



What is the Q in QRNG?

IN ORDER TO GUARANTEE ABSOLUTELY RANDOM NUMBERS, RNGS (RANDOM NUMBER GENERATORS) MUST NOT BE VULNERABLE TO PREDICTION OR BIAS, AND THUS DICTATED BY TRUE RANDOMNESS. BUT HOW CAN WE GENERATE A TRULY RANDOM NUMBER?

Instead of relying on a seed number and algorithm, an RNG can rely on an actual physical process to generate numbers. The most reliable processes are quantum physical processes, as quantum physics is fundamentally random.

In fact, the intrinsic randomness of subatomic particles' behaviour at the quantum level is one of the few completely random processes in nature. By tying the outcome of an RNG to the random behaviour of a quantum particle, it is possible to guarantee a truly unbiased and unpredictable system.

For example, physicists can record the reflection, or lack thereof, of light particles against a mirror to generate a random string of 1s and 0s. Other methods include splitting a beam of light into two beams and then measuring the light intensity of each beam as it fluctuates to generate random numbers.

Quantum RNGs generate outcomes by observing and recording a contained natural process as it happens – making it possible to perform live verification of the numbers and monitor the hardware to ensure it is operating properly.

What is a random number?

A RANDOM NUMBER IS A NUMBER GENERATED BY A PROCESS, WHOSE OUTCOME IS COMPLETELY UNPREDICTABLE. ALTHOUGH IT MAY APPEAR SIMPLE AT FIRST SIGHT TO PROVIDE THIS DEFINITION OF WHAT A RANDOM NUMBER IS, IT PROVES TO BE QUITE DIFFICULT IN PRACTICE.

A random number is a number generated by a process, whose outcome is unpredictable, and which cannot be subsequently reliably reproduced.

This definition works fine provided that one has some kind of a black box – such a black box is usually called a random number generator – that fulfils this task.

However, if one were to be given a number, it is simply impossible to verify whether it was produced by a random number generator or not. It is hence absolutely essential to consider sequences of numbers in order to study the randomness of the output of such a generator.

It is quite straightforward to define whether a sequence of infinite length is random or not. This sequence is random if the quantity of information it contains – in the sense of Shannon's information theory – is also infinite.

In other words, it must not be possible for a computer program, whose length is finite, to produce this sequence. Interestingly, an infinite random sequence contains all possible finite sequences.

Such an infinite sequence does for example contain the Microsoft Windows source code or the text of the Geneva conventions. Unfortunately, this definition is not very useful, as it is not possible in practice to produce and process infinite sequences.

AN INFINITE SEQUENCE SHOULD NOT BE COMPRESSABLE BY ANY KNOWN OR UNKNOWN TECHNIQUE.

In the case of a finite sequence of numbers, it is formally impossible to verify whether it is random or not. It is only possible to check that it shares the statistical properties of a random sequence – like the equi-probability of all numbers – but this a difficult task.

To illustrate this, let us for example consider a binary random number generator producing sequences of ten bits. Although it is exactly as likely as any other ten bits sequences, 1 1 1 1 1 1 1 1 1 1 does look less random than 0 1 1 0 1 0 1 0 0 0.

In order to cope with this difficulty, definitions have been proposed to characterize "practical" random number sequences.

According to Knuth, a sequence of random numbers is a sequence of independent numbers with a specified distribution and a specified probability of falling in any given range of values.

For Schneier, it is a sequence that has the same statistical properties as random bits, is unpredictable and cannot be reliably reproduced.

A concept that is present in both of these definitions and that must be emphasized is the fact that numbers in a random sequence must not be correlated. Knowing one of the numbers of a sequence must not help predicting the other ones.

Whenever random numbers are mentioned in the rest of this paper, it will be assumed that they fulfil these "practical" definitions.

Generating random numbers

IN A BINARY SCENARIO, THE PROBABILITY TO OBTAIN 1 VS 0 SHOULD BE EQUAL TO $\frac{1}{2}$. A RANDOM NUMBER GENERATOR IS A DEVICE THAT PRODUCES SEQUENCES OF NUMBERS COMPLYING WITH THE DEFINITIONS PROPOSED ABOVE.

