

On-Chip Verified Quantum Computation with an Ion-Trap Quantum Processing Unit

Cica Gustiani^{*}*LIP6, CNRS, Sorbonne Université, 4 Place Jussieu, 75005 Paris, France*Dominik Leichtle[†], Jonathan Miller, and Ross Grassie*School of Informatics, University of Edinburgh, 10 Crichton Street, EH8 9AB Edinburgh, United Kingdom*Daniel Mills[‡]*Quantinuum, Terrington House, 13-15 Hills Road, Cambridge CB2 1NL, United Kingdom*Elham Kashefi[§]*School of Informatics, University of Edinburgh, 10 Crichton Street, EH8 9AB Edinburgh, United Kingdom
and LIP6, CNRS, Sorbonne Université, 4 Place Jussieu, 75005 Paris, France*

(Received 16 December 2024; accepted 4 August 2025; published 14 October 2025)

We present a novel approach to cryptographically secure verification and benchmarking of quantum computing, demonstrating our approach on an ion-trap quantum computer. Unlike previous cryptographically secure verification protocols, which typically require quantum communication between client and server, our approach is implemented entirely on chip. This eliminates the need for a quantum capable client, and significantly enhances practicality. We perform tomography to justify the additionally required assumption that the noise is independent of the secret used to prepare the server's single-qubit states. We quantify the soundness error that may be caused by residual secret dependencies. We demonstrate our protocol on the 20-qubit Quantinuum H1-1 ion-trap quantum processing unit, using qubit measurements and resets to construct measurement patterns with up to 52 vertices. To our knowledge, these are the largest verified measurement-based quantum computations performed to date.

DOI: [10.1103/jpms-v3kw](https://doi.org/10.1103/jpms-v3kw)

Introduction—As quantum technology matures it brings with it a diverse array of exciting but difficult to classically simulate applications. While quantum processing units (QPUs) remain susceptible to noise, scalable and rigorous methods for verification will be essential to ensure confidence in the output of these computations.

A selection of benchmarking techniques exist to measure the accuracy of a QPU [1]. These may either be indicative of component [2] or holistic performance [3–5] from which the performance of a particular computation can be inferred. However, this inference requires either that some specific noise model is present, that the noise is separable between qubits, or that the noise depends only on some general characteristics of the circuit (such as two-qubit gate count). When these assumptions are not met such schemes

give no clear guarantees of the correct performance of arbitrary quantum algorithms.

Verifiable universal blind quantum computation (VUBQC) circumvents these shortcomings by utilizing cryptographic techniques; treating deviations and noise as potential security threats [6–9]. This gives the client the power to recognize corrupted results returned by a noisy or malicious quantum server. The efficiency of these schemes has drastically improved with time [6,8–13], leading to their first experimental implementations [14,15].

VUBQC relies on physically separating the verifier's trusted small quantum device (e.g., random single-qubit generator) from the large QPU to be tested. This allows the full computation to be obfuscated, ensuring that any malicious behavior of the server (any general noise of the QPU) is independent of the description of the computation. This in turn guarantees that the behavior is the same on tests, which are interleaved in the computation, and the desired computation itself.

However, such schemes require the transfer of quantum information between the verifier and QPU, adding significant noise and technological overhead. We propose and demonstrate an alternative means to enforce the requirements of VUBQC, called “on-chip verification.” We replace a fully malicious server with an honest but potentially noisy device.

^{*}Contact author: cicagustiani@gmail.com

[†]Contact author: dominik.leichtle@ed.ac.uk

[‡]Contact author: daniel.mills@quantinuum.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

By doing so, we remove the need for a physically separated verifier to achieve blindness, but maintain the independence of the server's actions from the type of round, compute, or test that it is implementing.

"Honest but noisy" describes current QPU providers. As such, and because our protocol requires no quantum communication, we implement it with existing technology. Since quantum communication will not be widely available for many years, we facilitate the early use of VUBQC. This enables verification of near term application-oriented computational advantage demonstrations, improving confidence in such demonstrations.

Rather than relying on computational assumptions or introducing new hardware overheads, we trade off quantum communication for assumptions on the behavior of the server's quantum device. In particular we make the minimal assumption that the noise is a completely positive trace-preserving map independent of the secret parameters used to prepare the input single-qubit states. We provide a methodology for measuring this secret dependency that, due to the robustness property of the protocol, will introduce a bound on the confidence of the accepted computation. This is a notable conceptual shift away from making idealized noise assumptions, and we expect applications of this approach elsewhere to be fruitful.

While other proof of principle classical client verification schemes have been demonstrated [16], a significant qubit overhead would be required to meet the computational assumptions of these schemes [17,18]. Protocols employing an information-theoretically secure cryptographic approach have also been demonstrated using photons to transfer data [14]. In that work, noise levels were identified as too high to scale the computation beyond a few qubits. Accreditation [19] is another gate-based approach to verification that attempts to replace cryptographic techniques with assumptions on the QPU's noise. However, the noise assumptions required to apply accreditation are strong compared to our minimal requirements.

