Article

Certified randomness using a trapped-ion quantum processor

https://doi.org/10.1038/s41586-025-08737-1

Received: 27 July 2024

Accepted: 4 February 2025

Published online: 26 March 2025

Open access

Check for updates

Minzhao Liu[1,2,3,11], Ruslan Shaydulin[1,11]✉, Pradeep Niroula[1,11], Matthew DeCross[4], Shih-Han Hung[5,6], Wen Yu Kon[1], Enrique Cervero-Martín[1], Kaushik Chakraborty[1], Omar Amer[1], Scott Aaronson[5], Atithi Acharya[1], Yuri Alexeev[2,10], K. Jordan Berg[4], Shouvanik Chakrabarti[1], Florian J. Curchod[7], Joan M. Dreiling[4], Neal Erickson[4], Cameron Foltz[4], Michael Foss-Feig[4], David Hayes[4], Travis S. Humble[8], Niraj Kumar[1], Jeffrey Larson[9], Danylo Lykov[1,2,10], Michael Mills[4], Steven A. Moses[4], Brian Neyenhuis[4], Shaltiel Eloul[1], Peter Siegfried[4], James Walker[4], Charles Lim[1]✉ & Marco Pistoia[1]✉

Although quantum computers can perform a wide range of practically important tasks beyond the abilities of classical computers[1,2], realizing this potential remains a challenge. An example is to use an untrusted remote device to generate random bits that can be certified to contain a certain amount of entropy[3]. Certified randomness has many applications but is impossible to achieve solely by classical computation. Here we demonstrate the generation of certifiably random bits using the 56-qubit Quantinuum H2-1 trapped-ion quantum computer accessed over the Internet. Our protocol leverages the classical hardness of recent random circuit sampling demonstrations[4,5]: a client generates quantum 'challenge' circuits using a small randomness seed, sends them to an untrusted quantum server to execute and verifies the results of the server. We analyse the security of our protocol against a restricted class of realistic near-term adversaries. Using classical verification with measured combined sustained performance of 1.1×10^{18} floating-point operations per second across multiple supercomputers, we certify 71,313 bits of entropy under this restricted adversary and additional assumptions. Our results demonstrate a step towards the practical applicability of present-day quantum computers.

In recent years, numerous theoretical results have shown evidence that quantum computers have the potential to tackle a wide range of problems out of reach of classical techniques. The main examples include factoring large integers[6], implicitly solving exponentially sized systems of linear equations[7], optimizing intractable problems[8], learning certain functions[9] and simulating large quantum many-body systems[10]. However, accounting for considerations such as quantum error correction overheads and gate speeds, the resource requirements of known quantum algorithms for these problems put them far outside the reach of near-term quantum devices, including many suggested fault-tolerant architectures. Consequently, it is unclear whether the devices available in the near term can benefit a practical application[11].

Starting with one of the first 'quantum supremacy' demonstrations[5], several groups have used random circuit sampling (RCS) as an example of a task that can be executed faster and with a lower energy cost on present-day quantum computers compared with what is achievable classically[4,12–14]. Yet, despite rapid experimental progress, a beyond-classical demonstration of a practically useful task performed by gate-based quantum computers has so far remained unknown.

Random number generation is a natural task for the beyond-classical demonstration because randomness is intrinsic to quantum mechanics, and it is important in many applications, ranging from information security to ensuring the fairness of processes such as jury selection[15–17]. The main challenge for any client receiving randomness from a third-party provider, such as a hardware security module, is to verify that the bits received are truly random and freshly generated. Although certified randomness is not necessary for every use of random numbers, the freshness requirement is especially important in applications such as lotteries and e-games, in which several parties (which may or may not trust each other) need to ensure that a publicly distributed random number was generated on demand. Moreover, certified randomness can be used to verify the position of a dishonest party[18–20].

Protocols exist for certifying random numbers based on the violation of Bell inequalities[15,21–24]. However, these protocols typically require the underlying Bell test to be loophole-free, which can be hard for the client to enforce when the quantum devices are controlled by a third-party provider. This approach thus necessitates that the client trust a third-party quantum device provider to perform the Bell test faithfully.

