WHITEWOOD

STRENGTHENING THE SECURITY FOUNDATION OF CRYPTOGRAPHY WITH WHITEWOOD'S QUANTUM-POWERED ENTROPY ENGINE

An overview of cryptographic randomness requirements for the data center/ cloud, the limitations of previous methods, and Whitewood's solution

Richard Hughes and Jane Nordholt

January 2016

whitewoodsecurity.com

WHITEWOOD 100 High Street 28th Floor Boston, MA 02110 USA
p. +1.617.391.0268 e. info@whitewoodsecurity.com

OVERVIEW

The security of the cryptography that makes much of our modern economy possible rests on the random numbers used for secret keys, public key generation, session identifiers, and many other purposes. The random number generator (RNG) is therefore a potential single point-of-failure in a secure system. But despite this critical importance, there continues to be difficulty in achieving high assurance random number generation in practice. The requirements for cryptographic random numbers – uniformity and independence, unpredictability and irreproducibility, and trust and verifiability – are clear, but the range of techniques in use today to create them varies enormously in terms of satisfying those requirements. Computational methods are fundamentally deterministic and when used alone are not sufficient for cryptographic use. Physical unpredictability (entropy) is a necessary ingredient in a cryptographic RNG. Providing sufficient entropy with assurances that it cannot be known, monitored, controlled or manipulated by third parties is remarkably challenging.

Fortunately, the quantum physics of matter and light at the atomic scale can provide an entropy source with dramatically higher rates of production and the ultimate security guarantee of a Law of Nature. In contrast to everyday physical phenomena, the outcome of certain quantum processes cannot be predicted by anyone, no matter what technology, present or future, they possess. It is now possible to capture randomness intrinsic to the behavior of photons – particles of light – to make a quantum random number generator (QRNG). While previous commercial QRNG designs require the detection of fragile single-photon signals, at exceedingly small power levels, Whitewood's Entropy Engine utilizes optical power levels more typical in telecommunications systems, which are ten orders of magnitude higher. By exploiting the natural, quantum-mechanically random behavior of many photons acting together we are able to produce a robust and high performance entropy source. The Whitewood Entropy Engine is a cost-effective hardware solution that provides cryptographic random numbers at a 200Mbps rate that easily satisfies present and projected demands within data centers and cloud providers, and that is compatible with national and international design standards for true random number generators (TRNG).

1. INTRODUCTION

"Random number generation is too important to be left to chance."
(R.R. Coveyou, 1970 [1])

One of the great breakthroughs in cryptography was reducing the problem of secure communications to the generation of random numbers for use as keys [2]. Rather than using a different secret cipher for every communication ("security by obscurity"), a small number of cryptographic algorithms could be standardized after being subjected to open peer review, and used with distinct, relatively short, secret keys as parameters for different channels and purposes. Random number generation therefore underpins all of modern cryptography – public/private key pair generation; symmetric key generation; initialization vectors; session identifiers; random challenges; and salt for stored password hashes. This makes the random number generator (RNG) a potentially catastrophic single point of failure [3], which is an even greater concern with the increasingly rapid adoption of cloud-based services. It is essential to provide trusted, verifiable randomness that meets the requirements of unpredictability and irreproducibility in sufficient quantities for both present and projected demands of the data center and cloud environments [4]. For this reason regulators, standards bodies and certifications, such as FIPS 140 [5], for cryptographic products are placing an increasing emphasis on validating vendor claims for random number generation .

It is believed that an inadequate supply of randomness underlies the discovery that far more RSA public keys are identical than would be expected by chance [6,7]. This reduces the strength of the security: the amount of computational work that would be expected to break only a single public key would instead break many. Not surprisingly there has been a steady drum beat of revelations of weak RNG vulnerabilities, and heightened concerns of exploits [8,9] using cryptographic backdoors introduced by systematically weakening RNGs [10]. The infamous Netscape Web browser SSL implementation flaw in the early 1990s [11] was an early example of inadequate randomness - the RNG output was too easily predictable for security. Multiple RNG weaknesses have been pointed out since, including: the Debian SSL RNG implementation [12]; randomness for virtual machines [13]; and the PlayStation 3 digital signature implementation [14]. The potential exploitability of RNG weaknesses has been outlined [15], and the feasibility of undetectable weakening of Intel's Bull Mountain RNG has been demonstrated [16,17]. With an eye to the future, an insufficient supply of randomness was the likely cause of BitCoin transfers going to an unintended entity [18].

WITHOUT ENOUGH ENTROPY, EXPLOITATION BECOMES EASY
MORE ENTROPY → MORE SECURITY

Several trends are placing even greater demands on RNGs. These include: increased awareness of online security with the widespread use of TLS and https everywhere; the desire for online privacy driving the deployment of perfect forward secrecy [19] using ephemeral Diffie-Hellman [20] and end-to-end encryption; and anticipation of a transition to post-quantum cryptography [21] with a variety of new protocols, such as one-time signatures [22] under consideration, which will require copious randomness.

The underlying problem is an insufficient supply of entropy – the fundamental physical quantification of unpredictability. Entropy cannot be provided by software methods alone - computer systems depend on predictable, deterministic functionality – and some analogue, physical "noise" must be utilized. Although there's no shortage of entropy in the physical world, finding and capturing a suitable noise source capable of meeting cryptographic requirements is the challenging problem addressed by true random number generators (TRNG).

Previously available TRNG products do not adequately address the security requirements of the data center/cloud environment. Products targeted at end-user devices are too slow and lack the appropriate interfaces; others use hardware whose operation is infeasible to verify; and many utilize entropy signals of questionable quality. There are concerns about the extent to which various existing TRNGs can be trusted. For example, the trust basis of TRNGs utilizing noise sources that are governed by the laws of everyday (macroscopic, or "classical") physics is the assumed difficulty of determining or influencing the system. But there is no limitation in principle with classical noise sources to an adversary obtaining information, whether by passive monitoring, malicious modification or signal injection.

CLASSICAL NOISE SOURCES CAN BE PASSIVELY MONITORED → UNKNOWN ADVERSARIAL INFORMATION

However, quantum random number generators (QRNG), which utilize noise sources governed by the laws of the microscopic, quantum physics of atoms and photons, provide dramatically higher levels of trust and verifiability because certain quantum processes are irreducibly random [23]. No one, no matter how much time, money or effort they devote could ever predict or influence the outcome - QRNGs have the ultimate guarantee of a Law of Nature.

Over the last decade various attempts have been made to produce a viable QRNG but these approaches have suffered practical limitations. Firstly, they have been limited in the rate at which

random numbers can be generated, and secondly they often utilize expensive components with strict manufacturing processes. Many utilize the behavior of individual photons, creating the requirement to detect exceedingly faint optical signals and to screen out all other extraneous noise. However, in this white paper we will describe Whitewood's Entropy Engine [24] – a new commercial product that uses a macroscopic quantum effect to provide full quantum entropy (one bit of quantum entropy per output bit) at the rates necessary for data center, cloud, large-scale internet of things (IoT), and mobile applications.

The rest of this white paper is organized as follows. In Section 2 we will review the randomness requirements for cryptography and the different classes of RNGs. Section 3 will cover notions of entropy for cryptography. In Section 4 we will describe approaches to TRNGs using physical noise, and in Section 5 we will review the randomness of quantum physics. Section 6 provides an overview of the Whitewood Entropy Engine. Section 7 summarizes the main points.

2. RANDOMNESS, CRYPTOGRAPHY AND TYPES OF RANDOM NUMBER GENERATORS

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin." (J. von Neumann, 1951 [25].)

