a single-photon light source pointed at a partially reflective mirror. Photons pass through the mirror at a measurable rate of 50%; however, quantum mechanics ensures that whether an individual photon will be detected after passing through or after being redirected by the mirror cannot be predicted (see Section 3.1.1).

In these cases, the quantum mechanical description is simple, and the origin of the entropy can be traced back to its quantum mechanical origin easily. In particular realizations of QRNGs, quantum mechanics even offers the verification of the quantum origin through experimental check of fundamental quantum mechanical principles (like non-locality) through the performance of the device itself, rather than relying on validating the design of the entropy generation (for example, see Section 3.1.2).

2.4.2.3 Post-processing/entropy extraction

Post-processing is meant to eliminate biases and correlations from the raw output of the entropy sources, ending up with sequences of bits that are equivalent to those produced by a perfect unbiased i.i.d. source. This comes at three costs:

- A computational cost
- A reduction of the bit rate
- Depending on the scenario, the consumption of an initial *perfect* seed so that, indeed, TRNGs can be generally described as means to expand randomness rather than generating it from scratch

In Figure 8, we provide a simple example, the von Neumann extractor, which allows one to extract a set of equally probable bits from a string of independently generated bits for which 0 and 1 are not necessarily equally probable.

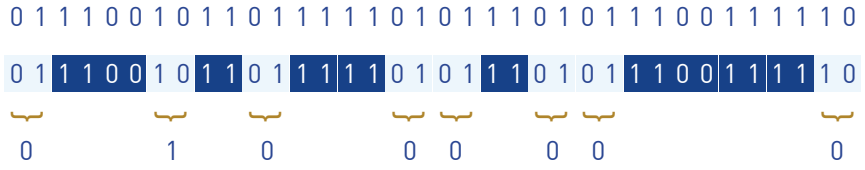

Figure 8. The von Neumann extractor. The first row lists the raw output of the entropy source. We assume that the source is
i.i.d., but it may be biased, e.g., a 1 is more probable than a 0. We can create an unbiased sequence if we group the output bits in pairs (second row). If they are the same, we do not output anything. If, instead, they are different (light blue pair), we output a 0 or a 1 according to the mapping 01 → 0 and 10 → 1. Given that the original entropy source is i.i.d. (by assumption), the two-bit subsequences 01 and 10 of the original string are equally probable. Thus, the final string (third row) is not only i.i.d., but also unbiased.

2.4.2.4 Randomness generation versus randomness expansion

As mentioned in Section 2.4.2.3, for randomness extraction, in general, protocols to generate randomness may themselves use randomness. We will encounter more processes of this kind when discussing designs and implementations of QRNGs in Section 3, in scenarios where, for example, one needs to randomly choose between different measurement options.

So, in general, randomness generation actually is a process of randomness expansion. In the best-case scenario, more randomness is produced than consumed, hence the *expansion*. Nonetheless, not all random numbers have the same properties or the same value. For example, as we discussed earlier, it might be that certain sequences of numbers are random, but not *fresh* or private. So, even in situations where a process or protocol generates randomness by consuming more randomness than it outputs, it can be that the randomness that is consumed is not necessarily private, while the randomness that is produced can be certified to be private.

2.5 Randomness certification

Certifying randomness can be difficult, and there are two main ways to build trust in the randomness of the output of an RNG:

- Testing the randomness of the sequences that it produces
- Knowing and validating the process through which those sequences are produced

Both components of the certification are essential. The first deals with the quality of the random sequences that can be ascertained by checking output sequences through a suite of tests and handling the RNG as a *black box*, that is, without caring about the *inner workings* of the RNG. Unfortunately, no finite test can determine with certainty that an RNG produces random strings, but such tests are designed to be stringent and ensure that the RNG produces strings that have properties that one would expect from random strings.

For example, the National Institute of Standard and Technology (NIST) standardized *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* (known as "SP800-22") [12] to test whether a PRNG produces high-quality random bits. SP800-22 specifies that the tests should target 3 main characteristics:

- **Uniformity:** At any point in the generation of a sequence of random or pseudorandom bits, the occurrence of a zero or one is equally likely, that is, the probability of each is 50%. The expected number of zeros (or ones) is n/2, where n is the number of bits in the sequence.
- **Scalability:** Any test applicable to a sequence can also be applied to subsequences extracted at random. If a sequence is random, then any such extracted subsequence should also be random. Hence, any extracted subsequence should pass any test for randomness.
- **Consistency:** The behaviour of a generator must be consistent across starting values (seeds). It is inadequate to test a PRNG based on the output from a single seed or to test a TRNG on the basis of an output produced from a single physical output.

In terms of unpredictability, a stream of (pseudo)random numbers should exhibit two forms:

- **Forward unpredictability:** If the seed is unknown, the next output bit in the sequence should be infeasible to predict, regardless of any knowledge of previous bits in the sequence.
- **Backward unpredictability:** It should also not be feasible to determine the seed from knowledge of any generated values. No correlation between a seed and any value generated from that seed should be evident; each element of the sequence should appear to be the outcome of an independent random event whose probability is 50%.

3. Quantum Random-Number Generators

In this section, we discuss the theory behind QRNGs and some of their physical realizations.

3.1 Theory

QRNGs exploit quantum properties like superposition and the randomness of measurement outcomes. They produce numbers that can be certified as random to a high degree of confidence based on fundamental laws of quantum physics.

3.1.1 Device-dependent QRNGs

Standard QRNGs are device dependent. They are typically based on the property that a quantum system can exist in a superposition of classically perfectly distinguishable physical states. The simplest quantum system is one that can exist in just two such states, which can aptly be labelled 0 and 1, independently of the exact physical realization. In the same way that a coin (with its *heads* and *tails* sides) or a lightbulb (with its *on* and *off* states) represents the physical realization of a bit, a two-state quantum system is the physical realization of the quantum bit, or qubit.

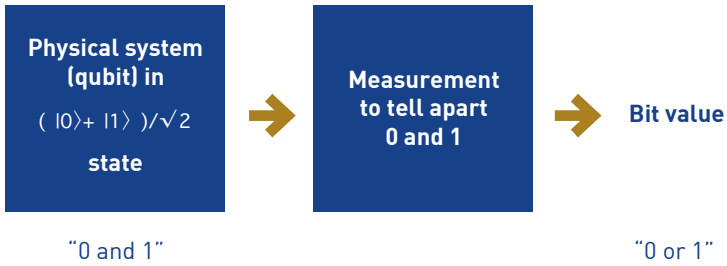

Figure 9. The basic conceptual set-up of a QRNG. A quantum system that is prepared in a superposition of states that can be distinguished by a measurement device will be found in one of such states upon measurement, and such a result will be genuinely random.

If a qubit can be prepared in the superposition state[3] (|0⟩+ |1⟩)/√2, then a measurement aiming at finding out whether the system is in the *classical* state 0 or 1 will produce a random outcome, which, according to quantum physics, is *inherently* random: no one could have known or predicted it, independently of the amount of information or computational power available. Furthermore, ideally, the result is perfectly unbiased.

While both the preparation of the system in the superposition and the measurement process—despite being *quantum*—are relatively simple, one still has to trust the physical or engineering realization of the preparation and measurement, hence the notion of *device dependence.*

3.1.2 Device-independent QRNGs

Quantum mechanics allows one to eliminate the need for trust in the details of the implementation of a QRNG. Scenarios where this is possible are called *device-independent* [13]. There are two main categories of approach for creating a device-independent QRNG: those based on the notion of non-locality and those based on quantum computation. Both approaches allow for the generation of *fresh and private* randomness.

3.1.2.1 Based on non-locality

Quantum *non-locality* is a feature of quantum entanglement that allows the results of measurements separately performed to be correlated in ways that are incompatible with any *classical explanation* where the results of the measurements could have been known or

[3] See Appendix 7.1 for details on the so-called ket notation, e.g. |0⟩, to denote the state of a system.