TYPES OF RANDOM NUMBER GENERATOR

- PSEUDO RNG
- CHAOTIC PHYSICAL RNG
- QUANTUM PHYSICAL RNG

There exist two main classes of generators: software and physical generators. From a general point of view, software generators produce so-called pseudo random numbers. Although they may be useful in some applications, they should not be used in most applications where randomness is required. These two classes, as well as their respective advantages, are discussed below.

SOFTWARE SOLUTIONS

COMPUTERS ARE DETERMINISTIC SYSTEMS. GIVEN A CERTAIN INPUT, A PROGRAM WILL ALWAYS PRODUCE THE SAME OUTPUT.

Because of this very fundamental property, it is impossible for a program to produce a sequence of random numbers. The sequence may have some of the properties of a random sequence, and thus pass some statistical randomness tests, but it is always possible to reproduce it.

As the sequences they produce look like random sequences, these generators are called pseudo-random number generators. It is however clear that they do not fulfil the definitions given at the beginning of this white paper.

Pseudo-random number generators consist of an algorithm into which some initial value – it is called the seed – is fed and which produces by iteration a sequence of pseudo-random numbers.

Although their period can be made very long, the sequence produced by such a generator is always periodic. When working with large sets of random numbers or long sequences, it is important to verify that the period is large enough.

One of the properties of the sequences produced in this way is that, as soon as one element of the sequence is known, all the other elements of the sequence, both preceding and following, can be determined.

It is immediately obvious that this property is especially critical when such numbers are used in cryptography for key generation. The sequences produced by a good pseudo-random generator initialized with an appropriate seed, pass most statistical tests.

However, it is also important to realize that if the sequence considered is long enough, it will always fail at least some tests, because of periodicity.

OUTPUT IS DETERMINED ONLY BY THE SEED AND THE GENERATING ALGORITHM.

One important issue when using pseudo-random number generators is the choice of the seed value. If one does not want to continuously cycle through the same sequence, it is essential to change periodically the seed value.

How should this seed value be chosen? Ideally, it should be random. As a pseudo-random generator is used, it is likely that no random number generator is available. It is a catch-22 situation.

A solution, which is not always satisfactory, is to use entropy gathering. System information – the clock, the time interval between keystrokes, etc. - is combined to produce a seed. This technique however requires extreme caution.

THE SAME SEED WILL ALWAYS GENERATE THE SAME OUTPUT

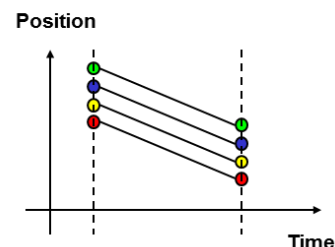
A security problem encountered by Netscape with its browser in 1995 illustrates the risks associated with the use of pseudo-random numbers in cryptography.

At the time, the web was full of promises for online commerce and Netscape had developed a protocol, called SSL and still in use today, to secure communications over the web.

Like in the case of other cryptographic protocols, the security of SSL crucially depends on the unpredictability of the key. The company implemented this protocol in its browser, but relied on a pseudo-random number generator for key generation.

Two Berkeley graduate students reverse-engineered the code of the browser and revealed a serious security flaw. They noticed that the seed used by the pseudo-random number generator depended on the time of the day and some system information (the process ID and the parent process ID).

Deterministic evolution



They showed that it was relatively easy to guess these quantities, and thus to reduce the number of possible keys that one should try to crack the protocol.

This attack reduced the time for a brute force attack of the protocol from more than thirty hours to a few minutes, and as little as a few seconds in some special cases.

This problem is also serious in other industries. For example, in 2014, a group of hackers managed to make millions by targeting and hacking slot machines from a well-known brand in the gaming industry. Casinos from US to Romania to Macau were the main victim of the scam.

It appeared that the hackers managed to figure out how to predict the slot machines' PRNG behaviour.

After a prolonged observation and analysis of the slot machines' behaviour, they could identify the pattern of the PRNG's algorithm that was used in the slot machines.

This security flaw illustrates why pseudo-random number generators are usually considered inappropriate for high-security cryptographic or other applications.

