IDQ
FROM VISION TO TECHNOLOGY

REDEFINING RANDOMNESS

QUANTIS

WHEN RANDOM NUMBERS CANNOT BE LEFT TO CHANCE

ID Quantique White Paper

Randomness Extraction for the Quantis True Random Number Generator

Version 1.0
September 2012

ID Quantique SA
Chemin de la Marbrerie 3
1227 Carouge/Geneva
Switzerland
T +41 22 301 83 71
F +41 22 301 83 79
info@idquantique.com
www.idquantique.com

Contents

ID Quantique SA
Chemin de la Marbrerie 3
1227 Carouge
Switzerland

Tel: +41 22 301 83 71
Fax : +41 22 301 83 79
www.idquantique.com
info@idquantique.com

Quantis is a True Random Number Generator which exploits quantum physics to produce random bits. It is a perfect entropy source based on the reflection or transmission of a light particle – a photon – on a semi-transparent mirror. However, like any other physical randomness source, Quantis produces strings of bits which are slightly longer than their entropy. This excess length is minute and is in most cases not noticeable, as extremely long sequences must be considered. In addition, these extra bit symbols can be removed using a so called Randomness Extractor. This white paper describes the principle of randomness extraction and explains how it is implemented in the Quantis software package.

Introduction

Random numbers are essential in various applications, from cryptography to numerical simulations and internet lotteries. However, generating good random numbers on a computer is a difficult task. Algorithms, known as Pseudo-Random Number Generators, can be used to produce sequences of numbers which exhibit the statistical properties of a random number sequence. However, with such generators, one number in the sequence is entirely determined by previous numbers. These generators mimic randomness, but fail to produce unpredictability. Techniques have been developed to solve this problem and allow operating systems to collect unpredictability from their environment – for example from peripherals – which can then be introduced in a software generator. This approach is however difficult to master and has risks, as recently illustrated by the research[1] of Lenstra [1] and Heninger [2]. The use of a dedicated physical device providing high-quality fresh randomness – a True Random Number Generator or TRNG – to a computer allows to solve this problem and guarantees that suitable random numbers are always available.

Quantis TRNG

Quantis is a TRNG developed by ID Quantique (IDQ) and commercially available since 2004. It exploits the random behavior of photons – photons are "particles" of light – to produce bits whose randomness stems from the laws of quantum physics. Photons are sent one by one on a semi-transparent mirror, where they are either transmitted or reflected before going to two detectors, which are associated with bit values of 0 and 1. The laws of quantum physics stipulate that each photon chooses randomly whether to be reflected or transmitted (see the "Random Number Generation using Quantum Physics" IDQ white paper for more detailed explanations [3]). Moreover, this choice cannot be influenced by external parameters. This scheme makes it possible for the Quantis TRNG to produce high-quality random bits at a bit rate of up to 16 Mbps. The Quantis TRNG is available as a USB device or as a PCI/PCIexpress card and can be used with the most common operating systems. It has been certified by Metas – the Swiss Federal Office of Metrology – and evaluated by several governments. The quality of the random numbers it produces and its ease of use make Quantis the leading TRNG in critical applications, from cryptographic key generation to online gaming and lotteries.

No matter how elegant its principle is and how carefully designed a TRNG is, a particular physical realization will by definition have imperfections, which can in turn impact the quality of its output. The random sequence can for example exhibit a bias – unequal probability of generating a 0 and a 1 – or correlations – probability dependant on previous bit values. In spite of its qualities, the Quantis TRNG exhibits some – very low – residual bias and correlations. The bias is caused by differences in photon detection efficiency between the detectors and correlations arise from memory effects in these components[2]. The imperfections in

[1] After collecting several millions of cryptographic keys over the Internet, these researchers analysed them and found a proportion of close to 1% of weak keys, a surprisingly high number – linked to problems in the way random numbers were generated in the system.

[2] The Quantis TRNG uses avalanche photodiodes for photon detection. These components are optolectronic

the output of Quantis are very low and not noticeable in most applications, as extremely long bit strings must be analyzed to reveal these deviations. In certain cases, it can however be necessary to post-process the output of the Quantis TRNG in order to remove these imperfections. This can be done using a so-called randomness extractor, which will be described in the next sections of this paper.