[1]Global Technology Applied Research, JPMorganChase, New York, NY, USA. [2]Computational Science Division, Argonne National Laboratory, Lemont, IL, USA. [3]Department of Physics, The University of Chicago, Chicago, IL, USA. [4]Quantinuum, Broomfield, CO, USA. [5]Department of Computer Science, The University of Texas at Austin, Austin, TX, USA. [6]Department of Electrical Engineering, National Taiwan University, Taipei City, Republic of China. [7]Quantinuum, Terrington House, Cambridge, UK. [8]Quantum Science Center, Oak Ridge National Laboratory, Oak Ridge, TN, USA. [9]Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, USA. [10]Present address: NIVIDA Corporation, Santa Clara, CA, USA. [11]These authors contributed equally: Minzhao Liu, Ruslan Shaydulin, Pradeep Niroula. ✉e-mail: ruslan.shaydulin@jpmorgan.com; charles.lim@jpmorgan.com; marco.pistoia@jpmorgan.com

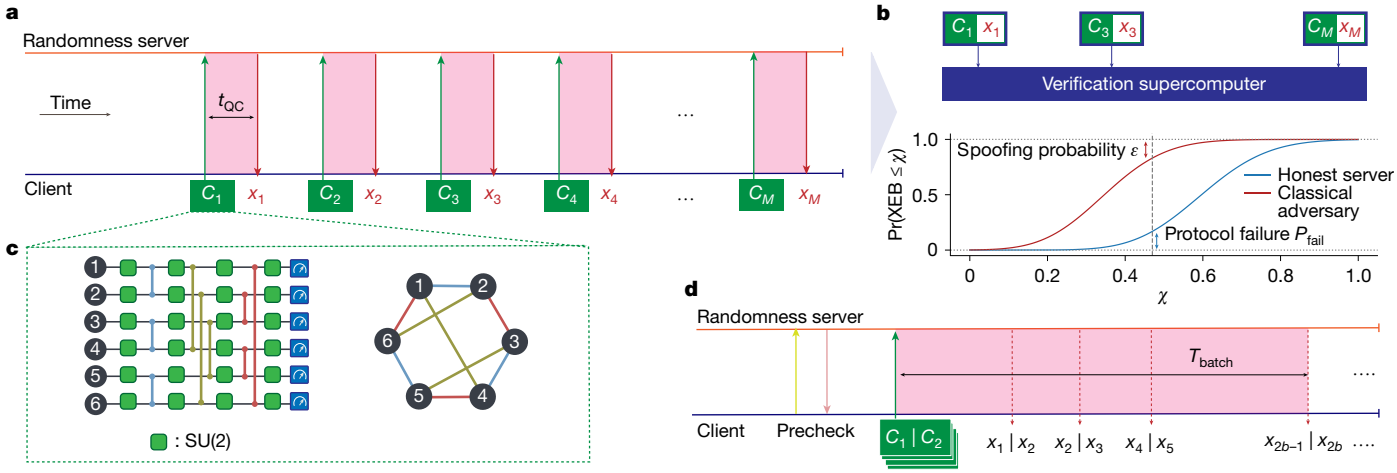

Fig. 1 | Overview of the protocol. a, The idealized protocol. A client submits M random circuits $\{C_i\}_{i\in[M]}$ serially to a randomness server and expects bitstrings $\{x_i\}_{i\in[M]}$ back, each within a time t_{QC}. **b**, A subset of circuit-bitstring pairs is used to compute the XEB score. The XEB score has distributions (bottom plot for qualitative illustration only) corresponding to either an honest server or an adversarial server performing a low-fidelity classical simulation. For any XEB target indicated by the dashed line, an honest server may fail to achieve a score above this threshold with a probability P_{fail}. **c**, Illustration of the challenge circuits, consisting of layers of U_{ZZ} gates sandwiched between layers of random SU(2) gates on all qubits. The arrangement of two-qubit gates is obtained via edge colouring (right) on a random n-node graph. **d**, Client-server interaction as implemented in our protocol. Following a device-readiness check ('precheck'), the client submits a batch of $2b$ circuits and expects all the samples corresponding to the batch to be returned within a cutoff duration $T_{b,cutoff}$. Note that only one batch with execution time T_{batch} is illustrated in the figure. The client continues the protocol until M total circuits have been successfully executed.