We demonstrate that our desired noise conditions are closely met by the Quantinuum H1-1 quantum charge-coupled device trapped ion device. We then exemplify our on-chip verification protocol with two classes of computations. In doing so we demonstrate the largest successful verified measurement-based quantum computation pattern, successfully constructing and utilizing a 52-vertex graph state for computation.

On-chip verification and benchmarking protocols—Our on-chip verification protocol, described in Protocol 1 and outlined in Fig. 1, adapts the noise-robust verification protocol from [8], Protocol 1]. The bounded-error quantum polynomial time (BQP) verification protocol proposed by [8] works by interleaving computation rounds, implementing randomly compiled versions of the target computation, with test rounds, which (probabilistically) detect any error that has the potential to harm the computation results.

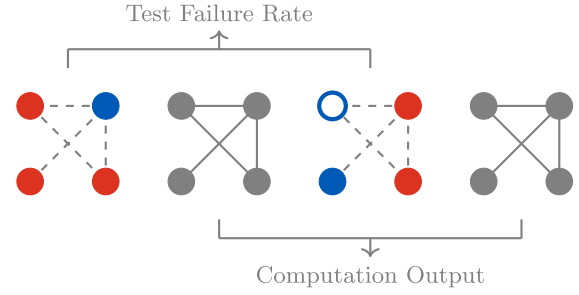


FIG. 1. Outline of on-chip verified quantum computing, as described in Protocol 1. The client sends a description of an MBQC computation to the server. The server randomly alternates between performing computation rounds and test rounds. Computation rounds, seen in gray, are randomized versions of the client's computation, with the same ideal output. By randomly compiling the computation rounds the actions of the server are indistinguishable between rounds. The output from the computation rounds is then ideally the original computation output. During test rounds a random color from a coloring of the graph is selected. Colored vertices are assigned as test vertices, seen in blue, and are disentangled using dummy vertices, seen in red. The measurement outcomes of test vertices are deterministic and known in the ideal case. Test rounds where any measurement is found to be incorrect, indicated with unfilled vertices, are said to have failed. The output of the test rounds is the test failure rate, which, as the operations performed by the server are indistinguishable between rounds, gives a measure of the confidence in the computation output.

The protocol is robust to a constant level of global noise and has an efficient number of repetitions as the only overhead.

Like [8], Protocol 1], Protocol 1 relies on measurement-based quantum computation (MBQC) [20], which performs computation by adaptively measuring a "graph state" of entangled qubits. While [8], Protocol 1] has the client prepare the qubits before sending them to the server to be entangled, Protocol 1 moves the preparation of the randomized single-qubit states from the client's to the server's device.

In this Letter, we prepare and measure qubits in the YZ plane, with the prepared states defined as

$$|\theta\rangle_{YZ} \equiv |\theta\rangle = \cos\frac{\theta}{2}|0\rangle - i\sin\frac{\theta}{2}|1\rangle. \quad (1)$$

Measurement with angle θ corresponds to projectors $\{|\theta\rangle\langle\theta|, |\theta+\pi\rangle\langle\theta+\pi|\}$. In the YZ plane, an edge corresponds to applying $XX_{ij} = H_i H_j C Z_{ij} H_i H_j$.

The prepared single-qubit states at the beginning of the protocol are chosen from the set $\{|\theta\rangle|\theta \in \Theta\} \cup \{|+\rangle, |-\rangle\}$, where $\Theta = \{(j\pi/4)|j = 0, \dots, 7\}$ and $|\theta\rangle = R_x(\theta)|0\rangle$. While previously proposed verification protocols use states in the XY plane, Protocol 1 is formulated using rotated states in the YZ plane, as this is the appropriate choice for the H1-1 machine, where $R_z(\theta)$ is implemented virtually.

Theorem 1 captures the security of Protocol 1 and formalizes the assumptions necessary for on-chip verifiability. Note that, unlike in other VUBQC protocols, the notion of blindness is not applicable here, and so not referenced in Theorem 1, as the server has full access to all previously secret information and parameters of the protocol.

Theorem 1 (Security of on-chip verification protocol)—For $n = d + t$ such that d/n and t/n are fixed in $(0,1)$ and w such that w/t is fixed in $\{0, (1/k) \cdot [(2p-1)/(2p-2)]\}$, where p is the inherent error probability of the BQP computation with a k colorable MBQC graph implementing it, Protocol 1 with d computation rounds, t test rounds, and a maximum number of tolerated failed test rounds of w achieves ε verifiability with ε exponentially small in n under the following assumptions: (1) All classical computations and controls on the server's side are operated honestly. (2) The noise during the initial single-qubit preparation is secret-independent, i.e., the preparation of the states can be simulated by applying a noise channel to perfectly prepared states that are independent of their classical description.

Protocol 1 differs from [[8], Protocol 1] only in the initial preparation of the single-qubit states, which is performed on the server's quantum device rather than on the client's. In the following, we show that the assumed secret independence of the initial preparation noise in Protocol 1 means that this noise could just as well have been simulated by the server in [[8], Protocol 1]. This reduces the security of Protocol 1 to the security of [[8], Protocol 1].

