Certifiable Randomness from Quantum Information

Matthew Yeh
Electrical Engineering and Computer Sciences,
University of California, Berkeley, CA 94720

(Dated: May 21, 2020)

Random number generators (RNGs) play an integral role in many scientific and commercial applications. While a myriad of pseudo-RNGs and hardware RNGs have been developed, true randomness is difficult to characterize. In recent years, there has been an explosion of research work on certifiable randomness, tied to the inherently random nature of quantum mechanics and quantum information. In this work, we assess three main routes toward certifiable randomness: 1) Einstein certification via Bell tests, 2) cryptographic certification via the hardness of learning with errors (LWE), and 3) certification from quantum supremacy, which is a very recent development. We suggest potential future directions and conclude that certified random number generators may well appear in the near future.

I. INTRODUCTION

The success and security of many applications today are contingent on having a high-quality source of random numbers [1]. For example, in both classical and quantum cryptography, random numbers are used to generate keys for data encryption. Randomness also plays a significant role in the sciences, from computational techniques such as Monte Carlo estimation to fundamental tests of quantum mechanics. Finally, random numbers lend themselves naturally to numerous commercial applications such as online gambling. It is unsurprising then that random number generators (RNGs) have provided a steady source of research interest for many years. A natural question then arises: What actually makes a good RNG?

To start, we should draw a clear distinction between true randomness and pseudorandomness. True randomness is akin to the theoretical construct of flipping a fair coin over and over—in other words, each outcome is equally likely, forming a uniform distribution. In contrast, pseudorandomness is deterministic in nature and thus can in principle be reverse-engineered by an adversarial attack. Specifically, pseudorandom number generators (PRNGs) start from a "seed," an input string of bits that fully determines the output bit sequence using some algorithm, often based off number theory [1]. While it is sufficient for most applications to only have the appearance of true randomness, it is important that the PRNG, while predictable, still follows the statistics of a uniform probability distribution and be free of correlations. In the early days of PRNG development this was not the case, with a particularly infamous example being IBM's RANDU generator: if used to produce points in 3D space, the points map to at most 2,344 planes [2]. Many scientific results from that time period that were based on Monte Carlo simulations are now questionable as a result. Since then, much work has been done to develop high-quality PRNGs. Notably, at present the most popular implementation of a PRNG, the Mersenne Twister, has a period of $2^{19337} - 1$ and thus one would be hard-pressed to find a pattern in any given sequence of numbers during a realistic operation time [3]. PRNGs also possess certain characteristics that make them quite attractive for most practical applications. Because they are classical algorithms that can be implemented in software, they are generally quite fast and produce bits with high throughput [1]. The predictability can additionally be a benefit when reproducibility is a desired feature, such as while testing and debugging systems.

On the other hand, hardware RNGs are often held as the "gold standard" for randomness. These devices are based off an unpredictable physical process, either classical or quantum. For example, classical randomness sources include white noise or clock drift in circuits, the timing of past events (e.g. the last time a disk operation took place), or user inputs (e.g. key-strokes or mouse motion) [1,4]. Quantum RNGS (QRNGs) work off the principle that quantum mechanics at its core is random, and in fact have a long history, with the earliest ones demonstrated around the mid-20th century by means of measured nuclear decays [5]. Briefly, Geiger-Mueller (GM) tubes are used to count β decays, i.e. emitted electrons, emitted from some radioactive source. Each decay event is independent from the others and thus the number of clicks in a given time period can be described by a Poisson distribution, i.e. $P_m(T) = \frac{(\lambda T)^m}{m!} e^{-\lambda T}$, where m is the number of pulses detected, T is the observation time, and *lambda* is the average decay rate and characterizes the Poisson distribution. Random numbers can then be produced by recording the state of a modulo-M electronic counter every time the GM tube registers a detection.

QRNGs have evolved since then, and with recent advances in the field of quantum optics, many now operate using quantum states of light without need of handling radioactive materials. In fact, photons in a coherent state (roughly, the most "classical" of the quantum states of light, e.g. a laser beam) follow Poissonian statistics [6], and so conceivably a time-of-arrival method similar to those for nuclear decays can be used for RNG. Single photons can also be used quite straightforwardly, by sending them into either a 50:50 beamsplitter or a polarizing

beamsplitter [7]. The photon then has a 50% chance of being in either output arm of the beamsplitter, and clicks on the corresponding detectors can be processed as "0" or "1". As it is, RNGs based off measuring quantum physical processes are one of the most mature quantum technologies, with multiple commercial products available on the market.