Random numbers are used in a surprisingly wide array of applications including numerical simulation [26], gambling [27] and lotteries, as well as cryptography. They are also used in great quantity within operating systems to organize software processes. These applications have quite distinct requirements, so that an RNG devised for one class of application may be completely unsuitable in another. For example, reproducibility of the RNG output would be desirable in numerical simulation, making it possible to repeat a numerical experiment, but would be completely unacceptable for cryptography.

The term "random number" is used rather loosely, but there is one necessary condition for all types of RNG, which is that their output should be unstructured – the numbers should be statistically uniform and independent – to ensure that all possible output sequences occur with equal probability. For example, in a sequence of random decimal numbers, for uniformity each digit should appear with probability 1/10, each pair of digits (digraph) with probability 1/100, etc., and for independence, each digit should appear with probability 1/10 immediately after any given digit, etc. The decimal representation of π (and other so-called normal numbers) is a uniform and independent sequence [28], although there is nothing unpredictable about the value of any particular digit. Random sequences can be formulated in any number base; base 2 (binary) is typical for cryptography.

Human beings are notoriously poor at assessing the uniformity and independence of candidate random sequences – longer "runs" of consecutive 0s or 1s occur more often in random binary sequences than intuition suggests [29]. Fortunately, comprehensive statistical test suites [21,30,31,32,33] can target and root-out undesirable features from failures of uniformity or independence. For example, a binary sequence in which 0s occur more often than 1s (bias) would not be uniform but could still have the property of independence (often referred to as *"i.i.d."*, independent identically distributed):

A biased sequence of independent bits ...

0	1	0	0	1	0	0	0	0	1

Conversely, a sequence in which 0s and 1s alternate would be unbiased but not independent:

An unbiased but correlated sequence ...

1	0	1	0	1	0	1	0	1	0

But statistical tests are not a "silver bullet" - passing these tests is necessary, but as we will see is not sufficient to make an RNG suitable for cryptography.

ENTROPY → RANDOMNESS

RANDOM ↛ ENTROPY

A variety of computer algorithms called pseudo-random number generators (PRNG) can produce sequences of numbers that pass comprehensive statistical randomness testing for uniformity and independence. Examples include: simple linear feedback shift register (LFSR) based PRNGs [34]; linear congruential generator (LCG) PRNGs [35]; and Mersenne Twister PRNGs [36]. With some care in selection these PRNGs are convenient for numerical simulation applications. However, they are unsuitable for cryptography because their outputs are predictable and reproducible: each number output by these PRNGs is unambiguously determined by an algorithm. Although there are PRNG parameters, such as the starting position in the output sequence that need not be public, the entire past and future output of the PRNG can be easily determined by an adversary from only a few output values. Only two consecutive outputs of an LCG PRNG [37], and only 624 of a Mersenne Twister PRNG in its native form are needed to diagnose their parameters [38].

Cryptographically secure pseudo-random number generators (CSPRNG), also referred to as deterministic random number generators (DRNG), such as the ANSI X9.31 DRNG [39], improve on standard PRNGs by introducing some unpredictability and irreproducibility through the seed parameter in their algorithms. Typically, a seed will be a relatively short random sequence derived from a physical noise source, or pool of noise sources. Different RNG processes access different seed values, and the seeds themselves should be updated with fresh noise subject to certain criteria. Also, the DRNG output usually includes a cryptographic hash operation. In this way, DRNGs can provide long output sequences, much longer than their seeds, that are uniform and independent, and sufficiently unpredictable – both forward and backward – and irreproducible for cryptographic uses. This makes them valuable ingredients of a secure system, provided their seed values cannot be even partially known or influenced by an adversary, that they are correctly implemented, and continue to operate as designed. Their seed values must be supplied by a TRNG.

In contrast, TRNGs do not rely on algorithmic processes to generate random numbers but instead digitize analogue, physical noise sources and process the results into uniform and independent random number outputs. With our natural environment being so full of "noise" one might imagine that building a TRNG would be a simple, solved problem. But one person's noise could be another's (partially or completely) known signal, which makes it challenging to devise a TRNG with strong unpredictability assurances. And it is a fundamental result of complexity theory [40] that there can be no proof that a particular sequence of numbers is genuinely unpredictable. Instead, one must follow a process of design, analysis, testing and verification. When assessing the 'quality' of TRNGs it is necessary to ask several key questions:

- How is the claimed unpredictability quantified ?
- How is the physical noise "conditioned" to produce uniform and independent random numbers ?
- How is the unpredictable component extracted from the digitized noise?
- What is the basis for trusting that the TRNG noise source cannot be even partially known or controlled by an adversary ?
- How can correct operation be verified ?
- What is the maximum rate at which the analogue source can be sampled, and therefore what digital bit rate can be provided?
- What are the effects of temperature and other non-malicious factors on the noise source?
- What variations in the noise source might arise from manufacturing tolerances or component degradation over time?

A TRNG architecture that addresses these issues is embodied in several national and international standards [41,42,43,44]. (See Figure 1.). The output of a TRNG can be used directly by cryptographic applications or to seed a DRNG in situations where a data rate is required that is higher than the TRNG can produce [45]. In the ideal world all DRNGs would be seeded by a TRNG at their output rate.

3. ENTROPY, UNPREDICTABILITY AND RANDOMNESS

"The enemy knows the system." (C. E. Shannon, 1949 [46])

To illustrate the issues and challenges in devising a TRNG we will consider coin flipping as a hypothetical source of entropy. Although widely-regarded as the impartial method for making choices, we will see that even this benchmark of fairness would need to be used carefully in the adversarial world of cryptography. The dynamics of each flip of a physical coin is completely deterministic in principle – governed by Newton's Laws of Motion – but in practice, lack of sufficiently precise knowledge of the initial conditions (position of the coin and the applied impulse), and the limitations of human fine motor control makes the outcome unpredictable [47]. It has been proved that if each flip imparts a rotation of the coin about an axis that is precisely horizontal, then the outcome will be unbiased [48]: the probability, p_H, of a heads (H) outcome or the probability $p_T = (1 - p_H)$ of a tails (T) outcome will each be 0.5. Intuitively, successive flips of a coin are independent events, and so we may propose a sequence of N coin flips as an unpredictably random N-bit binary sequence,

$$b_{N-1}, b_{N-2}, \ldots b_2, b_1, b_0 ,$$

using the algorithm heads (H) = 0, tails (T) = 1: each of the 2^N possible N-bit sequences is equally likely. We might then contemplate using coin flips to generate a 256-bit AES key. But if the flip does not impart a rotation to the coin that is precisely about a horizontal axis, the outcome will be biased: $p_H \neq p_T$. Remarkably, this is a direct consequence of the deterministic laws of motion governing the coin flip [49], and will be present even if the coin's mass distribution is perfectly uniform. (This is quite distinct from statistical fluctuations in the balance between zeroes and ones: even in an unbiased sequence, there will be fluctuations of $\sim N^{1/2}$ around the expected number ($N/2$) of zeroes or ones between different N-bit sub-sequences.)

There are two distinct problems introduced by bias: the bit sequences produced by coin flipping are not uniformly distributed; and they have less than maximal unpredictability. The first of these problems can be corrected with a procedure known in general as conditioning (or whitening) to

produce uniform and independent bit sequences. For example, in the above example applying the SHA256 cryptographic hash function to each biased 256-bit sequence will produce uniform and independent 256-bit binary sequences that pass comprehensive statistical tests. However, this would only mask, not remove the less-than-maximal unpredictability, and so conditioning with cryptographic hash functions must be used with great care.