Proof sketch—Assumption (2) in Theorem 1 guarantees that the noise during these preparations can be simulated by the server using a secret-independent noise channel after perfect preparations. In this way, errors due to the noise during the preparation phase are accounted for in the traditional security analysis of the verification protocol that does not restrict the server's deviations from the protocol instructions as long as they are independent of the client's secrets d_v, θ_v, r_v . In this way, Protocol 1 is equivalent to a protocol in which the single-qubit preparation is performed on a trusted and perfect device that is separated from the server and directly instructed by the client.

Assumption (1) in Theorem 1 guarantees that no other operations on the server's device depend on the client's secrets. Consequently, the server in this hybrid protocol becomes entirely secret-independent, which maps any potential attack on Protocol 1 satisfying the above assumptions back to the original verification protocol from [8] in a setting with physical separation between the trusted client and the untrusted server.

The application of [[8], Theorem 1] then implies the claim. ■

In the case of deterministic computations with inherent error probability $p = 0$ and $k = 2$ different types of test

rounds, the acceptable threshold for the choice of w/t equals 25%. Hence, also the maximum test failure rate due to noise that the protocol can be made robust to is 25%. In our experiments $k = 2$ is the relevant parameter, and so we will use the 25% test failure rate as a benchmark.

VUBQC protocols that achieve composable security must have the property of “independent verifiability” [21], which is to say the client's decision to accept or reject at the end of the protocol is effectively independent of the delegated computation [22]. After the successful execution of such a protocol, the client will have not only learned the correct result of the target computation evaluated on their input, but also that the protocol would have accepted with the same probability for any other target computation consuming the same resources. This observation led to the design of verification-inspired benchmarking schemes [23], which reinterpret verification routines as cryptographic quantum benchmarks. Following the benchmarking paradigm of [23], Protocol 1 can straightforwardly be reinterpreted as a benchmarking scheme, as we will do in our experiments.

Assessment of protocol requirements—Two key security requirements of Protocol 1 are (1) that any noise during the initial single-qubit state preparation is independent of the secrets and (2) that high-quality random numbers must be available. Fresh random numbers are required for each measurement shot, with circuit branching determined by these values. We generate these random numbers on the fly by measuring Hadamard basis states in the computational bases. This configuration is detailed and justified in Supplemental Material (SM) [24].

The secret independence of noise during the initial state preparations naturally holds in the traditional client-server setups. This is due to their spatial separation, and the explicit assumption that the client is operating an honest and perfect device. However, since our approach eliminates this separation, we must directly evaluate such noise secret dependency. If the noise during the state preparation is close in trace distance to a secret-independent noise channel, it can degrade the security guarantees of the protocol only by an amount proportional to this distance. This follows due to the limited distinguishability of the realistically prepared states from a family of states suffering at most from harmless, secret-independent noise.

The secret dependency is quantified by

$$\min_{\mathcal{E}} \frac{1}{|\Theta|} \sum_{\theta \in \Theta} \|\mathcal{E}(|\theta\rangle\langle\theta|) - \rho_{\theta}\|_1 := \min_{\mathcal{E}} \Delta(\mathcal{E}), \quad (2)$$

where θ is the secret, $|\theta\rangle$ is the noiseless state preparation, ρ_{θ} is the prepared state in the quantum computer, and $\|\cdot\|_1$ is the trace norm. Thus, the noise is independent of the secret if there exists a completely positive trace-preserving map $\mathcal{E}$ such that $\Delta(\mathcal{E}) = 0$. $\Delta(\mathcal{E})$ then quantifies distinguishability between resources in the ideal and the real world.