Alternatively, ref. 3 proposed a certified randomness protocol that combines RCS with 'verification' on classical supercomputers[3,25]. This type of protocol allows a classical client to verify randomness using only remote access to an untrusted quantum server. A classical client pseudorandomly generates n-qubit challenge circuits and sends them to a quantum server, which is asked to return length-n bitstrings sampled from the output distribution of these circuits within a short amount of time (Fig. 1a,c). The circuits are chosen such that no realistic adversarial server can classically simulate them within the short response time. A small subset of circuits is then used to compute the cross-entropy benchmarking (XEB) score[26] (Fig. 1b), which reflects how well the samples returned by the server match the ideal output distributions of the submitted circuits. Extensive complexity-theoretic evidence suggests that XEB is hard to 'spoof' classically[27,28]. Therefore, a high XEB score, combined with a short response time, allows the client to certify that the server must have used a quantum computer to generate its responses, thereby guaranteeing a certain amount of entropy with high probability. Our analysis quantifies the minimum amount of entropy that an untrusted server, possibly acting as an adversary, must provide to achieve a given XEB score in a short amount of time.

The protocol proposed in ref. 3 provides a complexity-theoretic guarantee of $\Omega(n)$ bits of entropy for a server returning many samples from the same circuit. This protocol is best suited for quantum computing architectures with overheads that make it preferable to sample a circuit many times after loading it once. In practice, the classical simulation cost of sampling a circuit many times is comparable to the cost of sampling only once[29]. Furthermore, the trapped-ion-based quantum computer used in this work is configured to feature minimal overhead per circuit, such that executing many single-shot circuits does not introduce a substantial time penalty per circuit compared with sampling one circuit many times. Together, these two observations motivate strengthening the security of the protocol by requesting the server to return only one sample per circuit. To this end, in Supplementary Information section I, we extend the complexity-theoretic analysis to this modified setting of one sample per circuit, guaranteeing $\Omega(n)$ bits of entropy.

In this work, we report an experimental demonstration of an RCS-based certified randomness protocol. Our main contributions are as follows. First, inspired by ref. 3, we propose a modified RCS-based certified randomness protocol that is tailored to near-term quantum servers. Second, we prove the security of our implementation against a class of realistic finite-sized adversaries. Third, we use a high-fidelity quantum computer and exascale classical computation to experimentally realize this proposed protocol, pushing the boundaries of both quantum and classical computing abilities. By combining the high-fidelity Quantinuum H2-1 quantum processor with exascale verification, we demonstrate a useful beyond-classical application of gate-based digital quantum computers.

In our proposed protocol, shown in Fig. 1d and detailed in the Methods, the client pseudorandomly generates a sufficiently large number of n-qubit quantum circuits and then sends them in batches of $2b$ circuits, where b is an integer. After a batch is submitted, the client waits for $2b$ length-n bitstrings to be returned within $T_{b,cutoff}$ seconds. The batch cutoff time prevents the protocol from stalling and is fixed in advance based on preliminary experiments to a value intended to maximize the amount of certifiable entropy while ensuring that the average response time per circuit remains low enough to preclude classical simulation as a viable strategy for the server to generate responses. If a batch times out or if a failure status is reported, all of the outstanding jobs in the batch are cancelled, and all bitstrings received from the batch are discarded. Consequently, results from a failed batch are not included in calculating the XEB score or entropy extraction. Batches are continually submitted until M valid samples are collected. The cumulative response time for successful batches gives the total time T_{tot} and the average time per sample $t_{QC} = T_{tot}/M$. Subsequently, the client calculates the XEB score on a subset of size m randomly sampled from the M circuit–sample pairs:

$$\mathrm{XEB}_{\mathrm{test}} = \frac{2^n}{m}\sum_{i\in\mathcal{V}} P_{C_i}(x_i) - 1, \tag{1}$$

where $\mathcal{V}$ is the set of indices for the random subset of size m and $P_C(x) = |\langle x|C|0\rangle|^2$ is the probability of measuring bitstring x from an ideal quantum computer executing circuit C. If the bitstrings x_i are perfectly drawn from the output distributions of sufficiently deep random circuits C_i, the XEB score is expected to concentrate around 1. By contrast,