The challenge is that is difficult to ascertain how random these sources actually are, and thus they are generally considered "weak," or imperfect [4, 8]. Indeed, while various suites of statistical tests can be used to look for certain patterns in RNG outputs, numerous modern PRNGs, including the Mersenne Twister, will pass every single statistical test and successfully masquerade as true random [1]. More fundamentally, any system that can be described by classical physics, even a chaotic one, is in principle deterministic. Any apparent randomness is then an artifact of an incomplete model [9]. QNRGs might seem an appealing alternative in this case, given that we trust that quantum physics is fundamentally random. In fact, generating random bits seems trivial; simply prepare a qubit in a Hadamard basis state $|+\rangle$ or $|-\rangle$, and measure in the computational basis $\{|0\rangle, |1\rangle\}$. However, the simple reality is that in any real-life scenario, there will always be noise, human error, and calibration variability that sullies the pure randomness of quantum mechanics, not to mention the possibility that an adversary has somehow corrupted the device to provide a cryptographic backdoor [9].

This brings us to the field of certifiable randomness, which has seen several key advances in the past decade. Certifiable randomness is the idea that one can prove that an output sequence of numbers is close to statistically random, usually given a few weak (i.e. general) assumptions. Generally, protocols that generate certifiable randomness are tied to the probabilistic nature of quantum mechanics and involve some form of a statistical or cryptographic test that a quantum computer can pass while a classical computer cannot [9-12]. The natural starting point for this field correspondingly was so-called Einstein-certified randomness, where violations of Bell's inequality certify random numbers, bar the assumption of non-interaction [9-11]. More recently, the assumed hardness of learning with errors (LWE) was used to develop a protocol that performs cryptographic certification, as well as test that a computer is indeed acting quantum [12]. Finally, very recently it has been suggested that sampling problems used to demonstrate quantum supremacy can also be used to generate certified randomness [13-15]. So far, only Einstein-certified RNG has been implemented experimentally, although the bit generation rate is notably limited [9,11]. It remains an open problem to develop devices that implement the other forms of certifications, as well as develop protocols that can maintain throughputs comparable to typical PRNGs and large entropy in/entropy out ratios.

II. TECHNICAL CONTENT

In this section we will review in greater detail the three aforementioned paths toward certifiable randomness: 1) Einstein-certification, 2) cryptographic certification, and 3) quantum supremacy. Key concepts to many of these include entropy and randomness extractors.

Entropy: Entropy has many definitions, but in general is a mathematical measure of disorder or randomness [1, 4]. Higher entropies imply a greater degree of randomness than lower entropies. We will discuss information entropy here, although this is closely related to thermodynamic entropy.

Definition 1 (Shannon entropy). For a random variable X with support S, where the support is $S = \{ x : P(X = x) > 0 \}$, the Shannon entropy of X is defined as:

$$H(X) = -\sum_{x \in S} P(X = x) \log_2 P(X = x)$$

Shannon entropy roughly corresponds to the number of bits of information in each outcome. A uniform distribution over $\{0, 1\}^N$ has Shannon entropy N, since each outcome is equally likely and thus has the maximal amount of randomness.

Definition 2 (Renyi entropy). Renyi entropy is a generalization of Shannon entropy, defined as

$$H_\alpha(X) = \frac{1}{1-\alpha} \log_2 \sum_{x \in S} P(X = x)^\alpha$$

where α is the order. Renyi entropy becomes Shannon entropy for $\alpha = 1$.

Definition 3 (min-entropy). Min-entropy is defined as

$$H_\infty(X) = \lim_{\alpha \to \infty} H_\alpha(X) = -\log_2 \max_{x \in S} P(X = x)$$

Note that min-entropy determines the max number of uniform bits that can be extracted from a distribution, i.e. given min-entropy k, the probability of each outcome is bounded by $P(X = x) \leq 1/2^k$ and thus at most k random bits can be extracted. It turns that this is only a necessary condition to extract k bits, rather than a sufficient one [4].