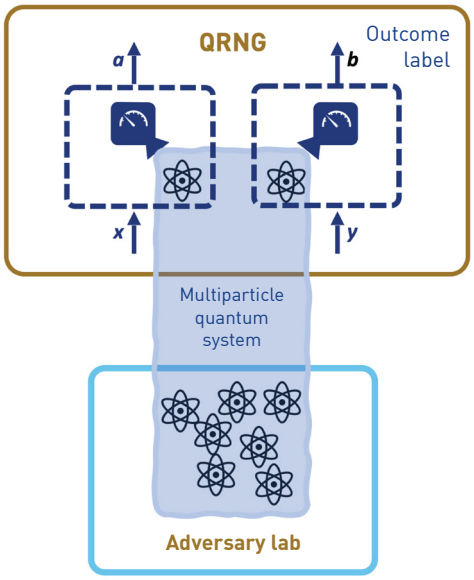

Figure 10. Conceptual basis of a device-independent QRNG. Within the QRNG (gold box) region, two or more black boxes (dashed blue boxes) accept classical inputs x and y and produce classical outputs a and b. The inputs prescribe which measurements to perform on some parts of a multiparticle quantum system. The statistical relations between input and outputs can be used to validate the fact that the subsystems subject to measurement are entangled and that they do not actually share significant correlation or physical connection with any other quantum system that may be in the hands of an adversary (light blue box), and which may have previously interacted with the subsystems being measured.

predicted in advance (see Appendix 7.1.2.2). Such *stranger & stronger* correlations can be used to certify that the outcomes of the measurements are genuinely random. Besides randomness, one can also certify that the numbers are private: the random numbers could not have been known in advance, and, if the numbers are not leaked after being produced, no malicious agent can know them.

Such certifications can be proven to hold under weak assumptions about the details, precision, and control of both the preparation of the physical systems to be measured and the measurements performed; in particular, assuming the validity of quantum mechanics[4], one does not need to rely on the knowledge and precision of the inner workings of preparation and measurement devices, as long as they produce results that can be exploited.

One only needs to rely on the relation between classical inputs and classical outputs of *boxes*, without knowing or caring about what really goes on inside such boxes (see Figure 10). Specifically, based solely on such input/output relations and on assuming the validity of quantum mechanics, it is possible to certify that some physical systems inside the box must have been entangled with each other and not significantly correlated with *any* other system outside the box. This means that even an all-powerful[5] adversary with full quantum capabilities to operate on any other system in the universe cannot know or learn the random numbers that are generated in this way.

Technically speaking, this device-independent generation of randomness exploits the violation of so-called *Bell inequalities*, which are tests for the non-locality of quantum

[4] One can further relax the conditions; it is sufficient that only the *no-signalling principle*, which postulates that information cannot be transmitted instantaneously or faster than light, hold. Quantum mechanics respects this principle, but it is conceptually striking that randomness can be certified even without trusting that we have perfect knowledge of physics.

[5] This is valid in abstract principle within the model where the QRNG does not leak any information. It should be clear that an all-powerful adversary might have ways to "peek" inside the QRNG through whatever infinitesimal leakage of information.

mechanics. Not only do the correlations exhibited by the boxes need to be strong enough to violate a Bell inequality, but such a violation must happen under some conditions, which, if not met, introduce *loopholes* that, in principle, invalidate the certification of the randomness generation.

Such conditions are made very clear in Ref. [13]:

1. The black-box devices should not be correlated with the inputs *x* and *y*.
2. The black-box devices cannot communicate during the measurement process, that is, during the generation of the outputs *a* and *b* after the input of *x* and *y*.

In Section 3.2.2, we show how these conditions can be ensured in practice. This comes at a cost in terms of the real-life size of the gold-box QRNG of Figure 10, which must accommodate distances large enough for light to take a non-negligible time to travel across it. Notice also that the inputs *x* and *y* need to be chosen randomly, which means that device-independent QRNGs based on Bell inequalities are tools that expand randomness, rather than simply creating it.

We remark that it is possible to utilize set-ups that are *semi-device-independent*, that is, where part, and only a part, of the set-up is trusted or where some of the non-locality conditions above are relaxed (see Section 3.2.2.1). Note that semi-device-independent set-ups may also be considered for systems that do not use correlations, for example, by varying in a known way the measurement applied on some physical system that was prepared in an *unknown way* [14].

With respect to semi-device-independence, one of the experts comments:

I think a half-way house to [device independence (DI)] is probably a good practical solution in the medium term. For instance, one could run a DI protocol, but making the fair sampling assumption [Note: this is the assumption that the events that the detectors reveal are a fair sampling of all the events]. The result is not fully device independent but has a large amount of self-checking (one of the advantages of DI), requiring a smaller amount of characterization of the devices and thus increasing security. The advantages of this are that the requirement on the detectors is lower and, hence, the costs can be much lower. – Roger Colbeck

3.1.2.2 Based on quantum computation

Quantum computers are still under development, but relatively fast progress is being made where several companies have demonstrated a relatively high level of control on quantum systems. The era of Noisy Intermediate-Scale Quantum (NISQ) computing is quickly approaching, where systems with tens to hundreds of qubits may run processes whose results or outputs are beyond the reach of standard computers, including supercomputers.

One potential use for NISQ computing is to generate *fresh* random numbers that are unknown in advance to anybody, including someone who has direct access to the quantum computer. Figure 11 illustrates how this would work:

1. A client with only access to a classical computer generates a random challenge.
2. The challenge is sent to the remote quantum computer; a valid response is based on running a computation based on the challenge.
3. The quantum computer sends a response within a very short time, in particular a time within which it would be impossible to calculate the same response by means of a classical computer.
4. The challenge is repeated as needed, and the responses are checked against a test.
5. If the test is passed, randomness can be extracted from the set of responses.

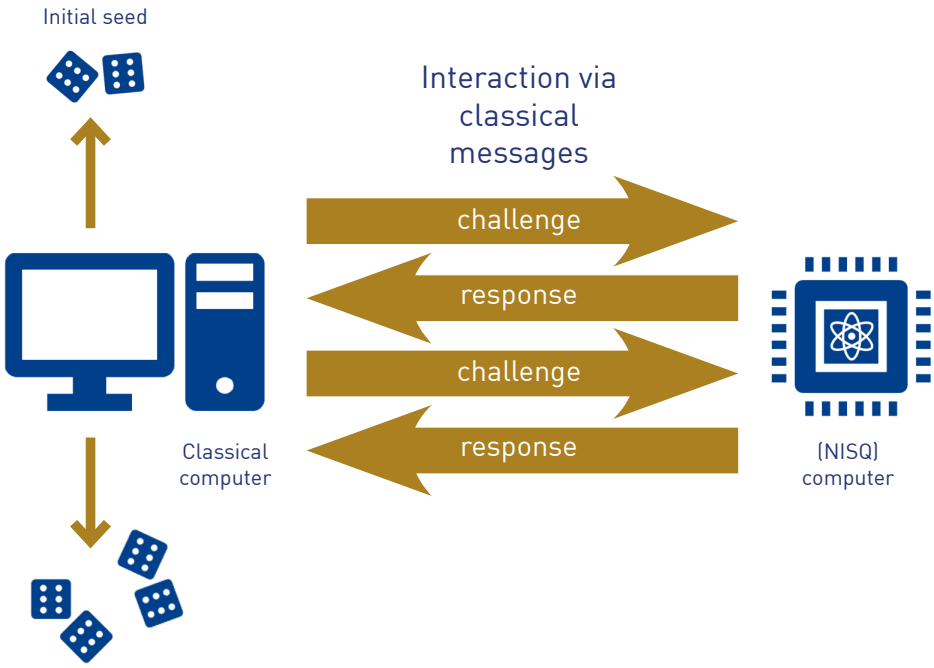

Figure 11. Scheme for exploiting noisy intermediate scale quantum (NISQ) computers or devices to produce certified randomness based on reasonable complexity assumptions. The NISQ computer does not need to be trusted. A random seed is used to issue challenges to the quantum computer, which has to send a response in a short amount of time. The responses are tested, and, if they pass the test, used to generate fresh, unbiased randomness.

The scheme is based on complexity assumptions that are deemed reasonable or likely to be true. A critical distinction from the use of complexity assumptions by CSPRNGs is that, for these QRNGs, the computational intractability assumption is only needed to validate the quantum/random behaviour of the apparatus in real-time. Once the random string is generated, a future break of the computational assumption does not compromise the secrecy of the random bits. Similarly, if the initial seed is later compromised, this does not compromise the secrecy of the generated random bits. Nonetheless, the assumption that the random numbers generated are private is only ensured if the classical computer and the quantum computer are both in possession of the agent(s), who are the only ones to know the random output.

One issue for this scheme is that the test validation of the response of the quantum computer by the classical computer requires a heavy brute-force classical computation, which depends on the number of qubits of the NISQ device. If the number of qubits of the quantum computer is too big, the test becomes infeasible.

Nonetheless, if the number of available qubits is high enough and of high-enough quality, it might become possible to use quantum computers able to run an arbitrary quantum algorithm. In such a case, protocols have been devised to ensure the production of a fresh randomness through challenges to the quantum computer that is untrusted or uncharacterized, and such protocols do not rely on any computational-hardness assumption.