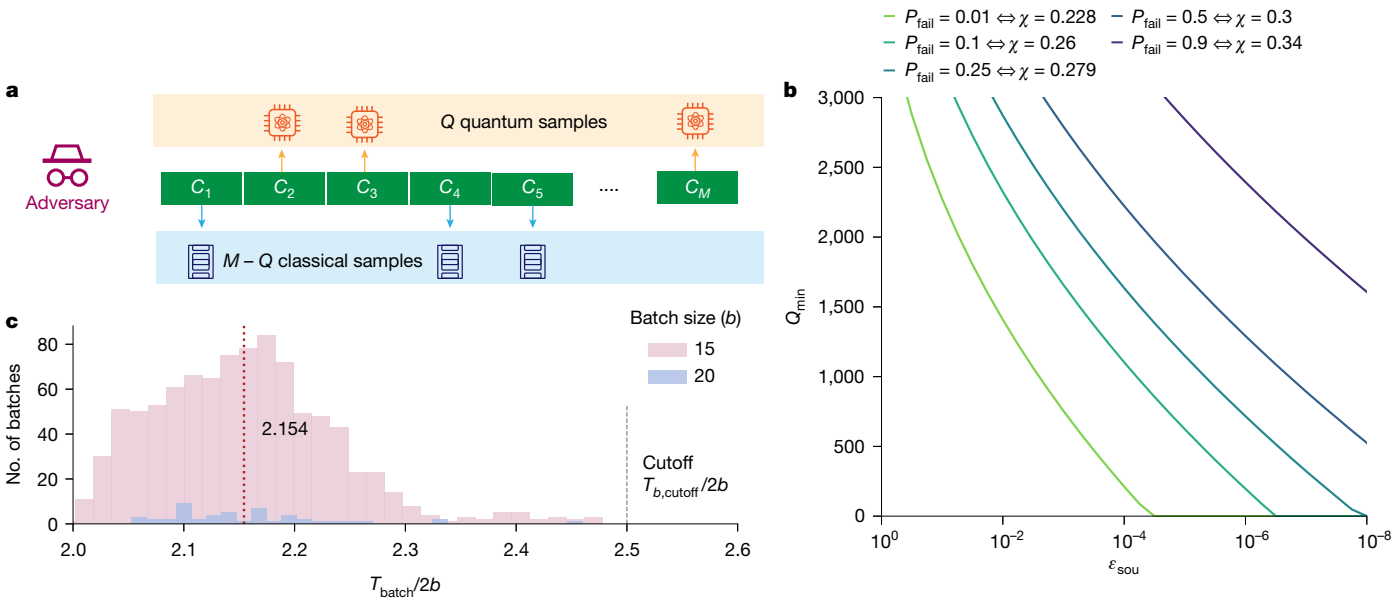

Fig. 2 | Adversary model and protocol security. a, In the adversarial model considered in this work, Q samples are obtained using a perfect-fidelity quantum computer and $M - Q$ using classical simulation. **b**, Probability of an honest server with fidelity $\phi = 0.3$ failing to certify Q_{min} quantum samples (and corresponding threshold χ) with soundness ε_{sou} against an adversary four times more powerful than Frontier over repeated experiments, with the protocol parameters set to those from Table 1. **c**, Distribution of batch times per successful sample, from a total of 984 successful batches, in our experiment. The vertical dashed line indicates the average time per sample.