Quantifying Randomness

Testing and quantifying randomness is a challenging task. Given a finite sequence of bits, it is impossible to prove whether it consists of random bits or not, without knowledge of the process used to generate them. What is usually done instead is to analyze the statistical properties of the sequence – the number of 0's and 1's for example is one such property – and to compare them with the values expected in the case of an ideal random number source. A vast number of statistical tests and test suites have been developed for this task, but they remain empirical and are typically tailored to detect specific deviations from perfect randomness.

Randomness has to do with unpredictability, for which information theory has a quantitative definition. The concept of entropy, first introduced by Shannon, measures the uncertainty associated with a random variable and is expressed in bits. A fair coin toss has an entropy of 1 bit, as the exact outcome – head or tail – cannot be predicted. If the coin is unfair, the uncertainty is lower and so is the entropy. And when tossing a two-headed coin, there is no uncertainty which leads to 0 bit of entropy.

The concept of entropy can be used to quantify the amount of randomness in a sequence of random bits and estimate deviations from that produced from a perfect random source, producing 1 bit of entropy per binary symbol.

Randomness Extractor

A Randomness Extractor is a mathematical function which can be used to postprocess an imperfect sequence of random bits into an output which is shorter, but more random. More formally, the imperfect input sequence has an entropy of less than 1 bit per binary symbol. As explained above, this can be caused by imperfections, such as bias or correlations, in the physical process used to produce the raw random sequence. The effect of the randomness extractor is to compress the sequence so that the post-processed entropy approaches 1 bit per symbol[3].

This principle can be illustrated by an analogy with toasts. Assume that you have a fixed amount of jam to spread on a toast, the quantity of jam and the size of the slice of bread will determine the thickness of the layer. This thickness is analogous to the entropy per symbol in the random sequence. Now, if the jam is collected and respread on a smaller toast, the thickness will increase, which is similar to the result obtained with a randomness extractor.

The quality of a randomness extractor is defined by the probability that the output deviates from a perfectly uniform bit string. While a value of 0 is normally not achievable for this probability, it can be made arbitrarily small by increasing the compression factor. The value of this factor depends on the entropy[4] of the raw sequence and the targeted deviation probability and must be adjusted accordingly.

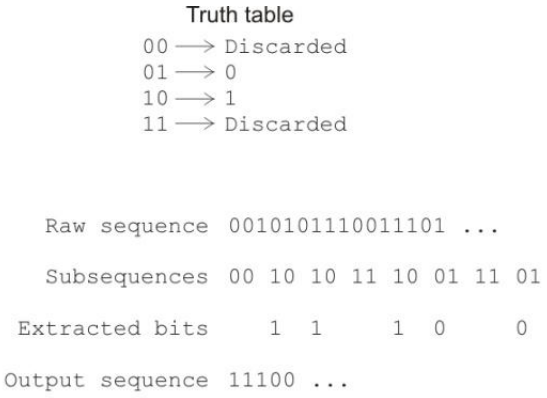

Figure 1: Von Neumann Unbiasing Algorithm

devices which can transform photons into macroscopic electronic pulses.

[3] Readers familiar with quantum information theory and quantum cryptography will recognize the similarity of the concept of randomness extraction with that of privacy amplification. This topic goes beyond the scope of this white paper, but we will note here that the same mathematical functions can be used for both of these applications.

[4] Note that in this context, the relevant entropy is the collision entropy and not the Shannon entropy. However, the difference between these two quantities goes beyond the scope of this white paper and will not be elaborated further.

A well known example of a randomness extractor is the Von Neumann unbiasing algorithm, which can be used to transform a biased sequence of bits into a shorter unbiased string. The bits are grouped in subsequences of two bits. If the two bits of the sequence are equal, it is discarded. When the two bits are different, the subsequence is replaced by its first bit. Applying this procedure removes the bias as the output bits have the same probability. This comes from the fact that probability of the subsequences with differing bits values ("01" and "10") is equal: $p_0 \times p_1 = p_1 \times p_0$, where p_i represents the probability of obtaining a bit value of i in the raw sequence. The compression effect of this procedure comes from the fact that subsequences with equal bit values are discarded and that subsequences with differing values are compressed to one bit. The compression factor depends on the bias. In the best case, when the probability of generating a 0 and a 1 are equal[5], the output string is 75% shorter than the input sequence. This compression ratio increases with the bias and eventually reaches 100% for a fully biased sequence (a sequence consisting only of one bit value).