3.2 Physical Realizations

In this section, we provide some illustrations of potential implementations of the ideas covered in the previous section, illustrating how QRNGs work in practice.

3.2.1 Device-dependent QRNGs

An example of entropy source for a QRNG that is device dependent is one where single photons are produced and directed towards a partially reflecting mirror. According to the rules of quantum mechanics, each photon has some chance of being reflected or going

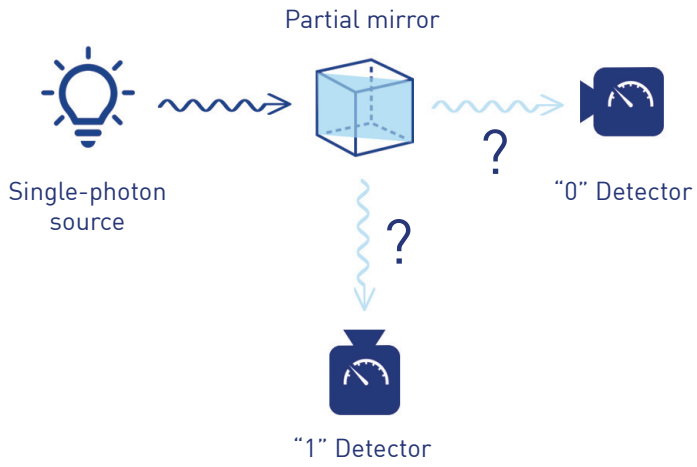

Figure 12. A standard set-up for a device-dependent QRNG: a single-photon source produces single photons that impinge on a partially reflecting mirror. Two detectors are positioned to detect which direction each single photon took. Depending on which detector fires, a 0 or a 1 is produced by the set-up.

through. The photon acts like an indivisible entity for the sake of detection: it is detected as either transmitted or reflected by one of two detectors, each associated with either "0" or "1" (see Figure 12). If the transmittivity of the mirror is chosen to be as close as possible to 50%, then, in principle, this set-up acts like an entropy source that is already very close to i.i.d. and unbiased. The main issues are related to imperfections in the components, starting from the source, which may not be exactly a single-photon source, or a non-perfectly balanced partial mirror. In particular, single-photon detectors typically have a *dead time* after a click, during which they are less sensitive to detecting a new photon. This leads to correlations in the generated sequence of bits, since it is more likely to generate a 1 after the generation of a 0; also, in general, it means that there is a limit to the generation rate, of the order of one Megabit (10^6 bits) per second (Mbps).

Many other methods exist to implement a quantum entropy source that is based on quantum light, with Ref. [11] providing a pretty exhaustive review that also includes information about the rates achievable by various schemes.

It is worth mentioning that set-ups like in Figure 12 are simple enough that they permit a continuous health check of the working status of the quantum entropy source. For example, even if there is some bias between 0 and 1, and a detector dead time, these are expected to stay consistent in time, leading to the production of *raw* bits (that is, the bits before randomness extraction) that follow some statistics that should also stay consistent in time and predictable. If the continuous health check based on verifying such statistics reveals some change, this may trigger a warning and even stop the production of output random bits altogether.

3.2.2 Device-independent QRNGs

3.2.2.1 Based on non-locality

As seen in Section 3.1.2.1, the conditions to certify non-locality QRNGs are demanding. A way to realize them is by means of a set-up where the relevant events are space-like separated so that, cording to the causality principle of special relativity, communication cannot take place. Figure 13 shows details of a set-up based on entangled photons that get measured at well-separated locations.

Detector efficiencies are important in this set-up because very high efficiencies are needed to close another potential loophole in a quantum non-locality experiment, referred to as a *detection loophole*.

Indeed, only recently, in 2015, the locality loophole and the detection loophole were closed in the same experiments [15–17]. Following such results, experiments have been performed to produce certified randomness using set-ups like that of Figure 13 [18,19]. NIST plans to integrate this kind of randomness generation in their randomness beacon service [20].

In the case of these device-independent QRNGs based on non-locality, the health check of the QRNGs is the verification of the continuous violation of a Bell inequality.

One can consider placing increased trust at least some of the devices in the set-up; for example, one could trust the source producing the entangled photons, but not put trust in the measurement devices. This *semi-device-independent* system may allow user to certify randomness in regimes of performance and statistics where a fully device-independent approach would not suffice to extract randomness.

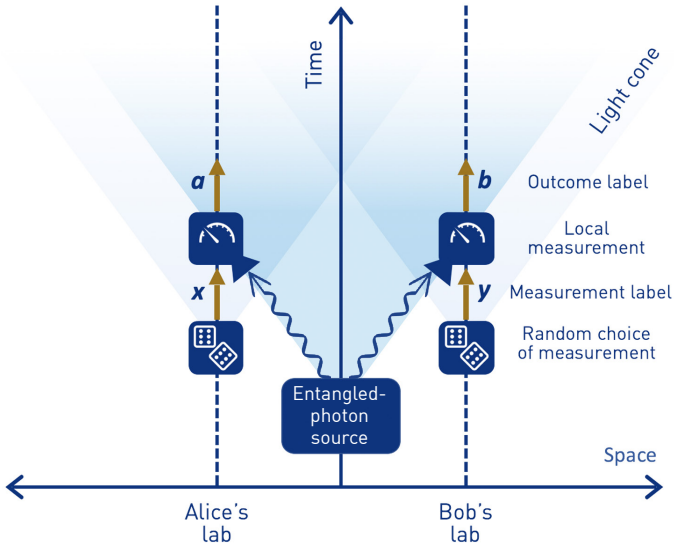

Figure 13. Experimental arrangement for the detection of quantum non-locality. Alice and Bob are separate experimenters who can perform localized operations. In this case, each perform measurements, e.g., polarization measurements, on photons emitted by an entangled-photon source. The blue and red shaded areas represent light cones in this spatio-temporal diagram, that is, the locations that can be reached by light at specific times, starting from the event at the vertex of the cone. The physical principle of no-faster-than-light signalling, at the basis of A. Einstein's relativity, implies that only events within the forward light cone of one event can be causally influenced by the latter. In this case, the measurements (say, polarization directions) x and y are randomly decided outside the light cone of the generation of the photons to be measured; similarly, Alice's choice of measurements as well as the measurements she performs are outside the light cone of the choice of measurement for Bob (and the other way around). This arrangement, together with the entanglement of the photons, can ensure that the outcomes a and b are random and unknown to anybody else. As long as the no-faster-than-light signalling is ensured by the physical set-up, and as long as the measurements are chosen randomly, this can be certified or validated by looking at the statistics of the outcomes, without relying on the quality or reliability of the source or of the measurements, hence the notion of "device independence".

In general, one may also be ready to trust that the devices involved do not display a behaviour that requires strictly enforcing the conditions of non-locality (see Ref. [13] for a discussion); for instance, there might be good reasons to believe that the measurement devices do not communicate at measurement time. Under such kind of assumptions, it is possible to generate randomness in a semi-device-dependent way in present NISQ devices by measuring the device qubits after they have been entangled.

3.2.2.2 Based on quantum computation

The tool to generate *fresh* (that is, previously unknown to anybody) random numbers making use of quantum computation according to the lines described in Section 3.1.2.2 is a quantum device that can outperform any classical computer at least in some specific task, which does not need to be of immediate real use other than for the purpose of generating randomness.

Such a feat was first reported by Google in 2019 in a landmark paper [21]. The task was that of sampling from a probability distribution obtained by running random transformations of the quantum state of all 53 qubits of their superconducting device, hence generating a substantial amount of entanglement, and measuring the qubits. Ref. [21] argued that such a task was practically impossible for any traditional computer.[6] Thanks to this fact, the task performed by the NISQ device can be used to rule out that somebody can run an appropriate classical computation in the short running time of the protocol of Figure 11.

3.3 Properties

The main features that set QRNGs apart from traditional TRNGs are the following:

- They rely on quantum features to generate new, physically fundamental randomness. This is in stark contrast to traditional RNGS, which are based on unknown, but, in principle, knowable information implicitly pre-existent in a physical device, a kind of information which could even be potentially implanted.