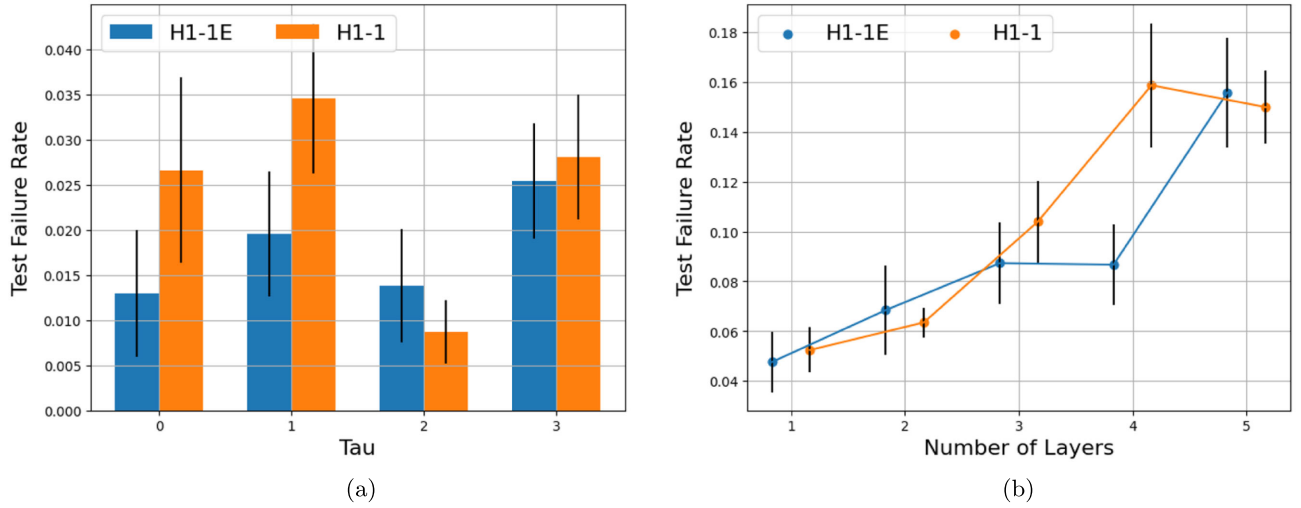


FIG. 2. H1-1 and H1-1E test failure rate experiments. Error bars are generated by resampling 800 shots, from a total of 1000 per circuit, per device, 10 times, and taking the standard deviation of the resampled test failure rate. The bars and scatter points indicate the mean of those resampled values. (a) Two qubit Grover. (b) CNOT grid circuits.

Despite the high fidelity of state preparation (≈ 0.997), our analysis reveals that the noise in the state preparation is not completely secret-independent. Using the optimization under Frobenius norm, we obtain a map $\tilde{\mathcal{E}}$ with objective value $\varepsilon_F = 0.015$, where

$$\tilde{\mathcal{E}}_P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.990 & 0.011 & 0.003 \\ 0 & -0.011 & 0.991 & 0.005 \\ 0 & -0.003 & 0.001 & 0.999 \end{pmatrix} \quad (3)$$

in the Pauli transfer matrix representation. The block-diagonal structure indicates the presence of rotation errors. Indeed, the states resulting from the tomography in SM [24] show the presence of low dissipative errors, as indicated by high purity average, i.e., $\text{tr}[\rho^2] = 0.995$. We elaborate on the details of our optimization and tomography results in SM [24].

Using the same map $\tilde{\mathcal{E}}$, we obtain a trace norm value of $\varepsilon_1 = 0.013$. This shows that the security guarantees of our protocol are not degraded more by the secret dependency of the initial single-qubit state preparation than ε_1 multiplied by the number of qubit preparations [43]. This remains a potentially loose bound; we leave the question of tighter bounds for future research.

On-chip verification experiments—We perform our experiments on the Quantinuum H1-1 quantum charge-coupled device ion-trap device [44], and its noisy classical emulator “H1-1E” [45–47]. We describe the results of two on-chip verification experiments: a two-qubit Grover’s search algorithm; and a larger measurement pattern implementing a grid of CNOTs.

We will study two quantities, as outlined in Fig. 1. (1) Test failure rate, The fraction of the test rounds where at least one trap fails; (2) Incorrect output rate, The fraction of the computation rounds where the incorrect output is produced. We present results for these experiments in SM.

It is possible to calculate the incorrect outcome rate in these cases as the ideal correct outcome is known and deterministic [48]. The two-qubit Grover’s search pattern is small enough that the solution is known but uses components that when scaled would not be efficiently classically simulatable. The CNOT grid implements a classical circuit that can be used as a scalable benchmark of verified quantum computation. However, the CNOT grid experiment measurement pattern uses a number of qubits far larger than any other verified MBQC computation implemented to date.

We give details on how on-chip verified computation can be compiled to the H1-1 device in SM [24]. In order to automate the generation of circuits implementing our on-chip verification scheme from an initial description of an MBQC computation, we present OCVQC-PY [49], which we use in all of our experiments.

Verifiable two-qubit Grover’s algorithm: We implement a verifiable two-qubit Grover search algorithm, as depicted in Fig. 3, for all four possible queries. Note that we execute the measurement pattern in a lazy fashion [50], initializing qubits only when required, which utilizes only three physical qubits.

The results of this experiment can be seen in Fig. 2(a). In all cases the failure rate is comfortably less than 25%, which is to say within the tolerance of our verification scheme. As seen in SM [24] the experiments also consistently yield the correct answer with high probability.

As a side effect, we have also benchmarked a 4×2 rectangular graph state resource, as depicted in Fig. 3(a). This implies that one will be able to successfully perform

computations other than the Grover search on the resource of Fig. 3(a) on H1-1 device, with the same confidence.

CNOT grid: Here, we implement CNOT grid circuits, as seen in Fig. 4, in MBQC, and verify the resulting computation. These circuits consist of m layers of CNOT gates acting between n neighboring qubits in a line architecture. If the input is a tensor product of computational basis states then the ideal output can be calculated efficiently classically. As discussed, one can interpret the results from the test rounds of this experiment as a benchmark of a larger class of circuits. As such the larger CNOT grid computations presented here give insights into the performance of computations that could not be calculated efficiently classically.