In spite of the fact that they do not produce random numbers, such PRNG generators do have some advantages. First, their cost is virtually zero, as they can be implemented in software and numerous libraries are freely available. Second, their main disadvantage – namely that the sequences they produce are reproducible – can in certain cases represent an advantage.

When using random numbers for scientific calculations, it is sometimes useful to be able to replay a sequence of numbers when debugging the simulation program.

This is one of the reasons – the other being the lack of good physical random numbers generators – why these generators are widely used in these applications.

However, it is important, even in this field, to remain careful when using pseudo-random numbers. They have been shown to cause artefacts in certain simulations.

A good approach is to use pseudo-random numbers when debugging, and then resort to a physical generator to refine and confirm the simulation.

PHYSICAL SOURCES OF RANDOMNESS

IN APPLICATIONS WHERE PSEUDO-RANDOM NUMBERS ARE NOT APPROPRIATE, ONE MUST RESORT TO USING A PHYSICAL RANDOM NUMBER GENERATOR. WHEN USING SUCH A GENERATOR, IT IS ESSENTIAL TO CONSIDER THE PHYSICAL PROCESS USED AS THE RANDOMNESS SOURCE.

This source can be either based on a process described by classical physics or by quantum physics. Classical physics is the set of theories developed by physicists hundreds of years ago, which describes macroscopic systems like falling coins. Quantum physics is a set of theories elaborated by physicists during the first half of the 20th century and describes microscopic systems like atoms or elementary particles. Some examples of generators based on each of these theories, along with their advantages, are presented below, after a brief discussion of biased random number sequences.

BIASED AND UNBIASED SEQUENCES

A problem encountered with physical random number generators is their bias. A binary generator is said to be biased when the probability of one outcome is not equal to the probability of the other outcome.

Bias arises because of the difficulty to devise precisely balanced physical processes. It is however less of a problem than one might expect at first sight. There exists some post-processing algorithm that can be used to remove bias from a sequence of random numbers.

The simplest of these unbiasing procedures was first proposed by Von Neumann. The random bits of a sequence are grouped in sub sequences of two bits. Whenever the two bits of a subsequence are equal, it is discarded.

When the two bits are different and the subsequence starts with a 1, the subsequence is replaced by a 1. When it starts with a 0, it is replaced by a 0. After this procedure, the bias is removed from the sequence.

The cost of applying an unbiasing procedure to a sequence is that it is shortened. In the case of the Von Neumann procedure, the length of the unbiased sequence will be at most 25% of the length of the raw sequence.

We mentioned previously that randomness tests basically all amount to verifying whether the sequence can be compressed. An unbiasing procedure can be seen as a compression procedure. After its application, the bias is removed and no further compression is possible, guaranteeing that the sequence will pass the tests. Other unbiasing procedures exist. The one proposed by Peres, for example, is significantly more efficient than the Von Neumann procedure.

PROCESSES DESCRIBED BY CLASSICAL PHYSICS – DETERMINISM HIDDEN BEHIND COMPLEXITY

Macroscopic processes described by classical physics can be used to generate random numbers. The most famous random number generator – coin tossing – indeed belongs to this class.

However, it is very important to realize that classical physics is fundamentally deterministic. The evolution of a system described by classical physics can be predicted, assuming that the initial conditions are known.

In the case of a coin, a physicist knowing precisely its weight, its initial position, the force applied to it by the hand, the speed of the wind, as well as all the other relevant parameters, should in principle be able to predict the outcome of the throw.

Why is that, that in practice this prediction is not possible? Coin tossing is a chaotic process. Chaos is a type of behaviour observed in systems whose evolution exhibits extreme sensitivity to initial conditions.

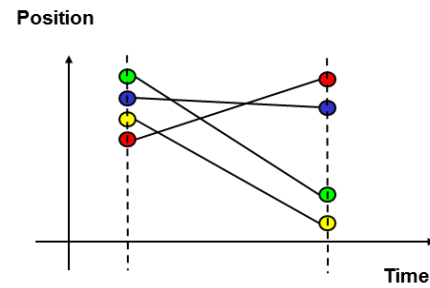
Coin tossing is not the only physical system with chaotic evolution. Turbulences in a flow (turbulences in a lava lamp have been used to generate random numbers) or meteorological phenomena are good examples of chaotic systems.