Property	Traditional/Classical	Quantum
Entropy source	Randomness based on complexity of process and partial ignorance.	Fundamental randomness.
Ease of certification	Limited ability to certify the underlying physical process, which is inherently a complex one. Certification of the quality of the output based on standard tests.	Can validate the underlying physical processes. Certification of the quality of the output based on standard tests.
Resistance to tampering	Some ability to run health check on entropy source.	Built-in check based on simplicity of process and more sensitive to tampering. Device-independent versions offer highest resistance against tampering of entropy source itself, even by the providers themselves.
Quality of entropy	Various degrees. The underlying process used as entropy source may work in a physical regime where there are large bias and relatively high correlations (that is, small entropy).	High entropy from the start based on the simple design of the source; a QRNG entropy source can be argued to be very close to i.i.d. from the start.
Speed	Can be very high, and several sources may be combined to obtain higher rates.	High, also because of the quality of the initial entropy, but device-independent implementations may be slow, for example.
Size	Can be very small and embedded on chip, e.g., exploiting a randomness source like thermal noise.	Varies substantially, going from embeddable in smartphones to room-size dimensions for implementing device-independent randomness generation based on non-locality.

Table 1. Comparison of properties of traditional physical true random-number generators and quantum ones.

[6] The claim has been subject to some controversy, but it appears there is general agreement that the targeted task poses an incredible challenge for classical computers. Also, the challenge can be made harder relatively easily by quantum processors with only a few more qubits, due to the exponential growth of the difficulty of the task with the number of qubits [5].

- Certification and validation are helped by the relatively simple physics and simple randomness-generation processes that are at the basis of QRNGs.
- The entropy source can be high-quality, meaning that random bits generated by the exploited quantum physical process are already relatively close to i.i.d.

The entropy source can be *argued* to be truly random in addition to running tests on the output of the QRNG to verify that it passes statistical tests. This argument still relies on an underlying physical model, but the processes at play are simpler compared to classical physical random generators that are based on complicated processes believed to be too difficult to model and predict. In this sense, QRNGs have a stronger underpinning as the procedure of building a model and refining it allows one to have more confidence in the model. The simple design of QRNGs may also lead to increased resistance to tampering and higher quality of initial entropy before extraction.

Advantages of using devices with these features are the following:

- They reduce or remove the risk that the random data that is being provided by the RNG is known, or partly known, in advance to some adversary, both because of the fundamental randomness of quantum processes and because the relatively simple processes at play, despite being quantum, are simple and resistant to external interference.
- They may permit an almost real-time validation that the randomness generation process is functioning properly. Such validation is particularly compelling in the case of device-independent QRNGs.
- The high-quality of the entropy source, already close to i.i.d., reduces any issues that may arise from post-processing or entropy extraction and reduces the chance that subtle correlations, typically present in TRNGs that do not use quantum mechanics, are exploited by an adversary.
- When used in parallel with other RNGs and combined/composed in a secure manner, they offer a qualitatively different source of randomness that mitigates the risk that the other RNG(s) being used might have been compromised.
- At a higher level, including a more secure RNG in platforms and tools may generate greater trust in those tools and platforms among end users.

4. Use Cases of Quantum Random-Number Generators

QRNGs could be the ideal solution in many scenarios or circumstances because of their fundamental properties. This is particularly true in those cases where any potential practical downsides related to speed, size, or cost can be overlooked because of the substantial gains.

What kinds of **threat scenarios and use cases** might already seriously benefit from the properties of QRNGs?

The opinions we gathered from the experts indicate the following ones as best fit for QRNGs:

- Those protecting *high-value assets and critical systems:* this may include confidential information (including financial and medical) to be shared in encrypted form or authentication for access to strategic applications and databases, at the military, governmental, or enterprise level, particularly in the financial sector. The private randomness provided by device-independent QRNGs may assume particular importance for such assets and systems.

- Those where the ability to ensure the random nature of the numbers generated is paramount to *guarantee fairness* and to *establish or maintain trust:* this may include lotteries, sport events and competitions, or the unbiased allocation of scarce resources.

Here are some opinions from the experts who were interviewed.

In general,

> *[i]f someone absolutely needs to have random data, then they would be far more likely to tolerate the downsides of using QRNG technology. [...] Essentially, any application where the ability to guarantee randomness via application of the Bell test is deemed worth the added expense/constraints associated with utilizing QRNG technology. – Bruno Couillard*

About device-independent realizations:

> *QRNGs which are device independent are also interesting in scenarios where the involved parties do not trust each other. Having a way to make each side satisfied that the numbers are truly random is their main advantage. – Anonymous expert*

The use of QRNGs can be a welcome solution even beyond the threat scenario and use cases listed above. For, example, they may be a welcome solution:

> *[f]or small mobile devices where it may be difficult to gather sufficient entropy to properly seed traditional PRNGs. – John Mulholland*

The unique properties of QRNGs may not always be needed:

> *For applications where secrecy is not an issue, such as in running simulations, the premium paid for QRNGs might not be a justifiable value[, but] QRNGs may also be used to seed pseudo-RNGs in [such] less sensitive applications. – Anonymous expert*

One respondent wrote:

> *I think that QRNGs will only constitute an alternative to [other] existing physical RNGs if they can be implemented in a secure, reliable, simple and cost-efficient manner in practice, and can provide sufficient performance. – Anonymous expert*

Another respondent thinks QRNGs are already the best solution whenever a true RNG is required:

> *QRNGs can now be used for all applications which require true random numbers.*
> *– Bruno Huttner*

Table 2 offers an overview of some use cases with a brief description of the relevant rationale or scenario, while in the following subsections we elaborate more on some specific ones.

4.1 Cloud-Based IT Security Infrastructure

Free sources of entropy that are made available by an Operating System or a processor should not be relied upon by cloud-bound software applications that perform security functions. There are many cloud platform choices, and there is no guarantee that an entropy source that works on a software developer's machine will be the same entropy source that is available while running in the cloud platform. Software and operating systems are virtualized in cloud environments, and the underlying RNG could be simulated with something less secure, unbeknownst to the secure software application.

Use case	Rationale/Scenario
Lottery / Casinos	High-value random values. The assurance that the randomness is genuine contributes to the trust the players put into the games. The business can use the easy verification to facilitate the smooth running of the activity.
IT equipment and infrastructure	In terms of security, integrated QRNG could help to overcome faulty private key generation in smartcards or cryptographic modules, which need to generate the keys internally. Secure device keying during manufacturing (IoT, automotive, consumer electronics), especially for products expected to be in use for a long time.
Cryptography	High-value keying material for cryptographic functions (e.g., military applications, certifying authority key generation using HSMs). Keying for health-related data to be stored for decades.
Quantum Key Distribution (QKD) devices	The security claim of QKD devices requires information-theoretic random numbers. There is no way to validate this assumption without physical, certifiable randomness.
Experiments	Fundamental experiments based on the assumption of a perfect random choice.
Sports/competitions	Fairness in assigning ordering, position, sides, or in breaking a tie.
Public services	Assignment of prosecutors, juries, judges. Random checks and auditing.
Allocation of scarce resources	Fairness in the distribution of a scarce resource at any level.

Table 2. Some use-case scenarios for quantum random-number generators.

To overcome this issue, special cloud-provider APIs that provide random numbers, or a third-party entropy-as-a-service, should be used for random numbers in security operations.

When choosing a cloud-based RNG, QRNG vendors have an opportunity to position their products as the superior security choice and could justify a premium price for the service.

4.2 Classified Government Networks and Communications Systems

High-value assets and critical systems used in classified government networks or military networks would benefit from the verifiable randomness attributes of QRNGs since the devices can be more easily examined to ensure that tampering has not occurred. The private randomness provided by device-independent QRNGs may be particularly relevant for such assets and systems.

Any downsides in terms of speed or size (e.g., to implement a non-locality-based QRNG without loopholes) may well be acceptable given the high stakes.

4.3 Secure Device Keying During Manufacturing (IoT, Automotive, Consumer Electronics)

There are many products that need to be delivered to customers in a secure state. The costs of setting up customer support capabilities to help customers generate security keys can drive up prices and shrink profit margins. Because of this, many manufactures pre-program or pre-generate security keys on a manufacturing line and program into their products

prior to shipping their products to retailers or customers. Electronics manufacturing is often done by contractor manufacturers, and one of the methods involves using random-number generators on the manufacturing line. This means using a manufacturing system that contains an RNG that interacts with the manufacturing line. Given that some products, like in the automotive industry, need to be secured for very long periods of time given their life cycle, QRNGs could be useful in these systems to ensure true verifiable randomness for long-lived keys. This would provide further assurance also, for example, in case the security of the manufacturing process is ever audited by a customer or other stakeholder for security purposes.