PASSING RANDOMNESS TESTS ≠ GUARANTEE OF ENTROPY
HASHED LOW-ENTROPY DATA (EVEN A STRING OF ALL 1S) WILL PASS RANDOMNESS TESTS
STATISTICAL RANDOMNESS TESTING MUST BE DONE BEFORE CRYPTOGRAPHIC HASHING

We must assume that an adversary could know about the presence of a bias, even if they did not know the result (H or T) of each flip: each 256-bit sequence is "less unpredictable" than in the unbiased case. For example, if the adversary knows that $p_H > p_T$, then a sequence of 256 zeroes is more likely than any of the other $(2^{256} - 1)$ possible outcomes. Nevertheless, even with bias a coin flip sequence is still somewhat unpredictable. So in addition to conditioning we need a quantitative measure of the "unpredictability", and an algorithm for extracting it to make a fully unpredictable sequence.

Shannon introduced the information theory concept of entropy [50] to quantify unpredictability. For a source that produces N-bit binary sequences, x, with probability P_x, Shannon's entropy is

$$H = -\sum_x P_x \log_2 P_x \ ,$$

measured in bits. For example, if a source outputs 256-bit sequences, of which 128 physical bits of each output are known and the remainder must be guessed, the (Shannon) entropy would be $H = 128$ bits. There is an entire family of entropy functions that characterize unpredictability in different circumstances [51,52]. Shannon's entropy is relevant for communications, but a more relevant quantity for cryptography is the min-entropy,

$$H_\infty = -\log_2 P_{max} \ ,$$

where P_{max} is the probability of the most likely output of the source. The min-entropy characterizes the probability that an adversary could correctly guess the output in a single trial, and is related to the notion of cryptographic work [53], which is the number of brute force trials required for an adversary to achieve a probability of 0.5 of determining the output. To see the difference

between Shannon entropy and min-entropy, consider a source producing N-bit sequences, with 50% probability for the "all 1s" output, and the balance of the probability equally distributed over the other outputs. For large N, the adversary's Shannon entropy in this case is approximately $N/2$ bits, whereas $H_\infty = 1$ bit. Clearly this would not be an acceptable source for cryptographic purposes. In general, H_∞ is the most conservative (smallest) measure of unpredictability, but all entropy measures are equal for uniform, *i.i.d.*, completely unpredictable binary sequences.

Returning to our example of 256 coin flips, if the bias was only 1% ($p_H = 0.51$) for instance, the min-entropy of the 256-bit sequence would be $H_\infty = 249$ bits (whereas the Shannon entropy is 255.9 bits). This means that if this sequence was used as a key an adversary would have to do two orders of magnitude less computational work (because $2^7 = 128$) to discover it by brute force than if it had full entropy ($H_\infty = 256$ bits). If the key was expected to provide 100 years of security, say, the actual security level would instead only be one year [54]. Because of this exponential dependence of brute force computational work on the number of bits of entropy in the key, an adversary should not be able to obtain even partial knowledge (a few bits) of a key.

X BITS OF ENTROPY → 2^X POSSIBLE STATES
7 BITS LESS ENTROPY → $1/2^7$ OR 1/128 LESS COMPUTATIONAL WORK FOR AN ADVERSARY

When we have a partially unpredictable N-bit binary output, with min-entropy $H_\infty = M < N$, we need an algorithm that will produce an M-bit uniform, *i.i.d.* sequence from it that has maximal entropy: one bit of entropy per bit. An algorithm with this property is called a randomness extractor.

For our example of independent but biased coin flips, von Neumann invented [25] a very simple streaming randomness extractor that works automatically: it is not necessary to know the actual value of the entropy. The sequence is divided into pairs of bits, $[b_{2i+1}, b_{2i}]$ for $i = 0, 1, 2 \ldots$, and each pair is replaced by the single bit, b_{2i}, if $b_{2i+1} \oplus b_{2i} = 1$*, i.e. if the pair is either $[0, 1]$ or $[1, 0]$, and the second bit, b_{2i+1}, is discarded. Both bits are discarded if $b_{2i+1} \oplus b_{2i} = 0$. This produces a shorter unbiased sequence with a yield of $p(1-p)$ bits per initial bit, where p is the probability of a 0. Subsequently, Peres [55] showed that von Neumann's procedure could be iterated on the discarded bits to produce an unbiased bit sequence with a yield asymptotically close to the

Here "$\oplus$" denotes exclusive-OR ("XOR"), binary addition modulo 2.

entropy bound: this algorithm accomplishes both entropy extraction and conditioning.

In practice, von Neumann's algorithm must be used with care. It should not be applied to bit sequences that are not independent, for instance. Doing so can introduce additional problems [56]. However, von Neumann's algorithm has been generalized to provide conditioning and randomness extraction for biased, correlated sequences [57]. We note that hashing [58] is a powerful operation for extracting maximal-entropy bits from partially-unpredictable ones [59]. Randomness extraction can also be conveniently performed with cryptographic hash functions [60].

This hypothetical example of coin flips illustrates the general architecture for cryptographic TRNGs that is embodied in multiple standards (see Figure 1):

VON NEUMANN'S ALGORITHM

BIASED **UNBIASED**

11 01 10 00 10 11 11 01 11 00 01 10 01 11 10 → 1 0 0 1 1 0 1 0

- a front end, comprising a digitized physical noise source producing internal raw bits;
- a mathematical model for the entropy of the internal bits based on the physical properties of the noise source and the digitization electronics;
- a back end, comprising algorithms for conditioning the internal bits to produce a uniform i.i.d. sequence and a randomness extractor to produce a maximal entropy output sequence (external bits)

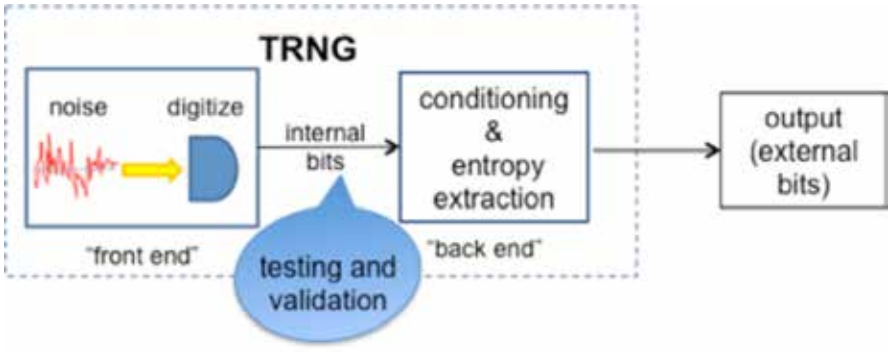

Figure 1: Standardized TRNG architecture. See text for details.

To see this, consider applying von Neumann's algorithm to the unbiased but correlated sequence ...0101010101... , which would result in the biased and correlated output ...11111...

4. NOISE SOURCES USED IN TRNG PRODUCTS AND THEIR SUITABILITY FOR DATA CENTER/CLOUD APPLICATIONS

TRNG products utilize a variety of noise sources. A few principal attributes characterize their suitability for data center/cloud applications, including: speed (output bit rate), quality (the level of trust in the amount of output entropy), and assurance (compatibility with TRNG architectural standards; availability of design information; and whether the hardware is open to verification both internally and externally). (See Table 1.)

The simplest TRNG front end captures entropy from observing human activity associated with an end-user either consciously or subconsciously doing something "randomly", such as typing or mouse movements. But human beings are notoriously poor noise sources: "random" typing tends to favor certain sections of a QWERTY keyboard, for example; and subjects who were asked to perform 300 coin flips reportedly produced overly-correlated binary sequences [61].§ The other major drawback of this approach is that a human user is not always present. In this case software systems can attempt to capture entropy from processes that occur in the host computer. Although these processes, such as network events, execution timing and the movements of the magnetic sensor in a hard-disk drive, are far from random, they possess jitter and suffer collision events that do introduce unpredictability. The challenge of this approach is that information about system-level activity is available to malicious as well as legitimate users.