if the x_i are drawn from distributions uncorrelated with the distributions induced by C_i, the XEB score is expected to concentrate around 0. The client decides to accept the received samples as random bits based on two criteria. First, the average time per sample must be lower than a threshold $t_{\text{threshold}}$, which is chosen to preclude high-fidelity classical simulation. This time can be lower than $T_{b,\text{cutoff}}$ because it is advantageous from the perspective of extractable entropy to accept some samples with response time slightly larger than $t_{\text{threshold}}$ as long as the average response time remains low. Second, the XEB score on $\mathcal{V}$ must be greater than a threshold $\chi \in [0, 1]$. All of $t_{\text{threshold}}$, χ and $T_{b,\text{cutoff}}$ are determined in advance of protocol execution, based on (for example) preliminary hardware experiments, with the goal of certifying a certain fixed amount of entropy at the end of the protocol with high probability. Together, the protocol succeeds if

$$t_{\text{QC}} = T_{\text{tot}}/M \leq t_{\text{threshold}} \quad \text{and} \quad \text{XEB}_{\text{test}} \geq \chi, \tag{2}$$

and otherwise aborts.

The security of our protocol relies on the central assumption that, for the family of pseudorandom circuits we consider, there exists no practical classical algorithm that can spoof the XEB test used in the protocol. We analyse the protocol security by modelling a restricted but realistic adversarial server that we believe to be the most relevant: for each circuit received, the adversary either samples an output honestly from a quantum computer or performs classical simulation (Fig. 2a). As only the former contains entropy, the adversary tries to achieve the threshold XEB score with the fewest quantum samples, to pass the XEB test while returning as little entropy as possible. For our protocol, we assume an adversary with a perfect-fidelity quantum computer, which allows the adversary to spoof the maximum number of bitstrings classically. We further assume that the classical computational power of the adversary is bounded by a fixed number of floating-point operations per second (FLOPS) $\mathcal{A}$, which may be measured relative to the most powerful supercomputer in the world (at the time of experiment, the Frontier supercomputer; see https://www.top500.org/lists/top500/2024/06/), and that the adversary possesses the same optimized methods to simulate the circuits as the client has. Note that an adversary possessing more powerful classical methods for simulating circuits than expected can equivalently be modelled as an adversary with identical classical methods and larger computational power. We note that as the adversaries we analyse are allowed only a restricted set of strategies, the subsequent mathematical results hold only in this limited setting, conditioned on some additional assumptions further detailed in Supplementary Information section IIIC. To the best of our knowledge, the restricted set of classical and quantum adversary strategies considered here correspond to the current state of the art. We leave the incorporation of a broader class of adversaries to future analysis.

The client needs to ensure that the circuits are difficult to simulate within the time $t_{\text{threshold}}$. Otherwise, the server can use its classical supercomputer to deterministically simulate the circuits with high fidelity and generate samples that readily pass the tests in equation (2). For the family and size of circuits we consider, tensor network contraction is the most performant known method for finite-fidelity and exact simulation[4] as well as sampling. If a circuit has a verification (exact simulation) cost of $\mathcal{B}$ FLOPS, the adversary can simulate each circuit to a target fidelity of $\mathcal{A} \cdot t_{\text{threshold}}/\mathcal{B}$ using partial contraction of tensor networks, for which the simulation cost and simulation fidelity are related linearly[30]. The protocol is successful only if the parameters are chosen such that the fidelity ϕ of an honest server satisfies

$$\phi \gg \mathcal{A} \cdot t_{\text{threshold}}/\mathcal{B}. \tag{3}$$

This condition requires that there exists a gap between the fidelity of an honest server and that achievable by an adversary performing mostly classical simulations. If this condition is satisfied, the XEB score of an honest server will have a probability distribution with a higher average value than the probability distribution of the XEB of the adversary (qualitatively shown in Fig. 1b), allowing the client to distinguish between the two.