4.4 Quantum Cryptography

Quantum cryptography protocols—and, in particular, Quantum Key Distribution (QKD) ones—rely on quantum physics to enable the information-theoretically secure establishment of keys that can be used for symmetric key encryption. This contrasts with traditional network security schemes that often rely on the hardness of special mathematical problems (equivalent to *trapdoor predicates*) to establish encryption keys.[7] Sectors that are most active in exploring quantum cryptography include the government, finance, and telecommunications sectors. QKD schemes and devices rely on randomness and, as one of the experts points out, may be one of the best matches for QRNGs:

> *The security claim of QKD devices requires information-theoretic random numbers (True Random-Number Generators). Typical security proofs simply assume that perfect randomness is available! There is no way to justify this assumption without physical randomness, and, at the paranoid level of QKD, the physical random-number generator should be as simple as possible (from the perspective of certifiability). – Norbert Lutkenhaus*

4.5 Financial and Healthcare Services

Financial and healthcare services data is long-lived and needs to remain confidential. Financial and healthcare details about any given person or corporation can be relevant for decades or even, particularly in the case of healthcare, for a lifetime. Given the uncertainty that is cultivated by the threat posed by quantum computers (see Section 2.3.1.1), banks, financial institutions, and healthcare providers that are worried about long-term confidentiality that stretches decades—and are currently considering upgrading cybersecurity systems to quantum-safe—may be open to looking at improving their sources of randomness at this time as well. This affords QRNG vendors an opportunity to position QRNGs as an appropriate pairing to other quantum-safe controls like post-quantum cryptography and quantum key distribution that may already be under consideration or on an upgrade path within the prospective customer.

4.6 Random Selection in Gaming, Sports, and Other Activities

Fairness is fundamental in several activities, including gaming, sports, and lotteries, and QRNGs offer an unbiased source of random numbers not derived from deterministic or computational means. QRNGs may be valuable where there are high stakes, or low levels of trust, or great value in establishing or maintaining trust. One expert writes:

> *[QRNGs] which are slower but provide a high perception of fairness [..] could be used in public processes (maybe sports team pairings, state lotteries) where the amount of random numbers required is relatively small. – Juan Carlos García Escartín*

[7] If one-time-pad encryption is used (which requires a key that is as long as the message and that is only used once), then the encryption is also information-theoretically secure. Even if not using the one-time-pad encryption algorithm, note that symmetric key encryption algorithms require less mathematical structure than a (classical) asymmetric key agreement algorithm and thus are generally considered much harder to cryptanalyze.

5. Outlook

How should one expect the market for Quantum Random Number Generators (QRNGs) to evolve in the coming years?

Besides input on practical use cases, the experts we interviewed shared some insight on factors that may speed up (accelerators) or slow down (inhibitors) the adoption of QRNGs. The information and opinions we collected also lead us to indicate some recommendations for both QRNG vendors and RNG end users.

5.1 Accelerators and Inhibitors

In the following subsections, we delineate in some detail some accelerators and inhibitors for the increase of market for QRNG, summarized in Table 3. These issues are separated from issues of performance of QRNGs.

Accelerators
Readily attainable service offering for quantum cloud vendors
Purposeful use of NISQ quantum devices
RNG flexibility in mainstream operating systems
Entropy needed in cloud for post-quantum cryptography
Race to the bottom for chip-based QRNGs
Sovereign fabrication for chip-based QRNGs
QKD adoption driving QRNG adoption
Innovation-focused strategic procurement
Inhibitors
FIPS140 security evaluations
Competing with *free* RNGs

Table 3. Some accelerators and inhibitors for the expansion of the QRNG market.

5.1.1 Accelerators

5.1.1.1 Immediate gateway service for quantum cloud computing vendors (accelerator)

There are several quantum computing cloud vendors announcing plans or entering the market. Existing non-quantum cloud computing vendors are also likely to explore market opportunities as Noisy Intermediate-Scale Quantum (NISQ) devices become more powerful, with the number and quality of qubits increasing over time.

The number of qubits available today is relatively small, with computing power still limited, and a *killer app* for cloud-based quantum computing has not yet emerged. Quantum cloud vendors who are trying to stake an early claim in the market need to develop business plans to justify investments. Management teams are looking for *signs of life* that a market exists and is viable, as they push for their respective quantum computing business units to demonstrate some sort of customer uptake metric to help validate that there is a market of users who are willing to pay for a quantum computing service.

QRNG-as-a-service is a potentially attractive component for a suite of quantum cloud service offerings because it can be made available immediately and has a potential market with users who are deploying quantum-safe cybersecurity systems.

If major quantum cloud vendors offer QRNG services as part of their suite of quantum computing offerings, it has the potential to make QRNG a defining feature of quantum clouds and promote the benefits of quantum randomness to a mass market of cloud computing end users.

Cloud-based QRNG services also help to alleviate QRNG technology's current cost and performance inhibitors. Until QRNGs reach a level of maturity and commoditization comparable to that of PRNGs and their traditional TRNG counterparts, costs will likely continue to be an inhibitor to QRNGs' widespread use. NISQ-based QRNGs that are available via cloud access can act as an accelerator if the cost is lower than that of other dedicated QRNG solutions in the market, and performance can be increased and scaled up as needed using the traditional cloud computing business model.

5.1.1.2 Purposeful use of Noisy Intermediate-Scale Quantum (NISQ) devices

The first demonstration of quantum supremacy (see Section 3.2.2.2) was attained by considering a task without immediate utility and rather tailored to emphasize the advantage of a quantum device over classical computers. However, opportune modifications of such a task combined with a proper protocol can be used to generate certifiably *fresh* random numbers (see Section 3.1.2.2). This could be the first use of NISQ devices for a practical application with commercial value.

As they are developed, new quantum protocols offer an opportunity for more efficient, more secure, and lower cost-per-bit access to NISQ-QRNG-based randomness. Over time, these improvements will allow NISQ-QRNG vendors to differentiate themselves through improved throughput rates and lower prices and costs over time—directly addressing the price vs. performance aspects that are inhibiting QRNG technologies today—and help to maintain a competitive and thriving quantum-randomness-as-a-service market.

5.1.1.3 Niche customer-driven RNG flexibility in mainstream products (accelerator)

High-security customers such as governments and banks use mainstream technology products like PCs and Android phones and push for features that promote RNG flexibility in these mainstream products. This opens the door for QRNG vendors and allows them an opportunity to integrate with popular and widespread computing platforms.

For reasons related to cost efficiency and ease of maintenance, it is common for high security–conscious customers to try to stay on the latest commercially available branches of a product and not require technology vendors to make special custom versions that are difficult and expensive to maintain over time. While these niche customers try to avoid customizations, they are still able to influence security features and product roadmaps of technology vendors through procurement programs and buying power.

Flexibility of RNG sources is common on most major operating system platforms. The HSM market has been reliant on such platform access for the past 20 years, and major operating system vendors have made APIs available to allow trusted 3rd party security vendors access to their platforms. RNG flexibility features fall into this category as high-security customers tend to pay close attention to sources of random numbers and often look to augment random-number generation and entropy collection to a level that suits their own security policies and requirements.

Major platforms that offer RNG flexibility afford QRNG vendors a chance to establish a technical foothold within the high-security niche customer market and later work their way towards the mainstream commodity RNG market over time.

5.1.1.4 Performance metrics after post-quantum crypto migration (accelerator)

It is possible for an entropy source to be exhausted and require time to build up entropy again. On single-user systems, entropy exhaustion may not be a limiting factor; however, in a cloud context, an entropy source might be shared by many different applications and virtual machines. Entropy exhaustion could be an important factor to consider for security-sensitive cloud-based applications.

Currently, most information security systems are using quantum-vulnerable cryptography like RSA and ECC, which have relatively small keys (hundreds of bytes). In 2023, NIST will be publishing a new set of public key cryptography standards that are quantum-safe; these standards do not include RSA and ECC and instead rely on cryptographic algorithms that use much larger key sizes. This has the potential to drive up demand for entropy on systems and platforms, and entropy exhaustion could be a cause for concern.

5.1.1.5 Chip-based QRNG: race to the bottom (accelerator)

Chip-based QRNG offers an opportunity for hardware-based QRNG vendors to produce QRNG devices at volumes that can drive down per-unit costs and bring QRNG component prices down to levels that are more attractive to electronics designers. This will allow QRNGs to proliferate but will also lead to commoditization and thin margins. QRNG chip vendors need to walk the balanced line faced by other chip companies of offering multiple product lines with low-cost commoditized chips (older designs) and higher-value products that offer premium features. In the case of QRNG, these premium features could revolve around verifiability, supply-chain pedigree, or in-country sovereign manufacturing of the QRNG chip.