Four qubit CNOT circuits with one to five layers, each with a random input state, were generated and run on H1-1, with the results presented in Fig. 2(b). In all cases the average test failure rate is below 25%. The largest measurement pattern run in Fig. 2(b) consists of 52 vertices, which is the largest verified MBQC computation to be run on a QPU to date. Results from extensive classical simulation of such circuits can be found in SM [24] and indicate that graphs with as many as 78 vertices could be expected to run successfully.

Conclusion—In this Letter, we have presented an on-chip verification and benchmarking protocol for quantum computations on near-term QPUs. Our protocol requires no quantum communication between client and server, but gives guarantees on the accuracy of the computation assuming a nonadversarial server with a very general noise model. Our protocol requires only a simple tomographic experiment to ensure the assumptions of the scheme are met.

We exemplify our protocol through experimentation on the Quantinuum H1-1 device. We see strong performance on MBQC patterns with up to 52 qubits, which is the largest verified MBQC computation implemented to date. While larger graph states have been created [51,52], to our knowledge no larger graph has been used for computation. Results from emulations of the H1-1 device predict strong performance with graphs containing up to 78 vertices.

Experiments were conducted via Quantinuum Nexus [53] on the Quantinuum H1-1 QPU [54].

Acknowledgments—We thank Alec Edgington for support with circuit construction, Vanya Eccles for support with the Nexus, and Charlie Baldwin, Cristina Cirstoiu, and Karan Khathuria for their valuable reviews of the paper. D.L. and E.K. acknowledge support by the Quantum Advantage Pathfinder (QAP) research programme within the UK's National Quantum Computing Centre (NQCC). D.L. acknowledges support by the Quantum Computing and Simulation (QCS) Hub. D.L., J.M., and R.G. acknowledge support by the Autonomous Quantum Technologies (AutoQT) project (UKRI Project No. 10004359). C.G. acknowledges support by the

French national quantum initiative managed by Agence Nationale de la Recherche in the framework of France 2030 with the reference ANR-22-PNCQ-0002.

Data availability—The data that support the findings of this article are openly available [49,55].

- [1] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, Quantum certification and benchmarking, *Nat. Rev. Phys.* **2**, 382–390 (2020).
- [2] J. Helsen, I. Roth, E. Onorati, A. H. Werner, and J. Eisert, General framework for randomized benchmarking, *PRX Quantum* **3**, 020357 (2022).
- [3] D. Mills, S. Sivarajah, T. L. Scholten, and R. Duncan, Application-motivated, holistic benchmarking of a full quantum computing stack, *Quantum* **5**, 415 (2021).
- [4] T. Lubinski, J. J. Goings, K. Mayer, S. Johri, N. Reddy, A. Mehta, N. Bhatia, S. Rappaport, D. Mills, C. H. Baldwin, L. Zhao, A. Barbosa, S. Maity, and P. S. Mundada, Quantum algorithm exploration using application-oriented performance benchmarks, [arXiv:2402.08985](https://arxiv.org/abs/2402.08985).
- [5] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Validating quantum computers using randomized model circuits, *Phys. Rev. A* **100**, 032328 (2019).
- [6] J. F. Fitzsimons and E. Kashefi, Unconditionally verifiable blind quantum computation, *Phys. Rev. A* **96**, 012303 (2017).
- [7] A. Broadbent, J. Fitzsimons, and E. Kashefi, Universal blind quantum computation, in *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, New York, 2009).
- [8] D. Leichtle, L. Music, E. Kashefi, and H. Ollivier, Verifying BQP computations on noisy devices with minimal overhead, [arXiv:2109.04042](https://arxiv.org/abs/2109.04042).
- [9] A. Gheorghiu, T. Kapourniotis, and E. Kashefi, Verification of quantum computation: An overview of existing approaches, *Theory Comput. Syst.* **63**, 715 (2018).
- [10] E. Kashefi and P. Wallden, Optimised resource construction for verifiable quantum computation, *J. Phys. A* **50**, 145306 (2017).
- [11] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier, Unifying quantum verification and error-detection: Theory and tools for optimisations, *Quantum Sci. Technol.* **9**, 035036 (2024).
- [12] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier, Asymmetric quantum secure multi-party computation with weak clients against dishonest majority, [arXiv:2303.08865](https://arxiv.org/abs/2303.08865).
- [13] E. Kashefi, D. Leichtle, L. Music, and H. Ollivier, Verification of quantum computations without trusted preparations or measurements, [arXiv:2403.10464](https://arxiv.org/abs/2403.10464).
- [14] P. Drmota, D. Nadlinger, D. Main, B. Nichol, E. Ainley, D. Leichtle, A. Mantri, E. Kashefi, R. Srinivas, G. Araneda, C. Ballance, and D. Lucas, Verifiable blind quantum computing with trapped ions and single photons, *Phys. Rev. Lett.* **132**, 150604 (2024).
- [15] B. Polacchi, D. Leichtle, G. Carvacho, G. Milani, N. Spagnolo, M. Kaplan, E. Kashefi, and F. Sciarrino, Experimental verifiable multi-client blind quantum computing on a Qline architecture, *Phys. Rev. Lett.* **134**, 200603 (2025).