After certification (that is, if the tests in equation (2) pass), the client uses a randomness extractor to process the M samples. An ideal protocol for certified randomness either aborts, resulting in an 'abort state', or succeeds, resulting in a uniformly distributed bitstring that

Table 1 | Summary of experimental parameters

Label	Meaning	Value
n	Number of qubits	56
$\mathcal{B}$	Cost of simulating challenge circuits	90×10^{18} FLOPS
$\mathcal{A}$	Sustained peak performance of the Frontier supercomputer	0.897×10^{18} FLOPS
–	Time to simulate challenge circuits on the Frontier supercomputer	100.3 s
χ	Threshold for XEB test	0.3
$t_{\text{threshold}}$	Threshold for average time per sample	2.2 s
$T_{b,\text{cutoff}}$	Cutoff time for the server to respond to a batch of $2b$ circuits	$2.5 \times 2b$ s
M	Number of successful samples	30,010
t_{QC}	Average response time per successful quantum sample	2.154 s
m	Number of samples used to measure XEB	1,522
XEB_{test}	Measured XEB	0.32

is uncorrelated with any side information. Viewing the protocol as a channel acting on some initial state composed of both the server and the client, an end-to-end protocol is said to be ε_{sou}-sound if, for any initial state, the end result is ε_{sou}-close (in terms of trace distance) to the ideal output: a mixture of the abort state and the maximally mixed state (see Supplementary Information section IIIA for the rigorous definition of soundness).

The entropy that the client can extract out of the received samples on successful execution of the protocol depends on how stringent its thresholds on the response time ($t_{\text{threshold}}$) and the XEB score (χ) are. It is in the interest of the client to set these thresholds as stringently as possible, to force the hypothetical adversary to draw more samples from the quantum computer, while still allowing that an honest server can succeed with high probability. As the thresholds are known to both parties, the strategy of the adversary is to minimize the use of the quantum computer while ensuring that the protocol does not abort. Based on the protocol thresholds, the client can determine the number of quantum samples Q_{min} such that the protocol aborts with a large probability $1 - \varepsilon_{\text{accept}}$ if the adversary returns fewer than Q_{min} samples from the quantum computer (see Supplementary Information section IVF for details). This lower bound on Q_{min} can be used to derive the minimum smooth min-entropy of the received samples. Note that the smooth min-entropy of an information source characterizes the number of random bits that can be extracted from the source. In particular, we devise an ε_{sou}-sound protocol that provides a lower bound on the smooth min-entropy $H_{\text{min}}^{\varepsilon_s}$ (defined in Supplementary Information section IIID) with smoothness parameter $\varepsilon_s = \varepsilon_{\text{sou}}/4$ and with $\varepsilon_{\text{accept}} = \varepsilon_{\text{sou}}$. The results in the paper are reported in terms of the soundness parameter ε_{sou} and the smooth min-entropy $H_{\text{min}}^{\varepsilon_s}$.

A smaller ε_{sou} makes a stronger security guarantee by making it more difficult for an adversary to pass the XEB test with a small Q_{min}. This may be achieved by choosing a higher threshold χ. However, a higher threshold also makes it more likely for an honest server to fail the XEB test, meaning that the honest server cannot be certified to have produced the target amount of extractable entropy. Note that this does not necessarily mean that the samples provided by the honest server do not contain entropy, only that they fail to satisfy the criteria of equation (2) and consequently the protocol aborts. In practice, it is desirable to ensure that an honest server fails only with a low failure probability P_{fail}. To that end, we may compute a threshold $\chi(P_{\text{fail}})$ corresponding to any acceptable P_{fail}. This threshold, along with $t_{\text{threshold}}$, then allows us to determine Q_{min} for a target soundness ε_{sou} (Supplementary Information section IIID). Figure 2b shows the achievable Q_{min} at different P_{fail} and ε_{sou}, showing the trade-off between the three quantities at the fixed experimental configuration and the classical computational power of adversary (ϕ, t_{QC}, M, m, $\mathcal{B}$ and $\mathcal{A}$).