The Von Neumann unbiasing algorithm is an example of a so-called deterministic randomness extractor. Given a particular input, a deterministic extractor will always produce the same output. This kind of extractors is well suited to removing bias, but will generally not remove all imperfections. In particular, deterministic extractors will not correct problems arising from correlations. Fortunately, a second class of algorithms, known as non-deterministic extractors, can be used for this. In addition to the raw sequence, non-deterministic extractors also take a seed as input. The raw input sequence and the seed are combined by the extractor to produce the output string.

The prime advantage of using a randomness extractor is that it allows to mathematically prove the quality of the output random sequence and base this proof on the well established information theory. In addition, non-deterministic extractors also offer a level of protection against a malevolent TRNG manufacturer, who introduces imperfections in the TRNG in order to reduce the randomness of the output, for example to make exhaustive searches easier. By customizing the seed, the end-user has the possibility to reduce the advantage of his adversary.

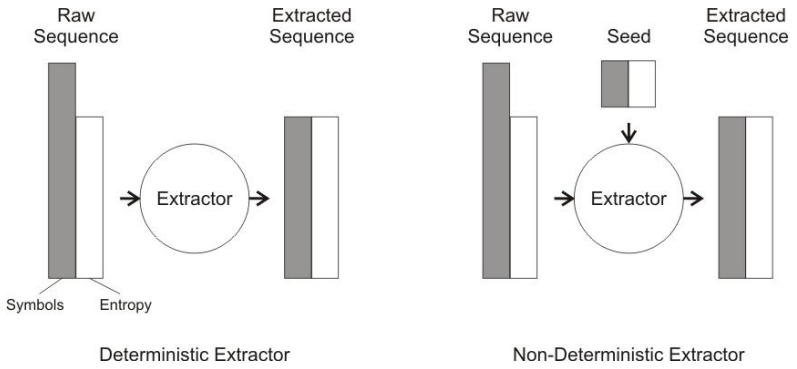

Figure 2: Deterministic / Non-Deterministic Randomness Extractor

Randomness Extraction in the Quantis Software Package

A non-deterministic randomness extractor based on Universal-2 hash functions has been implemented in the Quantis software package. The mathematical details are not presented in this paper, but can be found in reference [4]. In practice, performing randomness extraction consists in multiplying a rectangular binary matrix M with a bit vector produced by the Quantis TRNG, and corresponding to the raw sequence, to produce the extracted sequence. The matrix M is the seed. The properties of matrix multiplication require the number of columns of M to be equal to the number of bits in the input vector. The length of the output vector is equal to the number of lines of M. The compression ratio is thus equal to the number of lines divided by the number of columns of M. The binary multiplication of a matrix with a vector can be

[5] Of course, in this case using an unbiasing procedure is not really necessary.

efficiently implemented using bitwise operations. The generation of the matrix M is crucial to guarantee that the properties of the output of the extractor meet the requirements in terms of randomness quality. Each position in the M matrix should consist of an unbiased random bit, completely independent of other bits. Note that the matrix can be the same for multiple Quantis TRNG and can be reused. It must thus be generated only once.

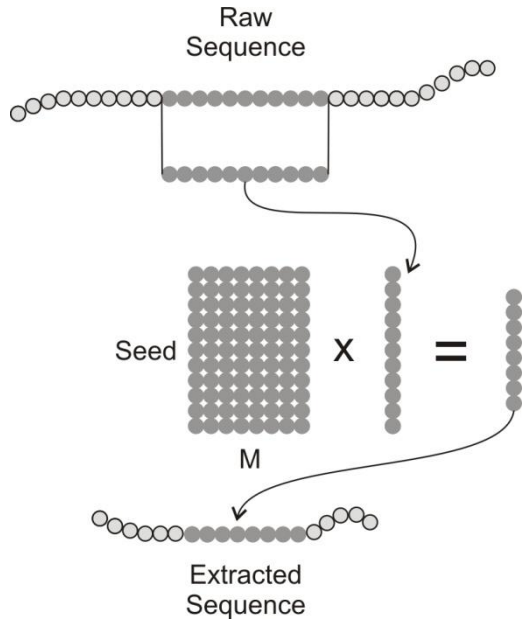

Figure 3: Randomness extraction in the form of a seed matrix multiplication

The Quantis TRNG is normally accessed using a library called `libQuantis`. The randomness extractor has been implemented in a new library `libQuantisExtensions` which reads raw random bits using `libQuantis` and processes them for randomness extraction (see Figure 4). This approach allows to maintain backward compatibility for existing applications. Users of Quantis can also upgrade their applications to use random bits post-processed by the randomness extractor simply by replacing the function of the `libQuantis` library with its correspondent in the `libQuantisExtensions` library.