The evolution of these systems is so sensitive to initial conditions that it is simply not possible to determine them precisely enough to allow reliable prediction of future transformations.

In spite of its popularity, coin tossing is clearly not very practical when many random events are required.

Other examples of physical random number generators based on chaotic processes include the monitoring of an electric noise current in a resistor or in a Zener diode. Formally, the evolution of these generators is not random; just very complex. One could say that determinism is hidden behind complexity.

Chaotic evolution



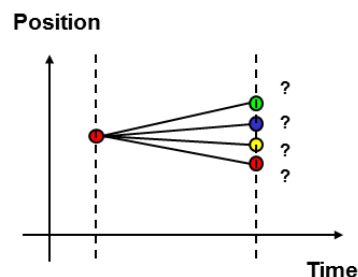
Chaotic evolution

Although their random numbers are likely to pass randomness tests, these generators are difficult to model. This means that it is impossible to verify, while acquiring numbers, that they are operating properly. In addition, it is difficult to ensure that the system is not interacting – even in a subtle way – with its environment, which could alter the quality of its output.

PROCESSES DESCRIBED BY QUANTUM PHYSICS –RANDOMNESS REVEALED BY SIMPLICITY

Contrary to classical physics, quantum physics is fundamentally random. It is the only theory within the fabric of modern physics that integrates randomness.

Quantum physics guarantees random evolution



This fact was very disturbing to physicists like Einstein who invented quantum physics. However, its intrinsic randomness has been confirmed over and over again by theoretical and experimental research conducted since the first decades of the 20th century.

When designing a random number generator, it is a natural choice to take advantage of this intrinsic randomness and to resort to the use of a quantum process as source of randomness.

Formally, quantum random number generators are the only true random number generators. Although this observation may be important in certain cases, quantum random number generators have other advantages.

This intrinsic randomness of quantum physics allows selecting a very simple process as source of randomness. This implies that such a generator is easy to model and its functioning can be monitored in order to confirm that it is operating properly and is actually producing random numbers.

Contrary to the case where classical physics is used as the source of randomness and where determinism is hidden behind complexity, one can say that with quantum physics randomness is revealed by simplicity.

Until recently the only quantum random number generator that existed were based on the observation of the radioactive decay of some element. Although they produce numbers of excellent quality, these generators are quite bulky and the use of radioactive materials may cause health concerns. The fact that simple and low cost quantum random number generators did not exist prevented quantum physics to become the dominant source of randomness.

The Quantis Quantum Random Number Generator

IN 2001, ID QUANTIQUE INTRODUCED THE FIRST COMMERCIAL QUANTUM RANDOM NUMBER GENERATOR, WHICH GENERATED STRONG INTEREST. QUANTIS IS A STATE OF THE ART QUANTUM RANDOM NUMBER GENERATOR, EXPLOITING AN OPTICAL QUANTUM PROCESS AS THE SOURCE OF RANDOMNESS.

Quantis is unique in that it relies on quantum physics to produce a high bit rate of 4 to 16 Mb/sec of truly random bits. The product comes as a USB device, a PCI express (PCIe) card as well as an OEM component. It is possible to download random numbers produced by a Quantis Quantum Random Number Generator by visiting www.randomnumbers.info

PRINCIPLE

Optics is the science of light. From a quantum physics point of view, light consists of elementary "particles" called photons. Photons exhibit in certain situations a random behaviour.

One such situation, which is very well suited to the generation of binary random numbers, is the transmission upon a semi-transparent mirror. The fact that a photon incident on such a component be reflected or transmitted is intrinsically random and cannot be influenced by any external parameters. Figure 1 schematically shows this optical system.