Randomness Extractors: In general, samples from weakly random sources such as those detailed above do not necessarily follow a uniform distribution [8]. Randomness extractors are algorithms that take the outputs of weakly random sources and extract a sequence of uniformly (or nearly uniformly) distributed random bits. As a simple example of a randomness extractor, von Neumann showed that if you have a Bernoulli sequence with success probability p, then you

can convert it into a uniform sequence by exploiting the fact that $P(10) = P(01) = p(1-p)$. Specifically, the procedure is as follows: split the sequence into consecutive, non-overlapping pairs of bits. Discard '00' and '11', and map '01' to 0 and '10' to 1 [16].

A. Einstein-Certified Randomness

Bell's Theorem and the CHSH Game: Bell famously proved that quantum mechanics and local hidden variable theories are incompatible, providing an experiment where quantum mechanics predicts correlations that would violate the constraints suggested by local hidden variables [17]. The CHSH game is a variant of this experiment, illustrated in Figure 1 [10, 18]. Two parties, Alice and Bob, are given challenge bits x and y respectively, chosen uniformly at random, and want to produce a and b, respectively, such that $x+y = xy$ Alice and Bob do not communicate during the game, although they can collaborate on a strategy beforehand. The classical strategy is to pick a=b=0 each time, which has win probability 75%. The quantum strategy is for Alice and Bob to share an EPR pair $(|00\rangle + |11\rangle)/\sqrt{2}$, and measure in different bases depending on the challenge bit they get. Specifically, Alice measures in the computational basis if $x = 0$, and the Hadamard if $x = 1$; Bob measures in the $|+\pi/8\rangle$ basis if $x = 0$ and the $|-\pi/8\rangle$ basis if $x = 1$, where $|+\pi/8\rangle = \cos(\pi/8)\,|0\rangle + \sin(\pi/8)\,|1\rangle$ and $|-\pi/8\rangle$ follows similarly. Following this strategy, Alice and Bob win $\cos^2(\pi/8) \approx 85\%$ of the time, which turns out to be the optimum strategy via Tsirelson's Inequality [19]. These results imply that if one plays the CHSH game while operating in the "quantum regime" where $0.75 < P_{CHSH} < 0.85$, then the output bits (a, b) must have some randomness, given the general assumption that Alice and Bob are not communicating. This randomness is not necessarily uniformly distributed, and thus the results of the CHSH game have to be run through a randomness extractor [9-11]. Note that two uniformly random bits (x, y) were inputted to get two nonuniformly random bits (a, b) out, resulting in a net loss of entropy.

The solution is to perform some of random expansion. In one of the first experimental demonstrations of an Einstein-certified RNG, Pironio et al. noted that it is not actually necessary that the challenge bits be selected uniformly at random; rather, for a large number of runs n, one can select one of the d2 input pairs (x, y) with probability $1-(d^2-1)q$ and the other d^2-1 with probability q, for small q [9]. Using this technique, approximately quadratic expansion can be achieved where a length $O(\sqrt{n}\log_2\sqrt{n})$ seed results in a $O(n)$ entropy output. Roughly, even though most of the challenge bits are the same, because the process is still random, Alice and Bob cannot predict when the challenges will change and thus cannot use local hidden variables to "cheat." Formally, Pironio et al. quantify the Bell inequality violation using the CHSH correlation function I and its corresponding estimator $\hat{I}$

$$I = \sum_{x,y}(-1)^{xy}[P(a=b|xy) - P(a\neq b|xy)] \quad (1)$$

$$\hat{I} = \frac{1}{n}\sum_{x,y}(-1)^{xy}\frac{[N(a=b|xy) - N(a\neq b|xy)]}{P(xy)} \quad (2)$$

where $P(ab|xy)$ are conditional probabilities and $N(ab, xy)$ are measurement counts. For large n, the conditional probabilities and thus the CHSH correlation I can be estimated even for a highly nonuniform distribution of (x, y) challenges. The correlation estimator, in conjunction with a statistical parameter δ can then be used to bound the min-entropy $H_\infty \geq nf(\hat{I}-\epsilon)$ with probability $> 1-\delta$, where ϵ is determined by δ and the function f from numerical methods [9].