The seed matrix M is stored in a file in the software package. The generation of this matrix is non trivial, but it can be shown that one possibility is to add (modulo 2) random bits coming from mutually independent imperfect random source to produce each element, provided that the number of sources is sufficiently large. In practice, several Quantis TRNG's have been used. Their raw output has been undersampled with a time much larger than the maximum possible correlation time. The matrix M included in the software package is the same for all Quantis TRNG's. It contains 1024 lines and 768 columns. This implies that the raw sequence is cut in vectors of 1024 bits, which are multiplied one by one with the matrix M to produce an extracted sequence of 768 bits. The efficiency of this extraction is obviously approximately 75%. This compression factor has been selected to guarantee that the probability for the extracted sequence to deviate from a perfectly random sequence is less than 2^{-100}. This implies that even if he had millions of Quantis devices, a user will not see any deviation from perfect randomness in a time longer than the age of the universe. Note that using a larger matrix could improve the efficiency, but would also increase the computational complexity of the extraction.

The price to pay when using this randomness extractor is a reduction of the useful bit rate produced by a Quantis TRNG by approximately 25%. A Quantis-USB-4M TRNG, which produces 4 Mbps of raw randomness, will then allow to generate 3 Mbps after extraction. The postprocessing also uses some CPU resources. It should be noted though that these bit rates normally exceed what is needed in most applications.

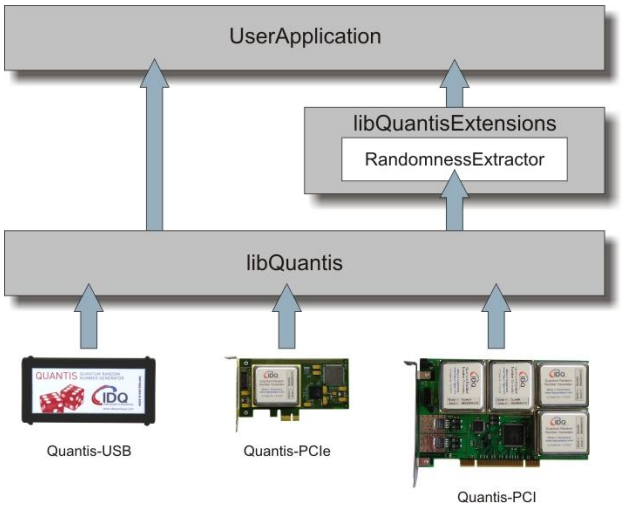

Figure 4: Software Access to the Quantis TRNG

The Quantis software package also includes an application called EasyQuantis will allows to read random bits from a Quantis TRNG in different formats (integer, binary, floating point) and to

perform scaling, before saving the data. The randomness extractor can also be used with EasyQuantis from version 2.0 and above. This tool can also be used to generate a new seed matrix from the output of at least one Quantis TRNG.

Conclusion

The randomness extractor implemented in the Quantis software package allows to produce bits whose randomness can be mathematically proven and to correct the minute imperfections, such as residual bias, of the output of the Quantis TRNG. It is implemented in a new library which ensures easy and seamless integration, while maintaining backward compatibility.

References

[1] Lenstra, A.K., Hughes, J.P., Augier, M., Bos, J.W., Kleinjung, T., Wachter, C.: Ron was wrong, Whit is right. http://eprint.iacr.org/2012/064 (2012)

[2] Heninger, N., Durumeric, Z., Wustrow, E., Halderman, J. A.: Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices.
https://factorable.net/weakkeys12.conference.pdf

[3] Random Number Generation using Quantum Physics. ID Quantique White Paper. http://www.idquantique.com/images/stories/PDF/quantis-random-generator/quantis-whitepaper.pdf

[4] Technical paper on randomness extractor. ID Quantique White Paper