A better approach is to craft a TRNG front-end that captures sources of noise that are independent of both the user and the host computer. For example, atmospheric radio noise resulting from distant lightning strikes, averaging 40 strikes per second globally, [62,63] is used as a TRNG noise source. This macroscopic source has the attractive features of being easily accessible with only a simple radio receiver, and random numbers can be provided to end devices over the Internet. But there are questions about the amount of entropy they can deliver. The radio spectrum, particularly in urban environments, is ever-more polluted with a background of man-made emissions (radio stations, cell phones, TV stations, Wi-Fi, etc.) that can be several orders of magnitude more powerful than the natural, unpredictable sources [64,65]. The very low entropy signal to background ratio makes it challenging to fully eliminate the man-made background from

‡ We do not discuss TRNG approaches that have only been demonstrated in a research setting, but not productized. There are many QRNG concepts in the scientific literature, often with very high reported bit rates. But these are typically at low Technology Readiness Levels, often utilize exotic components, and perform the back-end processing, if any, offline: the reported bit rates were usually not achieved as sustained throughput.

§ And worse, magicians and gamblers are reportedly able to flip coins with the appearance of fairness, but with a completely predictable outcome [49].

the output of atmospheric radio TRNGs, limiting the bit rate and the quality (trust) of the entropy. Any man-made component of the noise could be known to an adversary. Also, it is essential to have a bound on how much information an adversary might be able to obtain on the entropy signal itself. There are components of radio noise from the Sun, stars and distant galaxies [66], and the physics of these radio emissions leads to correlations between the observations of two different radio antennas pointed at the same source when their separation is less than that set by their resolution of the angular size of the emission region. This fact is used by radio astronomers to determine the size of stars [67], and has even been proposed as a means for establishing shared secret key material between two observers [68]. If an adversary could position their own antenna close to the TRNG's, they could obtain partial information about the bits produced. Also, because the source is external to the TRNG, it is impossible to know how much information an adversary may have about the bit sequences - an adversary could broadcast radio signals driven by a PRNG that could force known bits into the TRNG output. There is no diagnostic that could distinguish the spoofed signal from the desired one. (See "Atmospheric radio noise" entry in Table 1.) For cryptographic use it is preferable to have a noise source that is within the security perimeter of the TRNG.

Certain physical quantities, such as the position and velocity of each molecule of the air in a room, are for practical purposes unpredictable: although the time evolution of the position and velocity of each molecule is governed by simple deterministic laws of Nature – Newton's Laws of Motion – it is impractical to determine the position and velocity of each one, or microstate, at some initial time. The observable properties or macrostate – volume, pressure and temperature – are consistent with many microstates. In the 19th Century physicists discovered that this unpredictability of the microstates is described by a macroscopic physical quantity that was given the name "entropy", pre-dating Shannon's use of the same word in 20th century information theory. This thermodynamic entropy, denoted by the symbol S, has the dimensions of energy and the value

$$S = k \ln \Omega \quad ,$$

where k is a fundamental physical constant (Boltzmann's constant), and Ω is the number of microstates consistent with the macrostate of the system. Interpreting Ω^{-1} as the probability for the system to be found in any one microstate makes the connection with information-theoretic entropy possible [69, 70]; the equilibrium macrostate has the largest value of Ω, and is the "most random".

A direct manifestation of physical entropy is unpredictable fluctuations in macroscopic quantities. A time series of "snapshots" of a macrostate, taken by digitizing some macroscopic quantity, will reveal an unpredictable sequence of microstates. For a system of N particles, there will be snapshot-to-snapshot fluctuations in macroscopic quantities scaling in size like $\sim N^{1/2,}$ just as we saw that there will be $\sim N^{1/2}$ statistical fluctuations in the number of heads in sequences of N coin flips. Fluctuations in macroscopic quantities are used as the front-end noise source for several TRNGs.

Table 1. TRNG suitability for data center/cloud applications

TRNG type	Speed (entropy rate)	Quality (entropy trust and internal bit quality)	Assurance		
			Standards compatibility	Open design	Verifiable internal bits
Atmospheric radio noise					
Voltage fluctuations					
Clock jitter					
Logic gate metastability					
Radioactive decay					
Single photon on a beamsplitter					
Many photons on a beamsplitter					
Whitewood's Entropy Engine					

"low"

"medium"

"high"

"not accessible to user"

Table 1. A summary of the characteristics of several classes of TRNG types that use different entropy sources, relative to the needs of the data center/cloud environment. "Speed", "quality" and "assurance" capture, respectively: the output bit rate; the trust that can be placed in its entropy content; and the compatibility with accepted TRNG standards, availability of design details, and the ability to test and verify the internal bits. See text for details.

A number of TRNG products utilize the thermal voltage fluctuations (Johnson noise) in a resistor [71]. Even when no external current flows, there will be fluctuations resulting in an imbalance between the numbers of electrons in each end of a resistor, and so producing an externally measurable voltage. Another noise source, called shot noise, is present when external current flows: because electric current is carried by electrons, each with a discrete unit of charge, statistical fluctuations in the number of electrons will result in voltage fluctuations. One of the first examples of an RNG of this type was ERNIE (Electronic Random Number Indicating Equipment), used to pick winning Premium Bond numbers (a type of lottery) in the UK [72]. ERNIE is now on its fourth version, and other TRNGs capturing Johnson noise are available from small-scale manufacturers as USB stick devices. For example the thermal noise of a 50Ω resistor at room temperature into a 1 GHz bandwidth amplifier is approximately 20μV. But as with atmospheric radio noise, broadcasts from nearby radio and television stations, which fall into that bandwidth, can be picked up and will be larger than the thermal noise unless the electronics are very well shielded from external noise. These TRNGs are also susceptible to pick-up of power line voltage noise. Thus many of the same concerns about the amount and quality of entropy provided by atmospheric radio noise TRNGs also apply to localized electronic noise TRNGs. (See "Voltage fluctuations" entry in Table 1.)

The small noise signal to background problem is typical for macroscopic systems: there are relatively very few microstates corresponding to large fluctuations, which are therefore rarely encountered, and only very small fluctuations around equilibrium values predominate. The number of particles, N, in everyday objects is very large, typically $N \sim 10^{20} - 10^{23}$, and so the fractional size of the fluctuations, $\sim 1/N^{1/2}$, is correspondingly small. Some TRNGs seek to mitigate this by fabricating a macroscopically metastable system, triggered by microscopic fluctuations. Examples are: clock jitter in processors [73]; ring oscillator-based TRNGs in FPGAs [74]; and Intel's metastable logic gate TRNG - Bull Mountain [75,76]. But as with mechanical coin-flipping devices [48] they rely on the ability to reliably manufacture and reproduce macroscopically metastable systems. These man-made tools and processes are unavoidably susceptible to manufacturing variability and unrecognized manufacturing flaws [77] or worse, intentional interference in the manufacturing process [16] to force some known or set of known initial conditions. (A mechanical coin-flipping device with a completely predictable output has been demonstrated [49], for instance.) As we have seen earlier, even a small reduction in min-entropy can drastically reduce the cryptographic work an adversary must perform. Recognizing the risk of manufacturing variance - almost all manufacturing activity is outsourced - it becomes important to validate each manufactured component. This raises issues with the use of embedded RNGs. Incorporating the TRNG as part of the processor chip makes for excellent ease-of-use, but user inspection and verification of the internal bits is not possible. This difficulty in both trust and verification of the entropy has led to considerable controversy about use of this type of TRNG [78]. (See "Clock jitter" and "Logic gate instability" entries in Table 1.)