5.1.1.6 Chip-based QRNG: sovereign fabrication (accelerator)

While many countries design semiconductor devices, much of the world's chip manufacturing occurs in Asia. The largest chip manufacturer is TSMC (Taiwan) followed by Samsung (South Korea). Upgrading a chip fab to support new process nodes can cost more than $1B, so most chip design companies in the world now follow a *fabless* business model, meaning that they contract out chip manufacturing to companies like TSMC. While a large amount of chip manufacturing has migrated to Asia, many western countries still have chip fabs that use older manufacturing technologies and are capable of manufacturing QRNG devices.

An in-country fab can produce these chips, allowing the QRNG vendor to claim a *sovereign* supply-chain pedigree, which fits well with the *easy to verify* positioning that most QRNG vendors claim.

This sovereign supply-chain pedigree could be marketable to high-security customers who also care about verifiability, since many of the security concerns related to fabless outsourcing of chip manufacturing to foreign countries revolves around verifiability, and specifically the question of whether or not the design that was sent for manufacturing contains only the chip circuitry that was designed in, without extra, undocumented circuitry that might pose a security threat or present a hidden back-door.

5.1.1.7 QKD adoption driving QRNG adoption (accelerator)

Quantum Key Distribution (QKD) is a network security technology that positions itself as an alternative to computationally secure key agreement. The main selling feature is that customers who care about security and defense-in-depth strategies can add an extra layer of security protections to their network that guard against computational attacks, including quantum attacks.

QRNGs are used in QKD products and are an elegant addition to QKD product selling points.

Since PRNGs derive their security from mathematical algorithms, QKD technologies need QRNGs to decouple from mathematics-based security features—like PRNGs that could be vulnerable to computational attacks—in order maintain their quantum pedigree.

The QKD market is approaching a tipping point. There are many countries funding QKD national demonstrators, including the UK, the EU, Korea, Japan, China, the US, and Canada. While these national demonstrators are funded by the governments of their respective countries, sectors like banking and telecommunications are also investigating QKD on their own and as part of the demonstrators. As QKD networks proliferate and grow, there is an opportunity for QRNG vendors to participate if they can position the QRNG as a vital technology to use in quantum networks.

5.1.1.8 Innovation-focused strategic procurement (accelerator)

As quantum technologies continue to attract attention, large companies may choose to explore and experiment with quantum-based technologies, and QRNGs could fit well with this desire. Large companies have procurement business units with a primary function of dealing with suppliers. These procurement organizations are well equipped to deal with large vendors and ensure that the company is deriving good value from their vendors and partners. Not all vendors can be viewed as equal: some large vendors are considered strategic partners, where the vendor and the company are tightly vested in each other's continuing success. Small vendors can be strategic as well, especially when the vendor offers an innovative technology that fills a customer need or gives the company some competitive advantage. In these cases, the large company understands that the small vendor may not be able to scale to meet the company's needs and will instead try to partner the smaller vendor with one of their larger strategic partners. These strategic procurement programs could offer QRNG vendors opportunities to grow at an accelerated pace.

5.1.2 Inhibitors

While we consider only two inhibitors, they may be considered of greater importance than several of the accelerators.

5.1.2.1 Current cost & competing with "free" (inhibitor)

Currently, the cost of QRNG technologies is relatively high compared to alternatives such as traditional TRNGs and PRNGs, where implementations of these technologies are often included for free in processors or as part of operating systems. There is a barrier to establishing a market for QRNG because of the plethora of low-cost alternative offerings for obtaining random numbers to varying degrees of security.

QRNGs present themselves as the higher security choice when compared to PRNGs; however, in the eyes of product designers trying to meet a particular standard for randomness, if using a PRNG will allow them to access their respective market, then they have reached the minimum bar required and they will not spend extra to meet the higher security level.

5.1.2.2 North American FIPS140 security evaluations (inhibitor)

The FIPS140 security evaluation standard is used to certify cryptographic and security products for sale to the US and Canadian governments. It is also unofficially relied upon by the financial industry in North America when making purchase decisions for Hardware Security Modules (HSMs).

FIPS140 sets a minimum-security level for cryptographic implementations and focuses on testing products against a standard set of inputs and evaluating the product's outputs and behaviour.

Vendors tend to focus only on the minimum defined set of security features to obtain the certification level they are pursuing because adding extra features beyond that carries a downside risk that the product could fail to pass evaluation. Features that other markets may consider a security benefit, like upgrading firmware, may invalidate a FIPS140 evaluation rating.

Unless QRNGs are specifically added to and required by the FIPS140 standard, vendors will tend to choose the least costly path to certification. This may inhibit QRNG adoption for government sectors in the North American market.

5.2 Recommendations for QRNG Vendors

QRNG vendors face head winds in the market due to the widespread availability of other PRNG and TRNG solutions in Operating Systems and as part of mobile and desktop processors. Competing on price will be difficult because other products are perceived as free, so QRNG products must differentiate on increased security value. QRNG vendors are already positioning their products as *verifiably random*, but this approach could be taken a step further with additional positioning:

a. Emphasis on the geographic design and manufacture pedigree of a QRNG – "Designed in Country X, manufactured in your local country, easy to verify."
b. Claim quantum-safe randomness market category – "Companies upgrading their crypto to quantum-safe should also upgrade to QRNGs to avoid weak links in the security chain."
c. Position as verifiable in the cloud for peace of mind – "Virtualization and moving to the cloud wrecks entropy. Do you know where your random numbers come from? QRNGs can provide your cloud applications with verifiable random numbers."

The above market positions are suitable for hardware-based and cloud-based QRNG vendors. Cloud QRNG vendors that rely purely on quantum-software implementations (i.e., quantum algorithms running on a quantum computer to produce entropy) will avoid the supply chain costs of QRNG hardware devices. However, quantum software–based QRNGs have customer education hurdles to overcome because customers are used to traditional computing paradigms where software is always visible and under control of whomever has the computer. Quantum software–based QRNGs would greatly benefit from published academic scrutiny and endorsement from recognized security professionals to help with this education hurdle. However, the business implications of a software-only QRNG model are attractive for investors and customers because cloud-based QRNGs give a ready use case to quantum computing clouds and allow customers to access the technology at pricing that incrementally scales.

5.3 Recommendations for RNG End Users

Carefully consider where your random numbers are coming from. Random numbers are critically important to cybersecurity systems; simply put, if someone can discover the random numbers, then they will be able to compute the security keys protecting your data. Random numbers are most likely coming from your operating system, unless you are specifically asking for them from something else. While the O/S may be good at collecting randomness from various hardware components and system events around it, all of the places from which the O/S would source randomness are virtualized when you move to the cloud, and security assumptions made by the O/S designers may no longer hold true.

Consider RNG exhaustion and substitution on cloud platforms. Some software applications make direct calls to processors to obtain random numbers; for example, Intel and AMD processors that support the RDSEED and RDRAND processor instructions allow

applications to ask the processor directly for random numbers. In a virtualized or cloud environment, these processor instructions are serviced by a virtual machine, not by the Intel or AMD processor. While the RDRAND processor instruction is intended to produce a large amount of quality random data for an application, once moved to the cloud, a call to RDRAND is serviced by *something else* in a virtualized computing environment that could offer less security than needed, or might be shared with many other guest operating systems and applications. Some implementations of TRNGs can temporarily run out of entropy and, depending on the implementation, may tell the requesting application using an error message. If the error is ignored by the application, then the application may mistakenly use non-random numbers for critical operations.

Choose your RNG in a virtualized compute or cloud environment. The best way to avoid issues with inferior random numbers in cloud applications is to explicitly choose your RNG implementation when deploying applications to the cloud. Some cloud vendors offer cryptographically secure RNG implementations that can be called with special cloud APIs, but also consider using QRNG services from 3rd party vendors after you have verified the security of the vendor's QRNG implementation.

Consider upgrading to a QRNG when upgrading to Post-Quantum Cryptography. For better quantum-computing resistance, QRNGs offer higher-quality random numbers than PRNGs; they also offer improved ease of verification or certification and of real-time health checks with respect to traditional TRNGs. Given the effort to be spent in moving to a quantum-safe solution, it is reasonable to strengthen the source of randomness used at the same time, choosing one of the highest possible quality and reliability.

6. Conclusions

Quantum Random-Number Generators (QRNGs) are a relatively new emerging tool that should be given proper consideration for the added value and unique properties they may offer, particularly for products and services that are required to protect high-value assets for a long time.