Later work by Vazirani & Vidick went on to further improve randomness efficiency and achieve exponential randomness expansion [10]. Similar to Pironio et al., they use nonuniformly distributed challenges (x, y) to save on randomness. Specifically, to achieve n bits of randomness with "security parameter" $\epsilon > 0$, inputs are blocked off into m blocks of k input pairs (x, y), where $m = C(n\log\frac{1}{\epsilon})$, $C > 1$ and $k = 10\log^2(n)$. Each block only contains one possible input pair, with $10^3\log\frac{1}{\epsilon}$ random blocks chosen to contain one of the possible input pairs uniformly at random, and the rest chosen to contain $(0, 0)$. If in every block the CHSH game is won for at least 84% of the inputs, then the protocol succeeds and $O(n)$ bits of randomness are produced from a seed of length $O(\log n\log\frac{1}{\epsilon})$ [10].

Einstein-certified randomness is of particular interest since it has been implemented experimentally. In Pironio et al's implementation, they employed two entangled atoms to obtain > 42 random bits from $n = 3,016$ trials [9]. Later experiments have expanded to include loophole-free Bell tests and other quantum systems such as photons, with one obtaining $1,024$ random bits from a seed of length $d = 315,844$ [11].

B. Cryptographically-Certified Randomness

Trapdoor Claw-Free Functions (TCF): This is a family of 2-to-1 functions $f : \{0,1\}^n \to \{0,1\}^m$ that possess the following properties: 1) f is efficiently computable classically, 2) for a "claw" (x_1, x_2, y) where $x_1 \neq x_2$ and $f(x_1 = f(x_2) = y$, given a trapdoor it is possible to compute (x_1, x_2) efficiently from y with a classical computer, and 3) without the trapdoor, it is intractable for a quantum computer to compute the claw [12].

The drawback of Einstein-certified randomness is that it is limited to Bell test situations where entanglement is shared across multiple noncommunicating parties. Brakerski et al. got around this by constructing a "post-quantum noisy TCF" (NTCF) family of functions that

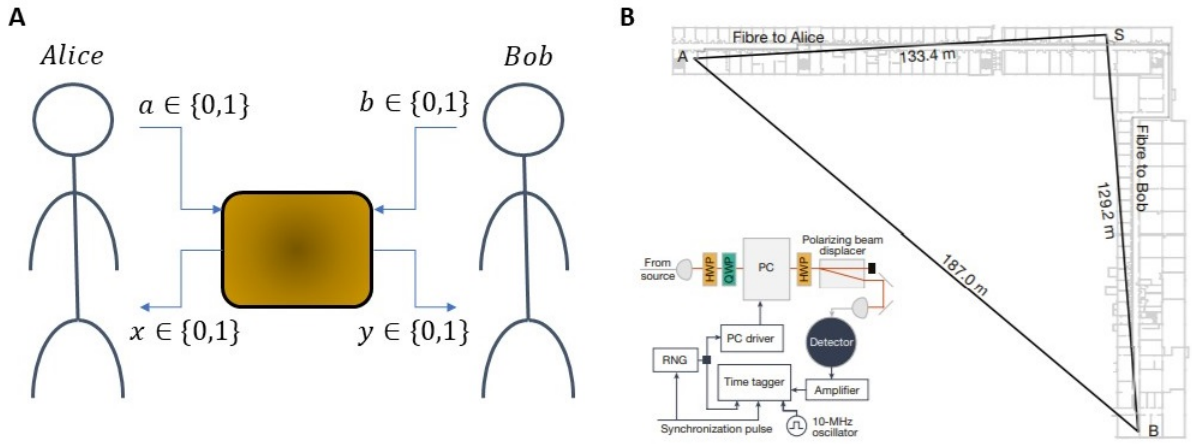

FIG. 1. Diagram of the CHSH Game. A) Schematic of the CHSH game with players Alice and Bob. B) Experimental realization of the CHSH game, using a loophole-free Bell experiment characterized by high-efficiency detectors and space-like separation. Adapted from reference [11].