Macroscopic instability TRNGs rely for their security on technological assumptions about an adversary's ability to know the initial conditions. To address these problems, sometimes multiple independent sources of entropy are pooled. If the bits from multiple entropy sources are XORed together, the entropy of the output is not less than the level of entropy present in the input with the greatest entropy. So if two sources are used, even if one is completely known to an adversary, the XOR output still has the entropy of the other input. Security is then based on the assumption that it would be infeasible for an adversary to compromise all of the entropy sources. However, this comes at the price of the added complexity of maintaining multiple sources and potentially introduces a sense of complacency - "at least one of my entropy sources is surely working…". When mixing entropy sources the lower-grade sources should only be considered as fail-safe sources – the strongest possible primary noise source with the highest entropy assurance and availability is always preferable and should be classified and treated as such.

TRNGs based on radioactive decay utilize the quantum-mechanically unpredictable emission time of individual decay particles from an atomic nucleus [79]. They record the time each decay particle is detected and turn that into a random number by using the time between detections, or the time

of detections during a given time period. Compared to electronic noise, radioactive decay has a much higher signal (energy per digitized sample) and a much lower background. Unfortunately, while the resolution of the timing may be quite high and thus many bits harvested from each detection, it is difficult to produce random bits at a very high rate because a high-speed source is also a very radioactive one. However, this does suggest looking for a TRNG noise source that has fluctuations directly rooted in the quantum physics of individual elementary particle behavior and which operates in a part of the electromagnetic spectrum that has low background that can be readily shielded. The optical regime is clearly indicated. (See "Radioactive decay" entry in Table 1.)

5. QUANTUM RANDOMNESS

'A philosopher once said, "It is necessary for the very existence of science that the same conditions always produce the same results." Well, they don't.'
(R. P. Feynman, 1965 [80])

Quantum theory, which describes the physics of atoms, photons and other elementary particles, has provided the most precise and accurately tested predictions of any physical theory. However, some quantities cannot be known except as probabilities, even in principle. For example, consider a beam of light incident on a half-silvered mirror, or 50-50 beamsplitter, designed to transmit 50% of the optical power and reflect 50%. We know that the transmitted and reflected beams have the same wavelength (color) as the incident beam. But now imagine reducing the incident optical power until individual photons arrive, one at a time, at the beamsplitter. Each photon has a definite energy, hc/λ, where h is a fundamental constant of Nature (Planck's constant), c is the speed of light, and λ is the wavelength of the light. How, then, do photons behave at the beamsplitter? A photon cannot split into two, because energy conservation would then require the reflected and transmitted photons to have one half of the incident photon's energy, and hence longer wavelengths, in contradiction with observation. Instead, in quantum physics, with probability 50% each incident photon is reflected, and with probability 50% transmitted. For any given incident photon no further specification of whether it will be reflected or transmitted is possible, even in principle. This irreducible quantum randomness [23] is strikingly different from the randomness in classical physics: no one, no matter what technology they may possess, money, time and effort they expend can predict individual photon behavior at a beamsplitter.

Einstein was deeply dissatisfied with this aspect of quantum physics, and proposed that photons might have additional, individual labels [81] that would precisely specify their behavior at a beamsplitter. However, physicist John Bell showed [82] that this notion of "hidden variables", beyond quantum physics, would have experimentally measurable consequences. Those

experiments have now been performed many times at higher and higher levels of precision [83]. The results show that Einstein was wrong: photons with the same energy, direction of travel and state of polarization are indistinguishable elementary particles, as predicted by quantum theory. This means that a photon stream, generated at optical wavelengths by an attenuated mass-produced light source of the type used, for example, for telecommunications, and impinging on a beamsplitter could constitute a quantum TRNG – a QRNG – with unprecedented trust and verifiability. A single-photon detector (SPD) would be placed in the reflected output port, and another in the transmitted output port. Then, registration of a reflected photon detection would be digitized as a "0", and registration of a transmitted photon detection as a "1", to produce a binary stream [84,85] capturing the irreducible quantum randomness. The energy per digitized sample ($\hbar c/\lambda$) would be more than an order of magnitude larger than for electronic noise TRNGs (fluctuation energy per sample ~ kT, where T is the absolute temperature), and the background from detector dark counts and after pulsing would be much lower, giving a superior signal to background ratio. Imperfections in the 50-50 split ratio of the beamsplitter or unequal detection efficiencies would lead to biases, but these could be removed and the entropy extracted with von Neumann's algorithm as discussed above.

However, SPDs at optical wavelengths are expensive components, which require specialized operating electronics and power supplies, and are typically limited to count rates ~ 1 MHz in their more affordable forms. (Higher count rates are possible in research-grade SPDs that require cooling to near absolute zero temperature.) This limits the output bit rate of this type of QRNG to only a few Mbps, which is insufficient for data center/cloud purposes [4]. Also, the optical power levels involved (~ 10^6 photons per second) are many orders of magnitude smaller than typical optical power levels encountered in telecommunications, for example. (See "Single photon on a beamsplitter" entry in Table 1.) So although the optical regime does not suffer from the difficult-to-shield, high levels of environmental background at radio wavelengths, the complexity of single-photon detection means that other, easier to use quantum-mechanically random sources should be sought.

Other QRNGs seek to avoid these rate limitations and background sensitivity by shining a brighter light source onto the beamsplitter [86]. The quantum property of the light will then show up as random fluctuations between the optical power in each output port. But putting more photons on a beamsplitter results in smaller relative fluctuations: the fluctuation size is the square root of the number of photons in each measurement (optical shot noise). If one photon is measured at a time the fluctuation level is $\sqrt{1}/1$ or 100%, but if 200 photons are put onto the beamsplitter the fluctuations at each output averages $\sqrt{100}/100$ or 10%. When enough photons are available so that fast photo-detectors can be used (~ 10^8 per sample) the fluctuations are typically less than $\sqrt{(10^8)}/10^8$ or 0.01%, and would have the same problem of detecting a small noise signal against

** Or equivalent sets of three labels.

a large background of TRNGs that utilize electronic shot noise. (See "Many photons on a beamsplitter" entry in Table 1.) These difficulties are overcome in Whitewood's Entropy Engine, which is based on a robust quantum phenomenon that is manifest at much higher optical signal levels, which are readily accessible with mass-produced components developed for modern optical telecommunications systems.

6. WHITEWOOD'S ENTROPY ENGINE

The Entropy Engine uses a macroscopic manifestation of quantum randomness that is directly traceable to the indistinguishability of photons, which we can illustrate with the following analogy. Consider a macrostate of one particle and two boxes; there are two possible microstates: particle in left box, right box empty; and left box empty, particle in right box. If both microstates are equally probable the particle is found in either box with probability ½, whether it is a classical or quantum particle. Now consider a macrostate of two particles and two boxes. (See Figures 2a and 2b.) If the particles obey the laws of everyday, macroscopic physics, they are distinguishable and there are four possible microstates: both in the left box; both in the right box; and two in which each box contains one particle. If all microstates are equally probable, the probability that both particles are found in the same box is 1/2. However, if the particles are quantum, they are indistinguishable, and there is only one distinct microstate with one particle in each box. Then, if all microstates are equally probable the probability that both quantum particles are found in the same box is 2/3, and the fluctuations in particle number are much larger than with classical, distinguishable particles. It is as if indistinguishable particles "like" to bunch together in boxes that are already occupied, leading to large quantum-enhanced fluctuations, which can be very much larger than the statistical fluctuations of classical physics. We can also see that this non-classical behavior only becomes apparent when the occupation number (average number of particles per box) is comparable to or larger than 1. And the fluctuations in the number of particles per box increases in proportion to the occupation number, whereas for classical particles, or shot noise, these fluctuations would only grow as the square root of the occupation number. Translating these notions to photons, each box represents a distinct photon "mode" – the complete set of labels (energy, direction and polarization) allowed by quantum physics – and the mode occupation behavior is called Bose-Einstein statistics.