Key features that characterize QRNGs are:

- Easier certifiability and continuous health checks than standard True Random-Number Generators (TRNGs)
- (For some QRNGs) the unique property of being able to certify that the random numbers are *fresh* and *private*

Such features make the consideration of QRNGs particularly compelling for those applications where:

- there is the necessity of protecting high-value assets and critical systems, and
- the ability to ensure the random nature of the numbers generated is paramount to guaranteeing fairness and establishing or maintaining trust.

While these are the most serious and compelling cases, through a continued increase in performance and a continuous decrease in size and cost, QRNGs could find more general adoption even at the level of smartphones, for example, which would both add a layer of security and contribute to an increased level of trust.

The interest of end users who transparently use randomness (e.g., consumers using cell phones or online banking) is inherently limited in the details and quality of the randomness

used in securing data and communication, and will likely remain so. On the other hand, the interest of vendors who may use QRNGs in the products and services they sell, although also still limited, is growing. Such growth has been more rapid in the telecommunications, defence, and financial sectors, where security concerns have driven the adoption of QRNGs. Nonetheless, many vendors focus on meeting institutional security criteria at a low price, for which standard RNG (either pseudo- or true-) often suffice, rather than trying to provide the highest-quality randomness.

There is the expectation that the QRNG take-up will increase considerably in the next 10 years. On top of the conscious adoption in areas where security is of the highest concern, this process may be sped up by the general appeal—also for the general public—of the fast-developing quantum technologies. It is worth noting that institutions and companies that are already considering moving to quantum-safe cryptography may use the transition opportunity to adopt high-quality randomness generators to further strengthen their systems.

References

[1]. A Million Random Digits with 100,000 Normal Deviates, (2001).
[2]. J. Kelsey, L. T. A. N. Brandão, R. Peralta, and H. Booth, A Reference for Randomness Beacons: Format and Protocol Version 2, No. NIST Internal or Interagency Report (NISTIR) 8213 (Draft), National Institute of Standards and Technology, 2019.
[3]. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition (Cambridge University Press, Cambridge, 2010).
[4]. M. Mosca and M. Piani, Quantum Threat Timeline, https://globalriskinstitute.org/publications/quantum-threat-timeline/.
[5]. Shtetl-Optimized » Blog Archive » Scott's Supreme Quantum Supremacy FAQ!, https://www.scottaaronson.com/blog/?p=4317.
[6]. E. B. Barker and J. M. Kelsey, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, No. NIST SP 800-90Ar1, National Institute of Standards and Technology, 2015.
[7]. M. Wertheimer, The Mathematics Community and the NSA: Encryption and the NSA Role in International Standards, Not. Am. Math. Soc. 62, 165 (2015).
[8]. M. Quinn, The Cypherpunks Who Cracked Netscape, San Franc. Chron. (1995).
[9]. L. Dorrendorf, Z. Gutterman, and B. Pinkas, Cryptanalysis of the Random Number Generator of the Windows Operating System, ACM Trans. Inf. Syst. Secur. 13, 10:1 (2009).
[10]. J. Kelsey, B. Schneier, D. Wagner, and C. Hall, Cryptanalytic Attacks on Pseudorandom Number Generators, in Fast Software Encryption, edited by S. Vaudenay, Vol. 1372 (Springer Berlin Heidelberg, Berlin, Heidelberg, 1998), pp. 168–188.
[11]. M. Herrero-Collantes and J. C. Garcia-Escartin, Quantum Random Number Generators, Rev. Mod. Phys. 89, 015004 (2017).
[12]. L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, S. D. Leigh, M. Levenson, M. Vangel, N. A. Heckert, and D. L. Banks, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, (2010).
[13]. A. Acín and L. Masanes, Certified Randomness in Quantum Physics, Nature 540, 7632 (2016).
[14]. P. R. Smith, D. G. Marangon, M. Lucamarini, Z. L. Yuan, and A. J. Shields, Simple Source Device-Independent Continuous-Variable Quantum Random Number Generator, Phys. Rev. A 99, 062326 (2019).
[15]. B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg,

R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson, Loophole-Free Bell Inequality Violation Using Electron Spins Separated by 1.3 Kilometres, Nature 526, 7575 (2015).

[16]. L. K. Shalm et al. Strong Loophole-Free Test of Local Realism, Phys. Rev. Lett. 115, 250402 (2015).

[17]. M. Giustina et al. Significant-Loophole-Free Test of Bell's Theorem with Entangled Photons, Phys. Rev. Lett. 115, 250401 (2015).

[18]. P. Bierhorst, E. Knill, S. Glancy, Y. Zhang, A. Mink, S. Jordan, A. Rommal, Y.-K. Liu, B. Christensen, S. W. Nam, M. J. Stevens, and L. K. Shalm, Experimentally Generated Randomness Certified by the Impossibility of Superluminal Signals, Nature 556, 7700 (2018).

[19]. Y. Liu, Q. Zhao, M.-H. Li, J.-Y. Guan, Y. Zhang, B. Bai, W. Zhang, W.-Z. Liu, C. Wu, X. Yuan, H. Li, W. J. Munro, Z. Wang, L. You, J. Zhang, X. Ma, J. Fan, Q. Zhang, and J.-W. Pan, Device-Independent Quantum Random-Number Generation, Nature 562, 7728 (2018).

[20]. M. Fischer, M. Iorga, and R. Peralta, A Public Randomness Service, in SECRYPT 2011: Proceedings of the International Conference on Security and Cryptography (2011), pp. 434–438.

[21]. F. Arute et al. Quantum Supremacy Using a Programmable Superconducting Processor, Nature 574, 7779 (2019).

7. Appendix

7.1 Quantum Properties

7.1.1 Basic quantum properties

7.1.1.1 Superposition

In classical mechanics, a physical object can have only one specific value for one of its physical properties.[8] An example is the position of a particle (which is *here* or *there*, for example) or the direction of rotation of a spinning top (*clockwise* or *counter-clockwise*). In information-theoretical terms, the corresponding fact for classical information is that a bit can only assume the value "0" or the value "1".

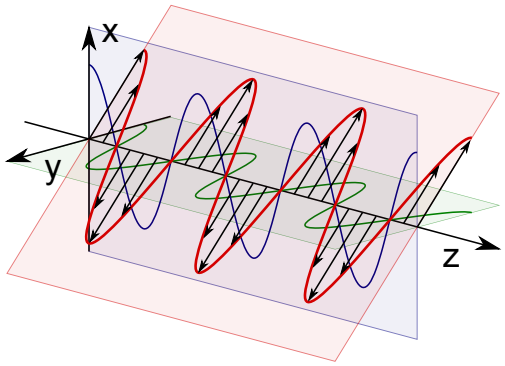

Figure 14. Polarization of light. Light is an electromagnetic wave. What oscillates are the electric and magnetic fields. The polarization of light is related to the plane of oscillation of such fields. The red plane in the figure indicates a generic plane of linear oscillation for, say, the electric field. An oscillation in such a plane can always be thought as the result of the combination of an oscillation in the horizontal plane (green) and in the vertical plane (blue). The notion of polarization applies also to the quanta of light, the photons.

In quantum mechanics, this is not true anymore. The state of a physical system is described by a *vector*, and any superposition (technically, linear combination) of two states or vectors describes a perfectly valid physical state because of the so-called linearity of the theory.

Here is an explicit example using the so called *Dirac* or *ket* notation to denote states and vectors

A quantum bit (better known as *qubit*) can exist in the quantum states $|0\rangle$ and $|1\rangle$ that correspond to the two possible *classical* values of a bit, 0 and 1. A possible realization of this kind of system is the polarization (see Figure 14)—e.g., horizontal (H) or vertical (V)—of a particle of light (also known as *photon*), for which we can imagine the assignment or relabelling H⇔0 and V⇔1.

For such a system, the superposition state given by the sum $(|0\rangle + |1\rangle)/\sqrt{2}$ is also a valid state[9]—in the case of a photon, simply associated with some other polarization.

The quantum states $|0\rangle$ and $|1\rangle$ may nonetheless be singled out as special, for example,

[8] In this case, we do not consider *ignorance*; we discuss the best possible conceivable knowledge about the system within classical mechanics or quantum mechanics.
[9] One has to be careful in defining *properly normalized* linear combinations or defining the notion of quantum state in a more sophisticated way, hence the $\sqrt{2}$, but here we will stick to the simplest possible and concise way of indicating such a superposition.