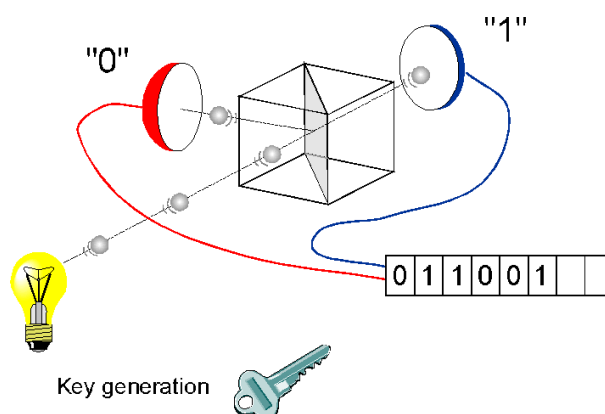


Figure 1: Optical system used to generate random numbers

Figure 2 shows the block diagram of the Quantis random number generator, which consists of three subsystems.

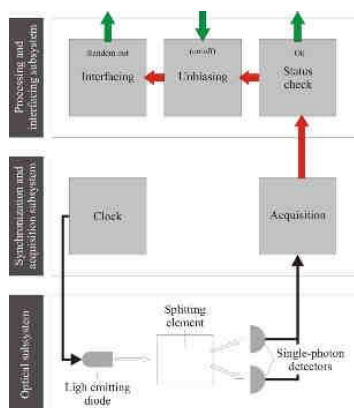


Figure 2: Block diagram of the Quantis RNG

The first one is the core of the generator and contains the optical elements that are used to implement the random process and produce the random outcomes. It comprises a light emitting diode producing the photons, a transmission element, where the random process takes place, and two single-photon detectors – detectors with single-photon resolution – to record the outcomes.

The optical subsystem is controlled by a synchronization and acquisition electronic circuit. This subsystem comprises a clock and triggering electronics for the photon source, as well the acquisition electronics for the single-photon detectors. The processing and interfacing subsystem perform statistical and hardware checks, as well as unbiasing of the sequence. These operations are discussed in more details in the following subsection. This subsystem also shapes the output electronic signals.

UNBIASING OF RANDOM NUMBERS

As mentioned above, physical processes are difficult to precisely balance. It is thus difficult to guarantee that the probability of recording a 0, respectively a 1, is exactly equal to 50%.

With Quantis, the difference between these two probabilities is smaller than 10% - or equivalently the probabilities are comprised between 45% and 55%. As this bias may not be acceptable in certain applications, the processing unit of Quantis performs unbiasing of the sequence.

STATUS MONITORING

One of the main advantages of quantum random number generators is that they are based on a simple and fundamentally random process that is easy to model and monitor.

The processing unit of Quantis performs a live verification of its functioning. It continuously checks that the light source and the two detectors are correctly working, and that the raw output stream statistics are within certain bounds.

A status bit is output by Quantis. If all the conditions are fulfilled, this bit is equal to 1. If one of the conditions is not fulfilled, the status bit is set to 0 and the bit stream is inhibited. Thanks to this feature, the users of Quantis can have a high level of trust in the random numbers they are using.

CERTIFICATIONS

The simplicity of Quantis is also its strength. As the underlying quantum mechanical processes are well understood and easily characterized, it is relatively easy to certify the Quantis products.

Quantis is the most certified true RNG in the market. It has successfully passed the following certifications or government validations:

- NIST SP800-22 Test Suite Compliance
- METAS Certification
- CTL Certification
- Several iTech Labs individual Certificates
- Compliance with the BSI's AIS31 standard (dedicated version of Quantis)

FORM FACTORS

Quantis is available as a component, in the form of a compact metal package that can be mounted on plastic circuit boards (see Figure 3).



Figure 3: The Quantis random number generator, available as an OEM component (left) and as a PCI-card (right), offers high-quality random numbers at a speed of up to 16 Mbits/sec.

It is also available as a USB device, a PCI Express (PCIe) card, that can be installed in a computer. Refer to www.idquantique.com for more information on Quantis.

Summary

- Random number generation is a critical security and reliability criterion in many demanding applications.
- Because of its intrinsic randomness, quantum physics is an excellent source of randomness.
- Quantis is a compact, low cost and easy to use random number generator exploiting a quantum optical process as the source of randomness.
- Quantis features a high bit rate output stream (up to 16 mbps) which does not exhibit any correlations and passes all statistical tests.

WHEN THE GENERATION OF RANDOM NUMBERS CANNOT BE LEFT TO CHANCE, USE QUANTIS.