builds off the conjectured hardness of learning with errors (LWE), a computation problem which seeks to infer a close approximation of a function $f : \mathbb{Z}_q^n \to \mathbb{Z}_q$ from samples (x, y) such that $x \in \mathbb{Z}_q^n$, $y \in \mathbb{Z}_q$ [12]. Specifically, they define a "single round test" of quantumness which consists of two stages: 1) a classical computer, which has knowledge of the trapdoor, sends over a quantum circuit describing the NTCF and queries the quantum computer for a random $y = f(x_1) = f(x_2)$ and then 2) asks the quantum computer to either produce x_1 or x_2 (a random preimage of y), or to produce $d : d \cdot (x_1 + x_2) = 0$. This scheme is illustrated in Figure 2. By definition of the TCF, the classical computer can efficiently verify either of these outputs. The first step can be easily done by creating a uniform superposition over all inputs, acting the provided quantum circuit on them, and then measuring the output bits which will collapse the inputs to the relevant preimage $\frac{1}{\sqrt{2}}(|x_1\rangle + |x_2\rangle)$. If this is measured in the computational basis, then the quantum computer will produce x_1 or x_2 uniformly at random, thus providing random bits. If measured in the Hadamard basis, this will produce the an "adaptive hardcore bit" $d : d \cdot (x_1 + x_2) = 0$. The adaptive hardcore bit has the important property that it is intractable to sample from any distribution on (y, x, d, b) and obtain values that satisfy the conditions $f(x) = y$ and $b = d \cdot (x + x')$ with probability $\geq \frac{1}{2} + \epsilon$, where ϵ is not necessarily negligible. Passing the single round test then implies that the device must have measured and collapsed at least one qubit in superposition (roughly, a qubit has to have been initialized to $|+\rangle$ and measured), and thus the device must be quantum lest there be an efficient way to compute claws [12].

The randomness protocol is then as follows: 1) Use a classical computer to request new preimages for some y to generate nearly uniform random bits, 2) to check for quantumness, occasionally inserts random requests for d instead, and 3) if that test passes, refresh the pseudorandom key used for the NTCF as a security precaution. In terms of randomness expansion, this method can obtain different degrees of expansion depending on how general of an assumption is made about the hardness of their version of LWE. If the hardness assumption is weak and only says that it is hard for polynomial-sized quantum circuits, then $O(N^\epsilon)$, $\epsilon > 0$ bits are required to generate $O(N)$ nearly uniform bits. For a strong hardness assumption that LWE is hard for sub-exponentially sized quantum circuits, then this can be improved to $O(N)$ bits from $poly \log N$ bits [12].

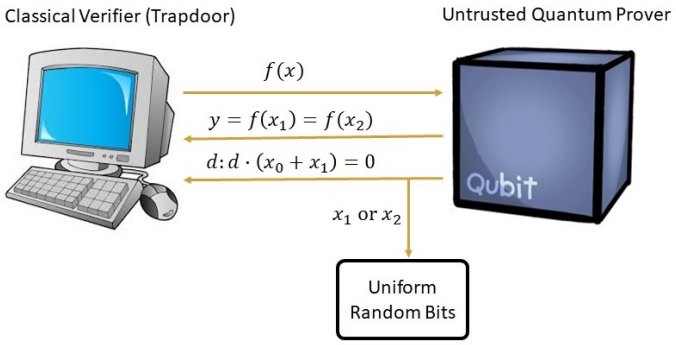

FIG. 2. Diagram of Certified Random Number Generation using Cryptographic Tests. The classical verifier sends one of two challenges to an untrusted quantum prover, known as a single-round test. The quantum prover can efficiently calculate both types of challenges, but cannot efficiently calculate the "claw" of the trapdoor-free claw function without knowledge of the trapdoor the classical verifier possesses. As a result, the result of one of the challenges must be near uniform random bits.