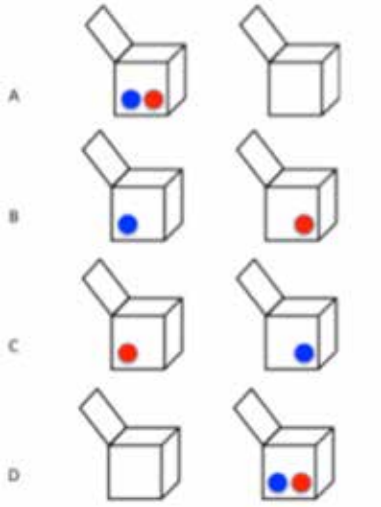

Figure 2a: Possible microstates of two distinguishable particles in two boxes.

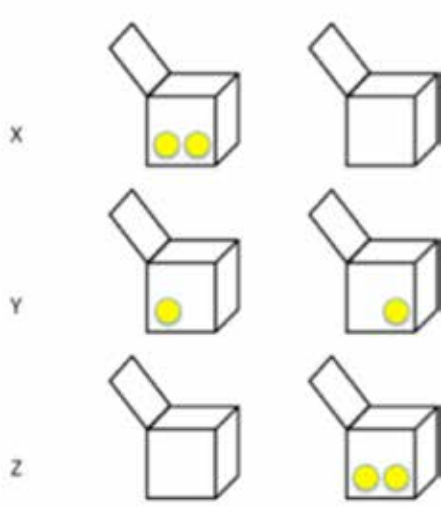

Figure 2b: Possible microstates of two indistinguishable (quantum) particles in two boxes.

This quantum-enhanced fluctuation behavior of photons is not readily apparent in everyday light sources because although they usually involve a large number of photons, many more modes are possible than are populated: the occupation number is well below 1. However, in the Whitewood Entropy Engine QRNG the occupation number greatly exceeds 1 because the number of modes is restricted, and large quantum-enhanced fluctuations are present at optical power levels typical of telecommunications, to give a macroscopic signal of quantum randomness with a very robust signal (a fluctuation energy per digitized sample two orders of magnitude larger than for many-photon beamsplitter QRNGs). SPDs are not needed, and high-bandwidth, low-noise optical detectors developed for telecommunications can be used to make a QRNG with bit rates of several 100MHz or much higher, the only limiting factor is the performance (and therefore price) of the electronics used to capture the analogue randomness. ‡

The Entropy Engine's front-end quantum noise source contains a photonics module in which a lens focuses photons from a carefully designed optical source onto a high-speed detector. The detector provides a high signal-level output in which the quantum fluctuation component is an order of magnitude larger than electronic noise. The detector output is amplified to a voltage level convenient for digitization, and a binary sequence is generated by a 1-bit comparator at the rate of 400 MHz. Each bit corresponds to the detection of ~10^8 photons.

Because the quantum and electronic noise components behave differently depending on the

‡ One prototype has already been operated at a sustained 6Gbps streaming output rate, and another has demonstrated the feasibility of 44Gbps output. Even higher rates are feasible.

Entropy Engine's front-end optical power, we can determine how much quantum entropy is present in the detected fluctuations by varying the output of the optical source. We conservatively assume that the non-quantum, electronic, noise component could be completely known by an adversary.

Real-world considerations require some conditioning of the internal bits. The bandwidth of the optical source is in the THz range while the detector has a bandwidth of >10 GHz. However, in line with the digitization rate, the amplifier has a bandwidth of 600 MHz. Although this bandwidth is adequate for the digitization rate of the comparator, no electronic circuit can respond perfectly to the electronic signal coming from the output of the optical source. Effects such as circuit damping and finite slew rate cause the signal input to the digitizer to not be completely independent from one step to the next. The number of steps over which a correlation is present is easily determined by examining the autocorrelation of the bit stream. Correlations are removed by duplicating the bit stream, delaying one copy by an amount greater than the time determined to be necessary by the autocorrelation and then XORing together the two data streams [87,88] to produce the Entropy Engine's internal bits.

The internal bit stream passes comprehensive statistical randomness tests, providing evidence that there are no entropy-reducing features beyond the short-range correlations removed by the very minimal "shift-and-XOR" conditioning [42]. Simple state-of-health randomness testing of internal bits is done continuously to verify correct operation, and the internal bits can be made available for verification of the correct operation of the optical and analogue components of the Entropy Engine. Compatibility with existing security standards requires further processing to turn the internal bits into the Entropy Engine's output of external bits [41]. This is also because an output with full quantum entropy is desired. Calculations show that the quantum entropy is typically 0.99 bits per internal bit. The non-quantum (electronic) noise components are removed by the random extractor in the back end – we throw away as untrustworthy the electronic noise that many previous TRNGs have used as their noise signal.

The Entropy Engine's 200Mbps output of processed external bits is formed by concatenating the 512-bit output blocks produced by applying the SHA512 cryptographic hash algorithm to 1,024-bit blocks of the internal bit stream. This (very conservatively) eliminates the classical entropy component of the internal bits. It also provides compatibility with NIST standards that require that TRNG outputs be processed through such a cryptographic randomizing function. This protects the system from producing a completely predictable output should there be an undetected reduction of the entropy in the source, for example, due to component failure. (See "Entropy Engine" entry in Table 1.)

7. SUMMARY AND CONCLUSIONS

In this white paper we have seen that cryptographic entropy is an essential security resource that is all too often in short supply, and that current trends are making ever-greater demands, especially in the data center/cloud environment. Previous TRNG products, including previous QRNGs, fall short in one or more of: entropy rate; entropy quality; standards compatibility; design openness; or hardware verifiability. What is needed is an entropy source with: high rate; a large random signal to background ratio to reduce susceptibility to environmental influences; an unpredictability guarantee; and an assurance of verifiable operation. Whitewood's Entropy Engine QRNG satisfies all of these requirements, providing a 200Mbps full quantum entropy output traceable to the indistinguishability of elementary particles. The Entropy Engine can provide the high-assurance randomness foundation that is necessary for the security of the data center/cloud environment today and into the future.

RICHARD HUGHES

Richard Hughes is a Consulting Physicist and a Senior Advisor to Whitewood Encryption Systems, Inc. of Boston, MA. Richard retired from a three-decade career at Los Alamos National Laboratory (LANL) in 2014, where he held the position of Laboratory Fellow in the Physics Division. Richard founded and for two decades was co-lead of the Quantum Communications team at LANL. He was co-principal investigator of multiple research projects until his retirement. Richard received his B.Sc. (Hons., Class I) degree in Mathematical Physics from the University of Liverpool, England, and his Ph.D. in Theoretical Elementary Particle Physics from the same institution. He held research positions at: Oxford University and The Queen's College, Oxford, England; the California Institute of Technology, Pasadena, California; CERN, Geneva, Switzerland; and the University of Edinburgh, Scotland; before joining LANL as a Technical Staff Member. Richard has held distinguished visiting scientist positions at the University of Oslo, Norway, and at Oxford University (Dr. Lee Fellow, Christ Church). In 1996, 1998, 2006, 2010 and 2012 he was awarded Los Alamos Distinguished Performance Awards for his quantum cryptography research, and in 1997 he was awarded the Los Alamos Fellows' Prize for his research on quantum information science. Richard is a Fellow of the American Physical Society. In 2001 he was co-winner of an R&D100 Award for Free-space quantum cryptography. Starting in 2001, Richard led the US Government's Quantum Information Science and Technology Roadmap. In 2004 Richard and the LANL Quantum Communications Team were co-winners of the European Union's Descartes Prize. He has acted in an advisory role on multiple occasions for several US Government agencies, and in 2008 he received the ODNI Distinguished Analysis Award. Richard has given many invited presentations at major international scientific conferences and research universities. He has 31 US and foreign patents and patent applications in quantum communications, and he has authored over 160 scientific papers on elementary particle physics, quantum field theory, the foundations of quantum mechanics, quantum cryptography and quantum computation.