- [16] L. Lewis, D. Zhu, A. Gheorghiu, C. Noel, O. Katz, B. Harraz, Q. Wang, A. Risinger, L. Feng, D. Biswas, L. Egan, T. Vidick, M. Cetina, and C. Monroe, Experimental implementation of an efficient test of quantumness, *Phys. Rev. A* **109**, 012610 (2024).
- [17] U. Mahadev, Classical verification of quantum computations, *arXiv:1804.01082*.
- [18] A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden, Qfactory: Classically-instructed remote secret qubits preparation, in *Advances in Cryptology—ASIACRYPT 2019* (Springer International Publishing, New York, 2019), pp. 615–645.
- [19] S. Ferracin, T. Kapourniotis, and A. Datta, Accrediting outputs of noisy intermediate-scale quantum computing devices, *New J. Phys.* **21**, 113038 (2019).
- [20] R. Raussendorf and H. J. Briegel, A one-way quantum computer, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [21] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner, Composability of delegated quantum computation, in *Advances in Cryptology—ASIACRYPT 2014* (Springer, Berlin, Heidelberg, 2014), pp. 406–425.
- [22] The decision of whether the client accepts is still allowed to depend on the resources used for the delegated computation, i.e., the graph that describes the resource state as well as the order in which the qubits are processed.
- [23] J. Frank, E. Kashefi, D. Leichtle, and M. de Oliveira, Heuristic-free verification-inspired quantum benchmarking, *Quantum Sci. Technol.* **10**, 025055 (2025).
- [24] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/jpms-v3kw> for further details on the assessment of protocol requirements, compilation to H1-1, and additional experimental results, which includes Refs. [25–42].
- [25] W. Research, Convexoptimization—wolfram language documentation, <https://reference.wolfram.com/language/ref/ConvexOptimization.html> (2024), accessed: 2024-09-13.
- [26] M. ApS, Mosek modelling cookbook (2023), available online at <https://mosek.com>.
- [27] C. J. Wood, J. D. Biamonte, and D. G. Cory, Tensor networks and graphical calculus for open quantum systems, *Quantum Inf. Comput.* **15**, 0579 (2015).
- [28] S. Olmschenk, K. C. Younge, D. L. Moehring, D. N. Matsukevich, P. Maunz, and C. Monroe, Manipulation and detection of a trapped Yb^+ hyperfine qubit, *Phys. Rev. A* **76**, 052314 (2007).
- [29] X. Mai, L. Zhang, and Y. Lu, High-fidelity detection on $^{171}\text{Yb}^+$ qubit via $^2d_{3/2}$ shelving, *arXiv:2402.18868*.
- [30] M. D. Barrett, B. DeMarco, T. Schaetz, V. Meyer, D. Leibfried, J. Britton, J. Chiaverini, W. Itano, B. Jelenković, J. Jost *et al.*, Sympathetic cooling of $^9\text{Be}^+$ and $^{24}\text{Mg}^+$ for quantum logic, *Phys. Rev. A* **68**, 042302 (2003).
- [31] J. Eschner, G. Morigi, F. Schmidt-Kaler, and R. Blatt, Laser cooling of trapped ions, *J. Opt. Soc. Am. B* **20**, 1003 (2003).
- [32] C. Monroe, D. Meekhof, B. King, S. R. Jefferts, W. M. Itano, D. J. Wineland, and P. Gould, Resolved-sideband raman cooling of a bound atom to the 3D zero-point energy, *Phys. Rev. Lett.* **75**, 4011 (1995).
- [33] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn *et al.*, A race-track trapped-ion quantum processor, *Phys. Rev. X* **13**, 041052 (2023).
- [34] J. P. Gaebler, T. R. Tan, Y. Lin, Y. Wan, R. Bowler, A. C. Keith, S. Glancy, K. Coakley, E. Knill, D. Leibfried *et al.*, High-fidelity universal gate set for $^9\text{Be}^+$ ion qubits, *Phys. Rev. Lett.* **117**, 060505 (2016).
- [35] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, High-fidelity quantum logic gates using trapped-ion hyperfine qubits, *Phys. Rev. Lett.* **117**, 060504 (2016).
- [36] J. Gaebler, C. Baldwin, S. Moses, J. Dreiling, C. Figgatt, M. Foss-Feig, D. Hayes, and J. Pino, Suppression of midcircuit measurement crosstalk errors with micromotion, *Phys. Rev. A* **104**, 062440 (2021).
- [37] A. Sørensen and K. Mølmer, Entanglement and quantum computation with ions in thermal motion, *Phys. Rev. A* **62**, 022311 (2000).
- [38] P. J. Lee, K.-A. Brickman, L. Deslauriers, P. C. Haljan, L.-M. Duan, and C. Monroe, Phase control of trapped ion quantum gates, *J. Opt. B* **7**, S371 (2005).
- [39] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient z gates for quantum computing, *Phys. Rev. A* **96**, 022330 (2017).
- [40] National Institute of Standards and Technology, Security Requirements for Cryptographic Modules (FIPS PUB 140-2), Federal Information Processing Standards Publication 140-2 (National Institute of Standards and Technology (NIST), 2001) includes Change Notices as of 12/3/2002, 10.6028/NIST.FIPS.140-2.
- [41] N. Norman, RNG-tools: A suite of random number generator related utilities, <https://github.com/nhorman/rng-tools> (2024).
- [42] Fujitsu Component Limited, Random Number Generator HM-RAE103 Datasheet, Technical Report (Fujitsu Component Limited, 2022) accessed: 2024-09-05.
- [43] Obtaining the optimal value of secret dependency remains an open problem, as the presence of the summation complicates its formulation within a convex or semidefinite optimization framework. Consequently, the value of ϵ_1 may be lower.
- [44] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, Demonstration of the trapped-ion quantum CCD computer architecture, *Nature (London)* **592**, 209 (2021).
- [45] Quantinuum, Quantinuum h1 emulator product data sheet v6.8.3, (2024), https://cdn.prod.website-files.com/62b9d45fb3f64842a96c9686/66997f9f44647b2f779fe7d1_Quantinuum%20H1%20Emulator%20Product%20Data%20Sheet%20v6.8.3%2018Jul24.pdf (accessed: 2024-09-06).
- [46] C. Ryan-Anderson, Pecos: Performance estimator of codes on surfaces, <https://github.com/PECOS-packages/PECOS> (2018).
- [47] C. Ryan-Anderson, Quantum algorithms, architecture, and error correction, Ph.D. thesis, University of New Mexico, 2018.
- [48] This is for demonstration purposes only and is not required in the protocol.
- [49] C. Gustiani, D. Leichtle, J. Miller, R. Grassie, D. Mills, and E. Kashefi, On chip verified quantum computation—Python, 2024, <https://github.com/CQCL/ocvqc-py>.

