Verification of Quantum Computation and Randomness Generation via Timed Random Circuit Sampling: Protocol, Perspectives, and Theoretical Equivalences

Generated Document based on User Input
Analysis incorporating referenced works and subsequent discussion

March 28, 2025

Abstract

The advent of cloud-based quantum computing platforms necessitates reliable methods for verifying the authenticity and capability of remote quantum processors. Furthermore, harnessing quantum phenomena for generating certified randomness is a significant application. This paper reviews a protocol proposed by Aaronson et al., designed to achieve both goals simultaneously using random circuit sampling (RCS). The protocol involves a classical client challenging a quantum server with randomly generated quantum circuits, demanding results within a time limit presumed too short for classical simulation. Verification relies on statistical analysis, potentially using techniques like Cross-Entropy Benchmarking (XEB), and successful verification allows the client to extract certified random bits from the results. We discuss the underlying principles, particularly RCS and XEB, analyze the protocol's structure, and critically examine its assumptions and potential limitations, including the theoretical equivalence of the core task implementation on different universal quantum computing architectures.

1 Introduction

Quantum computers promise computational capabilities exceeding classical machines for specific tasks [1]. As access to quantum hardware increasingly occurs via remote servers, verifying that a service provider genuinely utilizes a quantum computer, rather than simulating or spoofing results, becomes crucial [2, 3]. Simultaneously, quantum mechanics offers a foundation for generating true randomness, essential for cryptography, simulation, and scientific applications.

Scott Aaronson and collaborators have proposed protocols aimed at remotely verifying quantum computation while simultaneously generating certified random numbers [3, 4]. These protocols leverage the concept of computational hardness, specifically the presumed difficulty for classical computers to simulate certain quantum processes, such as Random Circuit Sampling (RCS). The core idea is to issue challenges (random quantum circuits) that only a true quantum computer could solve within a stringent time limit, using the verified output to generate randomness. This work reviews such a protocol, outlines its components based on published descriptions [2, 4], and discusses critical perspectives on its underlying assumptions and practicality.

2 Random Circuit Sampling and Verification

2.1 Random Circuit Sampling (RCS)

Random Circuit Sampling (RCS) is a task proposed as a benchmark for demonstrating quantum computational advantage, often referred to as "quantum supremacy" [5]. It involves the following conceptual steps:

1. Construct a quantum circuit composed of randomly chosen quantum gates (typically single- and two-qubit gates drawn from a universal set) arranged in a specific pattern or depth.

2. Execute this circuit on a quantum processor multiple times.

3. Measure the final state of the qubits for each execution, yielding a set of output bitstrings (samples).

The distribution of these output bitstrings, $P_i(s)$, is determined by the specific random circuit U_i. For sufficiently large and complex circuits, it is strongly believed that, due to quantum parallelism amounting to exponential performance when compared to classical bits, classically simulating this process and drawing samples from the correct output distribution requires computational resources that grow exponentially with the number of qubits and circuit depth, making it intractable for current and foreseeable classical supercomputers [5]. The sequence of outputs collected across multiple different challenge circuits $\{U_i\}$ is thus drawn from a collection of different, complex probability distributions $\{P_i(s)\}$.

2.2 Verification via Cross-Entropy Benchmarking (XEB)

A key challenge in RCS experiments is verifying that the quantum processor is performing the task correctly, albeit with inherent noise. A commonly used metric is the fidelity, which quantifies how close the experimental output distribution is to the ideal, noise-free distribution predicted by theory.

Cross-Entropy Benchmarking (XEB) is a technique used to estimate this fidelity [5]. In its linear form (LXEB), it compares the probabilities of observed bitstrings under the experimental distribution versus the ideal distribution. For a set of observed bitstrings $\{s_k\}$ obtained from executing circuit U_i, the LXEB fidelity F_{XEB} can be estimated as [6]:

$$F_{XEB} \approx 2^n \langle P_{ideal,i}(s_k) \rangle_k - 1$$

where n is the number of qubits, $P_{ideal,i}(s_k)$ is the probability of observing string s_k from the ideal simulation of circuit U_i, and $\langle \cdot \rangle_k$ denotes the average over the experimentally observed samples for that circuit.