We demonstrate our protocol using the Quantinuum H2-1 trapped-ion quantum processor accessed remotely over the Internet. The experimental parameters are provided in Table 1. The challenge circuits (shown in Fig. 1c, see Supplementary Information section IVC for the considerations involved in choosing the circuits) have a fixed arrangement of 10 layers of entangling U_{ZZ} gates, each sandwiched between layers of pseudorandomly generated $SU(2)$ gates on all qubits. The arrangement of two-qubit gates is obtained by edge colouring on a random n-node graph. Preliminary mirror-benchmarking experiments, along with gate-counting arguments based on the measured fidelities of component operations, enable us to estimate the fidelity of an honest server[4]. At the time of the experiment, the H2-1 quantum processor was expected to attain a fidelity of $\phi \gtrsim 0.3$ or better on depth-10 circuits (multiple improvements were made to the H2-1 device after the collection of the data of this experiment that slightly increased the fidelity estimate in ref. 4). Likewise, the same preliminary experiments also let us anticipate average time per sample to be approximately 2.1 s, with a long-tailed timing distribution out to just below 2.5 s, as also seen in the full experiment in Fig. 2c. Reasonable ($P_{\text{fail}} = 50\%$) protocol success rates can therefore be achieved with thresholds $t_{\text{threshold}} = 2.2$ s and $\chi = 0.3$. For illustrative purposes, we describe the experiment based on these choices (in practice, one might want to lower P_{fail} by setting χ somewhat below the expected value). The batch cutoff time is set to be $T_{b,\text{cutoff}} = (2b) \times 2.5$ s, anticipating that the relatively small expected fraction of batches taking average time per sample between $t_{\text{threshold}} = 2.2$ s and 2.5 s would contribute additional entropy to the received samples while being unlikely to increase the average time per sample from the expected 2.1 s past the threshold of 2.2 s.

The circuit family considered has a simulation cost of $\mathcal{B} = 90 \times 10^{18}$ FLOPS on the Frontier supercomputer of the Department of Energy[31], the most powerful supercomputer in the world, to our knowledge, at the time of writing (https://www.top500.org/lists/top500/2024/06/). Following a detailed estimate of runtime on Frontier, we determine an exact simulation time of 100.3 s per circuit when using the entire supercomputer at a numerical efficiency of 45%, where numerical efficiency is the ratio between the actual algorithm runtime and its theoretical expectation (see Supplementary Information section IVA for details on the circuit simulation cost).

In our experiment, we use two batch sizes, $b = 15$ and $b = 20$; most of the batches have $b = 15$. In total, we submitted 1,993 batches for a total of 60,952 circuits. From those, we obtain a total of $M = 30{,}010$ valid samples out of 984 successful batches. The cumulative device time of the successful samples was 64,652 s, giving an average time of $t_{\text{QC}} = 2.154$ s per sample, inclusive of all overheads such as communication time. Figure 2c shows the distribution of t_{QC} per successful sample.

In this work, the classical computational budget of the client is spread across the Frontier[31], Summit[32], Perlmutter[33] and Polaris[34] supercomputers equipped with graphics processing units (GPUs), which are especially suitable for quantum circuit simulations. Of the four supercomputers, Frontier and Summit were used at full-machine scale during verification. We measure the sustained peak performance of 897 petaFLOPS and 228 petaFLOPS, respectively (corresponding to numerical efficiencies of 45% and 59%), achieving a combined performance of 1.1 exaFLOPS (see Supplementary Information section IVE). We compute the XEB score for $m = 1{,}522$ circuit–sample pairs, obtaining $\text{XEB}_{\text{test}} = 0.32$. The complete set of experimental parameters is listed in Table 1.

The measured fidelity of $\text{XEB}_{\text{test}} = 0.32$ and measured time per sample $t_{\text{QC}} = 2.154$ s pass the protocol specified by $\chi = 0.3$ and $t_{\text{threshold}} = 2.2$ s. For a choice of soundness parameter ε_{sou} and a smoothness parameter $\varepsilon_s = \varepsilon_{\text{sou}}/4$, the protocol thresholds determine the number of quantum samples Q and the smooth min-entropy $H_{\text{min}}^{\varepsilon_s}$ guaranteed by the success of this protocol against an adversary with classical resources

Table 2 | Smooth min-entropy rate at varying ε_{sou} and $\mathcal{A}$

	$\mathcal{A}$ (multiples of Frontier)				
ε_{sou}	1	2	4	6	8
10^{-2}	0.19	0.16	0.11	0.06	0.01
10^{-4}	0.15	0.12	0.07	0.02	0.00
10^{-6}	0.12	0.09	**0.04**	0.00	0.00
10^{-8}	0.10	0.07	0.02	0.00	0.00
10^{-10}	0.08	0.05	0.00	0.00	0.00