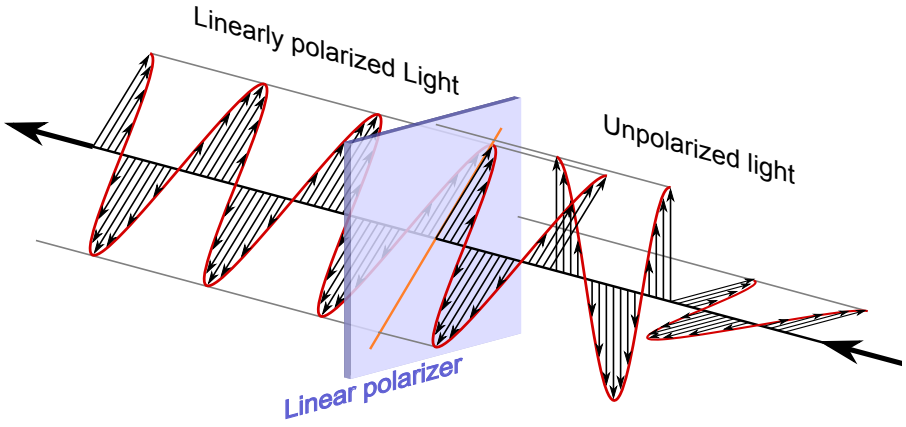

Figure 15. Photons that hit a linear polarizer may either go through or be stopped. The probability of one or the other depends on the relation between the incident polarization and the polarization selected by the linear polarizer. If a photon goes through, its polarization will be the one dictated by the polarizer. This is actually a way of preparing specifically polarized light or photons starting from unpolarized light.

because they may be the only ones that do not naturally evolve in time or because of how a quantum measurement takes place.

7.1.1.2 Genuine randomness from measurements

According to the postulates of quantum mechanics, a measurement can only yield as outcome one among a set of discrete values[10] that depend on the physical observable that is being measured.

As an example: It is possible to measure the polarization of photons along different directions. This roughly corresponds to rotating the polarizing filter that one uses on camera lenses. Such a filter, in its initial position, may let only horizontally polarized photons, that is, photons in the state $|H\rangle$, pass and block vertically polarized ones, that is, photons in the state $|V\rangle$. Rotating the filter allows one to choose the polarization direction that is let through while completely blocking the photons polarized in the orthogonal direction.

What happens if a photon prepared in the state $(|H\rangle + |V\rangle) /\sqrt{2}$ hits the polarizer when it is in its initial setting? Quantum mechanics tells us that the photon will either go through or not, with 50% chance. This depends on the coefficients in the linear combination of the H and V states: in this case they are equal. Moreover, there is no way, for anybody, of knowing in advance which result will be obtained. *The randomness in the measurement process is genuine, not due to ignorance.* This is different from tossing a coin in the realm of classical physics; in the latter case, in principle, someone with enough information about the toss dynamics and about the environment surrounding the coin, and with enough computing power, could be able to perfectly predict the outcome. Rotating the polarizer or preparing the photon in a different state are in practice equivalent—one can change one or the other to obtain the same change in the probability distribution. For example, we could rotate the polarizer so that a photon in the state $(|H\rangle + |V\rangle) /\sqrt{2}$ certainly passes.

The point is that it is in principle possible to control the level of randomness produced in the process of letting a single photon go through a linear polarizer. In an ideal situation, the outcome is always dichotomic and can generate one bit of randomness per photon, based on whether the photon passed or not.

[10] This is not the entire story, but we can limit ourselves to this for the sake of this discussion.

7.1.2 Quantum properties of distributed systems

Additional quantum properties become relevant when one deals with more than one system.

7.1.2.1 Entanglement

Entanglement is the quantum property where two or more quantum systems are so tightly intertwined that joint properties dominate over the individual ones. When two systems are in a highly entangled state, local properties are undetermined, even conceptually, yet the measurement of local properties of the two systems leads to outcomes that are highly correlated. An example of state for two entangled photons is $(|HH\rangle + |VV\rangle) /\sqrt{2}$, which is the linear combination of the state where both photons are horizontally polarized and of the state where both photons are vertically polarized. Such a state has several properties that make it special:

- The result of a measurement of either photon in the *H / V* basis gives a random outcome.
- If both photons are measured in such a basis, the result will always be the same for the two, despite being individually random.
- By checking how the correlations change for other choices of polarizations, one can prove that nobody could have ever known the result in advance. In particular, the situation is very different from the case where the two photons are both prepared in the *H* state or in the *V* state, at random. In the latter case, it is conceivable that somebody had recorded which case and could know in advance the result of the measurement.

Entanglement and the above properties of an entangled state are strictly related to the notion of non-locality, which is a more general concept than entanglement, but it is a property displayed by entanglement.

7.1.2.2 Non-Locality

Non-locality of quantum mechanics refers to the idea that measurements in one location can influence in a non-local way (faster-than-light, if you will) the results of measurements in another location.[11]

In a modern take, assuming that quantum mechanics is the correct underlying theory, non-locality is simply a strong manifestation of entanglement that allows one to conclude that two or more systems were in an entangled state, even when one does not have full control or characterization of the measurements performed. *Non-locality ensures that the results of the local measurements could not have been known in advance, by anybody. In this sense, local measurement performed on an entangled system can be certifiably random.* Nonetheless, tests of non-locality require repeating measurements many times with a random choice of measurement performed. In this sense, non-locality experiments both produce and consume randomness. The goal is that of generating more randomness than is consumed, in a process of genuine *randomness expansion*—to be compared to the *dilution* of randomness realized by pseudorandom generators which map deterministically short random strings into long pseudorandom strings.

7.2 Survey Questions

Here is the most recent version of the list of questions that we have posed to the experts with whom we got in touch. The questions have evolved in the period during which we have interrogated the experts, also thanks to the feedback from them.

[11] This can happen while still not relying on information, thus respecting a no-faster-than-light principle about the transmission of information.

- For which threat models and/or use cases do you consider QRNGs a welcome solution, for the moment ignoring cost and performance concerns? Why? Which distinguishing features of QRNGs would be bringing what value to those use cases?
- Are there threat models and/or use cases where QRNGs *that are not device independent* offer valuable security advantages? What disadvantages, if any, do they need to be weighed against?
- Are there threat models and/or use cases where the advantages of QRNGs *that are not device independent* already outweigh such disadvantages?
- If you estimate that current performance or cost parameters may be too high to justify the use of QRNGs *that are not device independent*, can you estimate what performance and cost would be acceptable? In other words, if you believe there are meaningful benefits, but the benefits do not outweigh the current costs, what would the costs need to be for the benefits to be worth the costs? From a slightly different perspective: Which deficiency should be addressed with the ighest priority to overcome barriers to the use of such QRNGs (e.g., speed rather than cost)?
- What disadvantages, if any, do QRNGs *that are device independent* need to be weighed against?
- Are there threat models and/or use cases where the advantages of QRNGs *that are device independent* already outweigh such disadvantages?
- If you estimate that current performance or cost parameters may be too high to justify the use of QRNGs *that are device independent*, can you estimate what performance andcost would be acceptable? In other words, if you believe there are meaningful benefits, but the benefits do not outweigh the current costs, what would the costs need to be for the benefits to be worth the costs? From a slightly different perspective: Which deficiency should be addressed with the highest priority to overcome barriers to the use of such QRNGs (e.g., speed rather than cost)?
- What is your estimate for the fraction of (potential) customers that are deeply interested in the quality and certifiability of randomness, rather than in simply meeting standards regulated by law?
- Do you think that authority or law standards may soon add to or address the notion of *genuine/genuinely quantum* randomness?
- How do estimate the market for QRNGs will change in the next 10-15 years, and why? For example, will virtualization or cloud computing impact the potential value of the extra features of QRNG? Or the increasingly complex supply chain? Or increasing concerns about the trustworthiness of vendors and other players in the IT ecosystem?

7.3 List of Respondents

- Fernando G.S.L. Brandão, California Institute for Technology
- Sergio Boixo, Google Research
- Roger Colbeck, University of York
- Bruno Couillard, Crypto4A
- Martin Ekerå, KTH Royal Institute of Technology and Swedish NCSA
- Chris Erven, KETS Quantum Security
- Juan Carlos García Escartín, Universidad de Valladolid
- Scott Fluhrer, Cisco Systems
- Tim Harden, BeyondEdge Networks
- Alan Ho, Google Research
- Bruno Huttner, ID Quantique
- Brian LaMacchia, Microsoft Research
- Jason Lawlor, Lightship Security Inc.
- John Leiseboer, Quintessence Labs
- Manfred Lochter, Federal Office for Information Security (BSI), Germany
- Norbert Lutkenhaus, evolutionQ Inc.
- Ben Merriman, Cambridge Quantum Computing
- John Mulholland, evolutionQ Inc.
- Rene Peralta, NIST