Calculating $P_{ideal,i}(s_k)$ for all possible s_k requires simulating the circuit U_i, which is classically hard for large circuits. To overcome this, techniques like *patch XEB* are employed. The circuit is divided into smaller, classically simulatable "patches". The fidelity of each patch is computed, and the overall fidelity of the full circuit is estimated by combining these patch fidelities (often by multiplication, assuming uncorrelated errors) [5]. XEB serves as a practical method to assess the performance of the quantum device executing the RCS task.

3 The Aaronson Protocol for Quantum Verification and Randomness Generation

Based on descriptions in [2, 4], a protocol for verifying a remote quantum computer and generating randomness can be outlined as follows:

1. **Challenge Generation:** A classical client (verifier) generates a description of an apparently random quantum circuit U_i. This circuit should be large enough to be considered hard to simulate classically within the protocol's time constraints.

2. **Challenge Transmission:** The client sends this circuit description U_i as a challenge to the server claiming to possess a quantum computer (prover).

3. **Timed Execution and Response:** The server executes the specified quantum circuit U_i on its hardware and measures the output state, obtaining one or more result bitstrings s_k. The server must return these result strings to the client within a strict time limit Δt. This time limit is chosen to be short enough to preclude the server from completing a classical simulation of the circuit U_i.

4. **Verification:** The client receives the response(s). Over multiple rounds with different random circuits $\{U_i\}$, the client performs verification checks. This involves classically simulating parts (or patches) of the circuits and comparing the statistical properties of the received results (e.g., using XEB) against the expected properties of an ideal quantum execution. The verification module determines whether the collective evidence supports the claim that the results were generated by the specified quantum circuits executed on a genuine quantum computer.

5. **Randomness Extraction:** If the verification step is successful across multiple challenges, the client accepts the server as authentic (for these tasks). The client can then use one or more of the received result strings $\{s_k\}$, potentially processed through a randomness extractor function, to generate a sequence of high-quality random bits.

This protocol aims to provide cryptographic assurance: the time constraint prevents classical simulation, and the statistical verification (like XEB) confirms fidelity to the quantum process. The output randomness is thus certified by the verified execution on a computationally superior quantum device.

4 Discussion and Critique

While the protocol presents an innovative approach, several points warrant critical discussion:

- **Reliance on XEB Sufficiency:** The verification step relies heavily on statistical tests like patch XEB being sufficient to distinguish genuine quantum computation from spoofing within Δt. Questions remain about the robustness of these methods against adversaries not using direct classical simulation. Could a simpler device generate outputs that pass XEB tests without performing the full computation?

- **Necessity of Universal Quantum Computers for Randomness Only:** The protocol requires an advanced, programmable quantum computer for verification via RCS. If randomness generation were the sole goal, simpler dedicated QRNGs based on various physical phenomena might suffice and be more cost-effective [7]. The protocol's complexity arises from linking randomness to the verification of a *computationally powerful* device.

- **Random Circuits vs. Deterministic Evolution (Initial View):** Initially, one might contrast RCS with fixed deterministic quantum evolutions, like a single Hadamard gate implemented via beam splitters. A fixed Hadamard produces samples from one fixed distribution, whereas RCS (across challenges) samples from many different distributions $P_i(s)$. This highlights a phenomenological difference in the output stream's stationarity.

- **Theoretical Equivalence of Universal Platforms:** However, the distinction becomes nuanced when considering *universal* quantum devices. The work by Reck et al. [8] demonstrates that any unitary transformation $U(N)$ can, in principle, be implemented using a finite number of controllable optical components (a generalized beam splitter network). Such a device, if perfectly controlled, is a universal quantum processor for N modes. Therefore, it could be programmed to execute the *exact same sequence* of random unitary

circuits $\{U_i\}$ as demanded by the RCS protocol. In this ideal, noise-free scenario, the *output phenomenology* (the sequence of samples drawn from the varying distributions $P_i(s)$) would be *indistinguishable* from that of an ideal gate-based quantum computer running the same protocol. The core computational task and its ideal output signature can be identical across different universal quantum architectures.

5 Alternative Implementations and Practical Distinctions

The theoretical insight that a generalized optical interferometer can perform the same computations as a gate-based quantum computer (including RCS) is significant [8]. This implies that the *choice* of physical platform (e.g., superconducting qubits, trapped ions, photonics) might not fundamentally limit the ability to perform the core task required by protocols like Aaronson's, assuming universality can be achieved.