JANE E. (BETH) NORDHOLT

Jane E. (Beth) Nordholt is a retired Fellow of the Los Alamos National Laboratory and Senior Adviser to Whitewood Encryption Systems, Inc. After Beth and Richard Hughes invented the methodologies that make long-distance, free-space quantum cryptography (QC) possible, she became co-lead of the LANL QC team and, with Richard, began to develop its free-space and fiber-optic QC capabilities. She is also the inventor of satellite-based QC, which is being integrated with satellite optical communications and pursued in many countries. While at LANL Beth was an inventor on 31 patents, patent disclosures, and foreign patent applications related to QC and a patent on the mass spectrometer she developed for the NASA-ESA Cassini mission to Saturn. With U.S. Government personnel she wrote two reports on the security of QC. Beth earned six LANL Distinguished Performance Awards, five of which were related to QC and also received four LANL Awards Program prizes. Beth started the effort to increase fiber QC distances and security by using specialized detectors in collaboration with personnel from NIST Boulder. This effort resulted in several world-record distances for QC in optical fiber. She was the lead inventor of the Velocirandor quantum random number generator, now called the Entropy Engine, and QKarD, a device and architecture designed to make QC commercially viable. Before concentrating her research efforts at LANL on QC, Beth was principal investigator on the NASA Genesis Concentrator, the NASA Deep Space 1 (DS1) Plasma Experiment for Planetary Exploration (PEPE), and led the NASA-ESA Cassini Ion Mass Spectrometer invention and development as part of the CAPS instrument. She received NASA Group Achievement Awards for her work on the Polar/TIDE, Cassini/CAPS, and DS1/PEPE spacecraft, as well as the DS1 flyby of comet Borrelly. She has been a member or chair of several NASA proposal review panels and was co-Lead on the NASA New Millennium Project's Instrumentation team.

REFERENCES

1. R. R. Coveyou, "Random number generation is too important to be left to chance", Studies in Applied Mathematics 3, 70 (1970).
2. A. Kerckhoffs, "La Cryptographie Militaire", Journal des Sciences Militaires 9, 5 (1883).
3. J. Kelsey et al., "Cryptanalytic Attacks on Pseudorandom Number Generators", Lect. Notes Comp. Sci. 1372, 168 (1998).
4. "Understanding and Managing Entropy and Random Data"white paper, Whitewood Encryption Systems (August 2015).
5. Federal Information Processing Standards FIPS PUB 140-2: "Security Requirements for Cryptographic Modules" National Institute of Standards and Technology (2001)
6. A. K. Lenstra et al., "Ron was wrong, Whit is right", https://eprint.iacr.org/2012/064.pdf.
7. N. Heninger et al., "Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices", Proceedings of the 21st USENIX Security Symposium, August 2012.
8. J. Menn, "Exclusive: Secret contract tied NSA and security industry pioneer" (December 20, 2013).
9. S. Checkoway et al., "An update on the backdoor in Juniper's ScreenOS", http://cseweb.ucsd.edu/~hovav/dist/rwc16.pdf (2016).
10. D. Shumow and N. Ferguson, "On the Possibility of a Back Door in the NIST SP800-90 Dual EC PRNG" (21 August 2007); "NIST Removes Cryptography Algorithm from Random Number Generator Recommendations", National Institute of Standards and Technology (21 April 2014).
11. I. Goldberg and D. Wagner, "Randomness and Netscape Browser" Dr. Dobb's Journal (January 1996).
12. "DSA-1571-1 openssl -- predictable random number generator", Debian Security Advisory 13 May 2008.
13. T. Ristenpart and S. Yilek, "When Good Randomness Goes Bad: Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography", Proceedings of the Network and Distributed System Security Symposium - NDSS 2010, Internet Society, 2010.
14. M. Bendel, "Hackers Describe PS3 Security As Epic Fail, Gain Unrestricted Access" (December 29, 2010).
15. G. Argyros and A. Kiayias, "PRNG: Pwning Random Number Generators" BlackHat 2012.
16. D. Goodin, http://arstechnica.com/security/2013/09/researchers-can-slip-an-undetectable-trojan-into-intels-ivy-bridge-cpus/, Ars Technica (September 18, 2013).
17. See, for example, "RdRand", https://en.wikipedia.org/wiki/RdRand.
18. D. Goodin, "Crypto flaws in Blockchain Android app sent bitcoins to the wrong address", http://arstechnica.com/security/2015/05/crypto-flaws-in-blockchain-android-app-sent-bitcoins-to-the-wrong-address/ , Ars Technica (May 29, 2015).
19. W. Diffie et al., "Authentication and authenticated key exchanges", Designs, Codes and Cryptography 2, 107 (1992).

20. V. Bernat, "SSL/TLS and Perfect Forward Secrecy", http://vincent.bernat.im/en/blog/2011-ssl-perfect-forward-secrecy.html.
21. "Cryptography Today", National Security Agency, https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml (August 2015).
22. J. Buchmann et al., "Hash-based digital signature schemes," in "Post-Quantum Cryptography" D. J. Bernstein et al. eds. (Springer, New York, 2009).
23. G. J. Milburn, "The Feynman Processor" (Basic, Reading, 1999).
24. J. E. Nordholt et al., "Quantum Random Number Generators", US patent application 13/754,457 (2013).
25. J. von Neumann, "Various techniques used in connection with random digits", Appl. Math. Ser., Notes by G. E. Forstyle, Nat. Bur. Stand., 12, 36, 1951.
26. F. Galton, "Dice for Statistical Experiments", Nature 42, 13 (1890).
27. See, for example, D. J. Bennett, "Randomness" (HUP, Cambridge, 1998).
28. R. Wicklin, "Analyzing the first 10 million digits of pi: Randomness within structure", http://blogs.sas.com/content/iml/2015/03/12/digits-of-pi.html (March 12, 2015).
29. See, for example, I. Peterson, "The Jungles of Randomness: a Mathematical Safari" (Wiley, New York, 1998).
30. "A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications," NIST SP800-22 National Institute of Standards and Technology (2001).
31. P. L'Ecuyer and R. Simard, "TestU01: A C Library for Empirical Testing of Random Number Generators" ACM Transactions on Mathematical Software, 33, article 22, 2007.
32. "NIST SP800-90B Entropy Assessment tool", https://github.com/usnistgov/SP800-90B_EntropyAssessment.
33. G. Marsaglia, "DIEHARD", http://stat.fsu.edu/pub/diehard/ (1995).
34. See, for example D. Welsh, "Codes and Cryptography" (OUP, Oxford, 1988) pp. 126-131.
35. See, for example, D. E. Knuth, "The Art of Computer Programming: Seminumerical Algorithms" (2nd ed.), (Addison-Wesley, Reading, 1981).
36. M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator", ACM Transactions on Modeling and Computer Simulation 8 (1), 3 (1998).
37. J. Reeds "Cracking a Random Number Generator", Cryptologia 1, 20 (1977).
38. See, for example, http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/efaq.html.
39. Accredited Standards Committee X9F1. Draft American National Standard X9.82 (Random Number Generation), Part 2, Entropy Sources, Jun. 2005; S. S. Keller, "NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES", NIST (2005).
40. G. Chaitin, "Information-Theoretic Computational Complexity", IEEE Trans. Inf. Th. IT-20, 10 (1974); see, for example, E. Beltrami, "What is Random?" (Springer, New York, 1999) for an overview.
41. "Recommendation for the Entropy Sources Used for Random Bit Generation", National Institute of Standards and Technology draft Special Publication 800-90B (2013).