- [50] C. Gustiani and D. P. DiVincenzo, Blind oracular quantum computation, *Quantum Sci. Technol.* **6**, 045022 (2021).
- [51] G. J. Mooney, G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, Whole-device entanglement in a 65-qubit superconducting quantum computer, *Adv. Quantum Technol.* **4**, 2100061 (2021).
- [52] S. Yokoyama, R. Ukai, S. C. Armstrong, C. Sornphiphatphong, T. Kaji, S. Suzuki, J.-i. Yoshikawa, H. Yonezawa, N. C. Menicucci, and A. Furusawa, Ultra-large-scale continuous-variable cluster states multiplexed in the time domain, *Nat. Photonics* **7**, 982 (2013).
- [53] Quantinuum nexus, (2024), <https://nexus.quantinuum.com/>.
- [54] Quantinuum h1-1, (2024), <https://www.quantinuum.com/>.
- [55] C. Gustiani, D. Leichtle, J. Miller, R. Grassie, D. Mills, and E. Kashefi, On chip verified quantum computation—data, 2024, <https://github.com/CQCL/ocvqc-data>.
- [56] V. Danos and E. Kashefi, Determinism in the one-way model, *Phys. Rev. A* **74**, 052310 (2006).
- [57] C. Gustiani and D. P. DiVincenzo, Blind three-qubit exact grover search on a nitrogen-vacancy-center platform, *Phys. Rev. A* **104**, 062422 (2021).

End Matter

On-chip verification and benchmarking protocol—Our on-chip verification protocol is presented in Protocol 1, and utilizes measurement-based quantum computing. An MBQC algorithm is defined as a set $\{(G, I, O), \phi, f\}$, where (G, I, O) represents an “open graph”—a connected graph $G = (V, E)$ with vertices V , edges E , input set I , and output set O . The vertices V represent qubits and edges E correspond to entangling operations between the connected qubits. The variable ϕ denotes

the set of measurement angles assigned to each measured vertex. The function $f: O^c \rightarrow I^c$, known as the “flow” [56], is an injective mapping from the nonoutput set O^c to the noninput set I^c . This flow defines the measurement corrections required to achieve deterministic computation.

Experiment circuit—The two-qubit Grover search algorithm, implemented in MBQC, is shown in

Protocol 1. On-chip verification of BQP computations.

Parameters: Number of computation rounds d , number of test rounds t , total number of rounds $n = d + t$, flow f on graph $G = (V, E)$, and a k -coloring K of G .

Client’s inputs: Angles $\{\phi_v\}_{v \in V}$, classical input to the computation $x \in \{0, 1\}^{|I|}$ (where $|I|$ is the size of the set I of input vertices).