However, this theoretical equivalence does not erase the substantial practical differences between these platforms:

- **Error Models:** The dominant noise sources and error mechanisms differ vastly. Superconducting qubits face decoherence, gate errors, and readout noise. Photonic systems contend with photon loss, detector inefficiency, phase instability, and mode mismatching. Verification techniques like XEB might need adaptation or replacement depending on the platform and its specific error characteristics.
- **Control and Engineering:** The engineering challenges for building, stabilizing, and precisely controlling thousands of coupled qubits or thousands of phase shifters in an interferometer are immense and distinct. The methods for compiling an abstract circuit U_i into physical control sequences (microwave pulses vs. phase settings) are entirely different.
- **Technological Maturity:** Currently, significant experimental progress on RCS and demonstrations of quantum advantage have utilized superconducting qubit platforms [5]. While linear optics is foundational, building large-scale, fully programmable, low-loss interferometers capable of complex computations faces its own set of hurdles.

Therefore, while the Aaronson protocol could theoretically be implemented on any universal quantum computer, its practical feasibility, security analysis (especially concerning the time limit Δt relative to realistic error correction or simulation attempts tailored to specific hardware), and verification methodology are deeply intertwined with the specific physical platform being used. The discussion around the protocol is often implicitly tied to gate-based systems due to current experimental focus, but the underlying principles could apply more broadly, contingent on overcoming the platform-specific practical challenges.

6 Conclusion

The protocol proposed by Aaronson and collaborators offers an innovative framework for combining remote quantum computer verification with certified randomness generation, using RCS hardness and time limits. Its verification relies on statistical checks like XEB to confirm fidelity to the requested quantum process.

Critical analysis highlights the dependence on the robustness of verification methods. Furthermore, while the protocol demands a universal quantum computer, the theoretical work of Reck et al. [8] shows that universality is not exclusive to gate-based models; optical interferometers possess this capability in principle. This implies that the ideal output phenomenology

of RCS could be identical across different universal platforms executing the same sequence of circuits.

Despite this theoretical equivalence, the practical implementation, dominant error sources, control mechanisms, and current maturity levels differ significantly between platforms like superconducting qubits and photonic interferometers. The practical security and feasibility of the protocol must therefore be assessed within the context of the specific hardware used. Future research should continue to explore the security of verification techniques and compare the potential of different quantum architectures for implementing such verification and randomness generation protocols.

References

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[2] S. Aaronson, "My visit to Google to see the new quantum supremacy experiment". Shtetl-Optimized Blog Post, October 23, 2019. Available: `https://scottaaronson.blog/?p=4372` (Note: The user provided `https://scottaaronson.blog/?p=8746`, which discusses later developments/protocols, accessed March 28, 2025).

[3] S. Aaronson and S.-H. Hung, "Certified Randomness from Random Circuit Sampling". arXiv preprint arXiv:2303.01625 [quant-ph], 2023. Available: `https://arxiv.org/abs/2303.01625` (Accessed March 28, 2025).

[4] S. Aaronson, "System and Method for Generating Certified Randomness Using a Cloud Quantum Computer". US Patent Application US20220100473A1, filed October 15, 2020, published March 31, 2022. Available: `https://patents.google.com/patent/US20220100473A1/en` (Accessed March 28, 2025).

[5] Google AI Quantum Team, "Validating random circuit sampling as a benchmark for measuring quantum progress". Google Research Blog Post, November 17, 2022. Available: `https://research.google/blog/validating-random-circuit-sampling-as-a-benchmark-for-measuring-quantum-progress/` (Accessed March 28, 2025).

[6] Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., ..., Martinis, J. M., "Quantum supremacy using a programmable superconducting processor". *Nature*, vol. 574, no. 7779, p. 505-510, 2019. DOI: 10.1038/s41586-019-1666-5

[7] M. Herrero-Collantes and J. C. Garcia-Escartin, "Quantum random number generators". *Reviews of Modern Physics*, vol. 89, no. 1, p. 015004, 2017. DOI: 10.1103/RevModPhys.89.015004

[8] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, "Experimental realization of any discrete unitary operator". *Physical Review Letters*, vol. 73, no. 1, pp. 58-61, 1994. DOI: 10.1103/PhysRevLett.73.58