42. W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators" (BSI AIS 31, Bonn, Germany, 2011).
43. ISO/IEC 18031:2011-11 (E): Information technology -- Security techniques -- Random bit generation.
44. D. Eastlake et al., RFC 4086 "Randomness Requirements for Security" (IETF, 2005): http://www.rfc-editor.org/rfc/rfc4086.txt
45. "Recommendation for Random Bit Generator (RBG) Constructions", National Institute of Standards and Technology draft Special Publication 800-90C (2013).
46. C. E. Shannon, "Communication Theory of Secrecy Systems", Bell Sys. Tech. J. 28, 656 (1949).
47. J. Ford, "How Random is a Coin Toss", Physics Today April 1983, 40; V. C. Vulovic and R. E. Prange, "Randomness of a true coin toss", Phys. Rev. A 33, 576 (1986).
48. J. B. Keller, "The Probability of Heads", Am. Math. Mon. 93 (March 1986) 191.
49. P. Diaconis, S. Holmes and R. Montgomery, "Dynamical Bias in the Coin Toss", SIAM Review 49, no.2, 211 (2007).
50. C. E. Shannon, "A Mathematical Theory of Communication", Bell Sys. Tech. J. 27, 379 (1948).
51. A. Renyi, "Measures of Entropy and Information" (UCP, Berkeley, 1961).
52. C. Cachin, "Entropy Measures and Unconditional Security in Cryptography", Ph.D. Thesis, ETH-Zurich, 1997.
53. J. L. Massey, "Guessing and Entropy", Proc. IEEE Int. Symp. Inf. Th. 1994, p.204.
54. See, for example, http://www.keylength.com.
55. Y. Peres, "Iterating von Neumann's procedure for extracting random bits," Ann. Stat. 20, 590 (1992).
56. R. Solca, "Testing of a Quantum Random Number Generator", Master's Thesis, ETH-Zurich, 2010.
57. H. Zhao and J. Bruck, "Streaming algorithms for optimal generation of random bits", arXiv:1209.0730 (2012).
58. J. L. Carter and M. N. Wegman, "Universal classes of hash functions", J. Comp. Sys. Sci. 18, 143 (1979).
59. D. R. Stinson, "Universal hash families and the left-over hash lemma, and applications to cryptography and computing", J. Combin. Math. Combin. Comput. 42, 3 (2002).
60. Y. Dodis et al., "Randomness extraction and key derivation using the CBC, Cascade and HMAC modes," Lect. Notes. Comp. Sci. 3152, 494 (2004).
61. D. Mackenzie, "The Fifty-one percent solution", in What's Happening in the Mathematical Sciences, 7 (AMS, Providence, 2009).
62. A. D. Spaulding and J. S. Washburn, "Atmospheric Radio Noise: Worldwide Levels and Other Characteristics", NTIA Report 85-173 (1985).
63. "Handbook of Geophysics and Space Environment", A. S. Jursa ed., (National Technical Information Service, Springfield, VA, 1985).
64. M. A. McHenry et al., "Phone to Fridge: Shut Up!", IEEE Spectrum (September 2015); A. J. Wagstaff et al., "Man made noise measurement programme", Mass Consultants report (2003); J. A. Wepman and G. A. Sanders, "Wideband Man-Made Radio Noise Measurements in the VHF and Low UHF Bands", NTIA Technical Report TR-11-478 (2011).

65. C. Bianchi and A. Meloni, "Natural and man-made terrestrial electromagnetic noise: an outlook", Ann. Geophys. 50, 435 (2007).
66. See, for example, J. D. Kraus, "Radio Astronomy" (McGraw-Hill, New York, 1966).
67. R. Hanbury Brown and R. Q. Twiss, "Interferometry of the Intensity Fluctuations in Light", Proc. Roy. Soc. A 242, 300 (1957).
68. U. M. Maurer, "Secret Key Agreement by Public Discussion from Common Information," IEEE Trans. Inf. Theory 39, 733 (1993).
69. E. T. Jaynes, "Information Theory and Statistical Mechanics", Phys. Rev. 106, 620 (1957).
70. A. Katz, "Principle of Statistical Mechanics: the Information Theory Approach" (Freeman, San Francisco, 1967).
71. H. F. Murry, "A General Approach for Generating Natural Random Variables", IEEE Trans. Comp. (December 1970) 1210.
72. R. K. Hayward et al., "Computer Selects Premium Bond Winners", Electronics (July 1, 1957).
73. V. Fischer and M. Drutarovsky, "True Random Number Generator Embedded in Reconfigurable Hardware", LNCS 2523, 415 (2003).
74. B. Sunar et al., "A Provably Secure True Random Number Generator with Built-in Tolerance to Active Attacks", IEEE Transactions on Computers 56, 109 (2007).
75. M. Hamburg, P. Kocher and M. E. Marson, "Analysis of INTEL's Ivy Bridge Digital Random Number Generator", Cryptography Research, Inc., white paper (March 2011).
76. G. Taylor and G. Cox, "Digital Randomness", IEEE Spectrum (August 2011).
77. W. Schindler and W. Killmann, "Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications", LNCS 2523, 431 (2003).
78. R. Chirgwin, "FreeBSD abandoning hardware randomness", The Register, December 9, 2013; September 6, 2013 Google Plus posting by Theodore Ts'o.
79. M. P. Silverman et al., "Tests for randomness of spontaneous quantum decay", Phys. Rev. A, 61, 042106 (2000); R. Aguayo et al., "Throwing Nature's Dice" Am. J. Phys. 64, 752 (1996).
80. R. P. Feynman, "The Character of Physical Law", pg. 147 (MIT Press, Cambridge, 1965).
81. A. Einstein, B. Podolsky and N. Rosen, "Can the Quantum-Mechanical Description of Physical Reality be Considered Complete ?", Phys. Rev. 47, 777 (1935).
82. J. S. Bell, "On the Einstein Podolsky Rosen Paradox", Physics 1, 195 (1964).
83. B. Hensen et al., "Experimental loophole-free violation of a Bell inequality using entangled electron spins separated by 1.3 km", arXiv:1508.05949v1 (2015), and references therein.
84. A. Stefanov et al., "Optical quantum random number generator", J. Mod. Opt. 47, 595 (2000).
85. T. Jennewein et al., "A fast and compact quantum random number generator", Rev. Sci. Inst. 71, 1675 (2000).
86. See, for example, C. Gabriel et al., "A generator for unique quantum random numbers based on vacuum states", Nature Photonics 4, 711 (2010).

87. M. Santha and U. V. Vazirani, "Generating quasi-random sequences from semi-random sources," J. Comp. Sys. Sci. 33, 75 (1986).

88. U. V. Vazirani, 'Towards a Strong Communication Complexity Theory or Generating Quasi-random sequences from two communicating semi-random sources," 15th Annual ACM Symp. on Theory of Computing, pp 366-378, 1983.