C. Quantum Supremacy

Random Circuit Sampling: Random circuit sampling is the task of (pseudo)randomly generating a n-qubit quantum circuit using gates from a universal gate set, acting these on $|0\rangle^n$, and sampling from the probability distribution of output bitstrings [20]. The distribution of the probabilities $p = |\langle s|\psi\rangle|^2$, defined as the probability of measuring out bitstring s from a state $|\psi\rangle$, where $|\psi\rangle$ is chosen uniformly at random from Hilbert space of size $N = 2^n$, is given by

$$P(p) = (N-1)(1-p)^{N-2} \Rightarrow Ne^{-Np} \tag{3}$$

for large N. This is sometimes called the Porter-Thomas distribution and is characteristic of quantum chaos [20]. It has been argued that simulating this task classically would take exponential overhead, thus making it a strong candidate for demonstrating quantum supremacy.

Heavy Output Generation (HOG): HOG is a relation problem proposed by Aaronson & Chen that is conjectured to be exponentially hard for a classical computer [14]. It states: Given a random n-qubit quantum circuit C, generate output strings $s_1 \dots s_k$, at least 2/3 of which are heavy for C. A bitstring is heavy if it has a probability greater than the median of C's distribution, i.e. $P(s) > med\{probList(C|0\rangle^n)\}$. HOG can be easily solved on a quantum computer by simply running random circuit sampling with C and collecting k samples [14]. In essence, the test indicates that no classical computer can generate the same amount of entropy that random circuit sampling does in a comparable time. This is a particularly remarkable result in that it talks directly about the outputs of a sampling problem, as opposed to the process and corresponding distribution by which the samples are generated, which is nominally the source of quantum supremacy.

Aaronson very recently suggested that this can be used to expand upon Google's quantum supremacy experiment and generate verifiable randomness from moderately sized quantum computers (50-100 qubits) [13-15]. The outline of the proposal is then as follows: 1) Use a trusted classical computer to pseudorandomly generate quantum circuits $C_1 \dots C_M$, 2) perform random circuit sampling on those circuits, which should return a list of k samples $S = (s_1 \dots s_k)$ from the distribution $C_i|0\rangle^n$, 3) check for high entropy in the output bitstrings using HOG (only for a few randomly chosen iterations), and 4) if the statistical tests pass, feed the output bitstrings $S = (S_1 \dots S_M)$ to a randomness extractor. This protocol is illustrated in Figure 3. Note that since the HOG verification takes place on a classical computer, this protocol would only work for moderately sized systems before the computation takes too long [14,15].

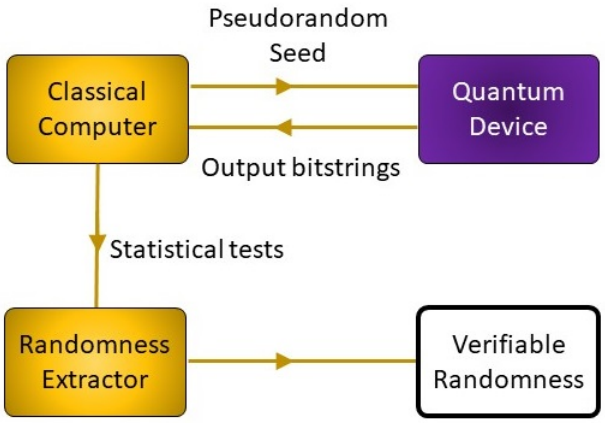

FIG. 3. Proposal for Certified Random Number Generation from Quantum Supremacy. A pseudorandom seed from a classical computer can be used to generate random circuits. A quantum device performs a task such as random circuit sampling and sends the output to a classical computer. The classical computer verifies the randomness and uses the high-entropy output to seed a classical randomness extractor.

III. DISCUSSION

Which of these methods of certifiable randomness is best? To determine this, we have to consider multiple criteria, including the near-term feasibility, the bit rates, and the degree of randomness expansion, i.e. entropy in to entropy out. It is interesting that while Einstein-certified randomness as of now is the only method to have been experimentally implemented, its practicality outlook is the poorest. As noted by Brakerski et al., Einstein-certified randomness is quite limited to situations where entanglement is being shared. It is also worth noting that in the work of Pironio et al., the 3,016 runs required to extract 42 bits of true randomness took around a month to complete [9]. The degree of expansion was additionally limited since they picked challenge bits from a uniform distribution, in order to simplify the experimental setup [9]. Later "loophole-free" experiments have made similar simplifications, which speaks to the fact that in general, Bell tests require complex setups and expensive equipment [11]. If desired for practical applications, future work should consider implementing the protocol of Vazirani & Vidick in order to achieve testable exponential randomness expansion.