Protocol:

1. The client chooses uniformly at random a partition (C, T) of $[n]$ ($C \cap T = \emptyset$) with $|C| = d$, the sets of indices of the computation and test rounds respectively.
 2. For $j \in [n]$, the client and the server perform the following subprotocol (the client may send message **Redo_j** to the server before step 2.c while the server may send it to the client at any time, both parties then restart round j with fresh randomness):
 - a. If $j \in T$ (test), the client chooses uniformly at random a color $V_j \leftarrow \{V_k\}_{k \in [K]}$ (this is the set of traps for this test round).
 - b. The client instructs the server to prepare $|V|$ qubits. If $j \in T$ and the destination qubit $v \notin V_j$ is a nontrap qubit (therefore a dummy), then the client chooses uniformly at random $d_v \leftarrow \{0, 1\}$ and instructs the preparation of the state $Z^{d_v}|+\rangle$. Otherwise, the client chooses at random $\theta_v \leftarrow \Theta$ and instructs the preparation of the state $|\theta_v\rangle$.
 - c. The server performs an **XX** gate between all its qubits corresponding to an edge in the set E .
 - d. For $v \in V$, the client sends a measurement angle δ_v , the server measures the appropriate corresponding qubit in the δ_v basis of the YZ plane, returning outcome b_v to the client. The angle δ_v is defined as follows:
 - i. If $j \in C$ (computation), it is the same as in UBQC, computed using the flow and the computation angles $\{\phi_v\}_{v \in V}$. For $v \in I$ (input qubit) the client uses $\tilde{\theta}_v = \theta_v + x_v\pi$ in the computation of δ_v .
 - ii. If $j \in T$ (test): if $v \notin V_j$ (dummy qubit), the client chooses it uniformly at random from Θ ; if $v \in V_j$ (trap qubit), it chooses uniformly at random $r_v \leftarrow \{0, 1\}$ and sets $\delta_v = \theta_v + r_v\pi$.
 3. For all $j \in T$ (test round) and $v \in V_j$ (traps), the client verifies that $b_v = r_v \oplus d_v$, where $d_v = \bigoplus_{i \in N_G(v)} d_i$ is the sum over the values of neighboring dummies of qubit v . Let c_{fail} be the number of failed test rounds (where at least one trap qubit does not satisfy the relation above), if $c_{\text{fail}} \geq w$ then the client aborts by sending message **Abort** to the server.
 4. Otherwise, let y_j for $j \in C$ be the classical output of computation round j (after corrections from measurement results). The client checks whether there exists some output value y such that $|\{y_j | j \in C, y_j = y\}| > (d/2)$. If such a value y exists (this is then the majority output), it sets it as its output and sends message **Ok** to the server. Otherwise it sends message **Abort** to the server.
-

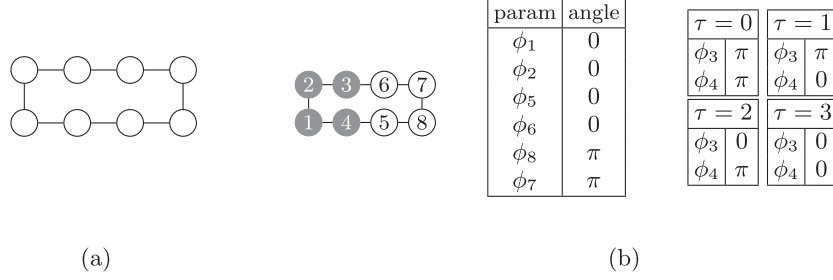


FIG. 3. Measurement-based two-qubit Grover's algorithm. (a) Verified 4×2 rectangular graph state resource. (b) MBQC specification. Gray nodes represent oracle operation, where the measurement angles depend on the queried value τ [57]. The measurement flow is linear, proceeding from the left node to the next consecutive node.

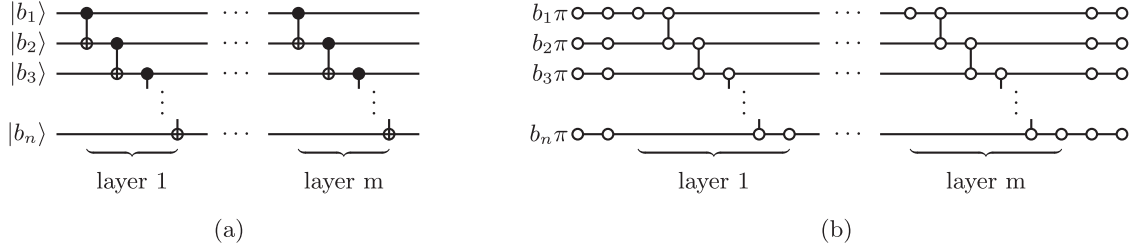


FIG. 4. CNOT grid. Equivalent circuit and MBQC model implementations of CNOT grid. (a) Circuit. Qubits are initialised in computational basis states $b \in \{0, 1\}^n$. m layers of CNOT gates act on neighbouring qubits in a line architecture. (b) Measurement pattern. Input vertices are measured by $b \in \{0, 1\}^n$. All other measurements are by 0. Measurement order is from left to right and flow is along horizontal edges. The last layer of vertices are the output.

Fig. 3(b), for all four possible queries. These patterns are used for the experiments of Fig. 2(a). CNOT grid circuits are seen in Fig. 4, and used in the experiments of

Fig. 2(b). These circuits consist of m layers of CNOT gates acting between n neighboring qubits in a line architecture.