In comparison, certified randomness from cryptographic testing and quantum supremacy both appear to offer more appealing outlooks. In terms of randomness expansion, cryptographic certification in the best case could theoretically offer exponential expansion, comparable to that of Einstein certification [12]. In terms of throughput however, the question becomes more complicated. As an illustration, Google's Sycamore superconducting processor is claimed to have taken three minutes to sample one instance of a quantum circuit a million

times, so one might infer that the bit generation rate should be reasonable [13]. However, a major limiting factor is in fact the HOG statistical test used to verify the high-entropic outputs of random circuit sampling. For 50-100 qubits the overhead is reasonable, but above that threshold this protocol becomes intractable for classical supercomputers [14-15]. Cryptographic certification has the potential to be faster (no randomness extractor or HOG needed!), but only becomes practical with qubit counts in the hundreds to thousands. Indeed, Brakerski et al. estimated 2,000 qubits would be needed for 50 bits of security [12]. One can imagine that certifiable randomness from quantum supremacy can play an intermediate role as qubit numbers scale, before cryptographic certification takes over. In fact, this appears to be the case, with certifiable randomness likely to be Google's next big experiment [13].

IV. CONCLUSION

In conclusion, we have provided an overview of recent progress in the field of random number generation. While for many applications pseudo-random generators are sufficient, it is of particular cryptographic and scientific interest to be able to produce numbers that can be "certified" random. Proof of randomness is usually achieved using probabilistic tests that only quantum computers can achieve, tying certification to the inherent randomness of quantum mechanics. The first protocols developed and experimentally demonstrated in this field were based off Bell tests of nonlocality, providing truly random bits at relatively low generation rates. Future experimental work might involve proposals based off cryptographic tests or quantum supremacy, which have promise to be more practical in terms of experimental setup, bit throughput, and randomness expansion. We predict that the quantum supremacy proposal is likely to take off in the near term.

[1] M. Herrero-Collantes and J. C. Garcia-Escartin, Rev. Mod. Phys. **89(1)**, 015004 (2017).
[2] G. Marsaglia, Proc. Natl. Acad. Sci. U.S.A **61(1)**, 25 (1968).
[3] M. Matsumoto and T. Nishimura, ACM Transactions on Modeling and Computer Simulation **8(1)**, 3 (1998).
[4] R. Shaltiel, *International Colloquium on Automata, Languages, and Programming* (Springer, 2011).
[5] H. Schmidt, J. Appl. Phys **41(2)**, 462 (1970).
[6] C. G. Gerry and P. L. Knight, *Introductory Quantum Optics* (Cambridge University Press, 2005).
[7] T. Jennewein, et al., Rev. Sci. Instrum. **71(4)**, 1675 (2000).
[8] D. Zuckerman, Algorithmica **16(4-5)**, 367 (1996).
[9] S. Pironio, et al., Nature **464(7291)**, 1021 (2010).
[10] U. V. Vazirani and T. Vidick, arXiv preprint arXiv:1111.6054 (2011).
[11] P. Bierhorst, et al., Nature **556(7700)**, 223 (2018).
[12] e. a. Z. Brakerski, in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2018).
[13] F. Arute, et al., Nature **574(7779)**, 505 (2019).
[14] S. Aaronson and L. Chen, arXiv preprint arXiv:1612.05903 (2016).
[15] S. Aaronson, Aspects of certified randomness from quantum supremacy (2019), see https://www.scottaaronson.com/talks/.
[16] J. von Neumann, Appl. Math Ser. **12.36-38**, 5 (1951).
[17] J. S. Bell, Physics Physique Fizika **1(3)**, 195 (1964).
[18] J. F. Clauser, et al., Phys. Rev. Lett. **23(15)**, 880 (1969).
[19] B. S. Cirel'son, Letters in Mathematical Physics **4(2)**, 93 (1980).
[20] S. Boixo, et al., Nat. Phys. **14(6)**, 595 (2018).