The adversary is assumed to have the same efficiency for classical simulation as client verification. The ratio corresponding to the entropy we report in the main text is boldfaced.

bounded by $\mathcal{A}$. In Table 2, we report the smooth min-entropy rate, $H_{\min}^{\varepsilon_s}/(56 \times M)$, for a range of $\mathcal{A}$ and ε_{sou} (see Supplementary Information section IVF for details of this calculation). This is to show that if we want to increase the security of the protocol either by increasing the assumed classical computational power of the adversary or by reducing the soundness parameter, the amount of entropy that we can obtain must reduce. In particular, we highlight that at $\varepsilon_{sou} = 10^{-6}$, we have $Q_{min} = 1{,}297$, corresponding to $H_{\min}^{\varepsilon_s} = 71{,}313$ against an adversary four times more powerful than Frontier (under the assumptions discussed earlier).

We feed the 56 × 30,010 raw bits into a Toeplitz randomness extractor[35] and extract 71,273 bits (see Supplementary Information section IVF for details on extraction and the determination of extractable entropy). We note that the Toeplitz extractor is a 'strong' seeded extractor for which the output is independent of the seed. For private use of the randomness, in which the extracted bits are not shown, the extractor seed can be reused. We append the seed used in the extractor to the protocol output and do not count the seed as randomness 'consumed' by our protocol. The total input randomness used to seed the pseudorandom generator is thereby only 32 bits, and our protocol achieves certified randomness expansion. We further note that other extractors can be used that may consume less seed but have different security guarantees.

Future experiments are expected to improve device fidelity (higher ϕ) and execution speed (lower t_{QC}). Adjusting protocol thresholds (χ and $t_{threshold}$) against improved device specifications stands to improve our protocol in terms of the achievable entropy, the adversarial computational power that can be guarded against and the soundness parameter. Figure 3 shows these metrics as we improve t_{QC} and ϕ (see Supplementary Information section V for details of this calculation). Conversely, for a fixed adversary and soundness parameter, any improvement in t_{QC} and ϕ reduces the verification budget required to certify a target number of quantum samples Q, making our protocol more cost-effective. Any improvement in entropy, all else being equal, translates into a higher throughput in the sense of a higher rate of entropy generation per second. With $\chi = 0.3$ and $t_{threshold} = 2.2$ s, our experiment has a bitrate of 71,273/(30,010 × 2.2 s) ≈ 1 bit per second at $\varepsilon_{sou} = 10^{-6}$. For $\varepsilon_{sou} = 10^{-6}$ and $P_{fail} = 0.1$, improving fidelity to $\phi = 0.67$ and response time to $t_{QC} = 0.55$ s would let us achieve the bitrate of the NIST Public Randomness beacon[36] (512 bits per minute). We note that improvement in t_{QC} can come from higher clock rates as well as parallelization over multiple quantum processors or over many qubits of one large quantum processor.

The security of our protocol relies on the circuits being difficult to simulate. When better exact simulation techniques are developed by researchers in the future, both the adversary and the client can use the improved techniques to spoof and verify: these symmetric gains neutralize each other. Although a notable improvement in approximate simulation techniques may benefit spoofing asymmetrically, the client might be able to neutralize those gains by modifying the ensemble of challenge circuits to make approximate simulations more difficult.

In summary, this work implements a protocol for certified randomness, which also lends itself to multiparty and public verification. We note that the bit rate and soundness parameter achieved by our experiment, the restricted adversarial model, as well as the numerous assumptions used in our analysis limit the immediate deployment of the proposed protocol in production applications. However, we numerically analyse how future developments may improve the security and cost-effectiveness of our protocol. Our experiments pave the way for new opportunities in cryptography and communication.

Disclaimer

This paper was prepared for informational purposes with contributions from the Global Technology Applied Research Center of JPMorgan Chase. This paper is not a product of the Research Department of JPMorgan Chase or its affiliates. Neither JPMorgan Chase nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

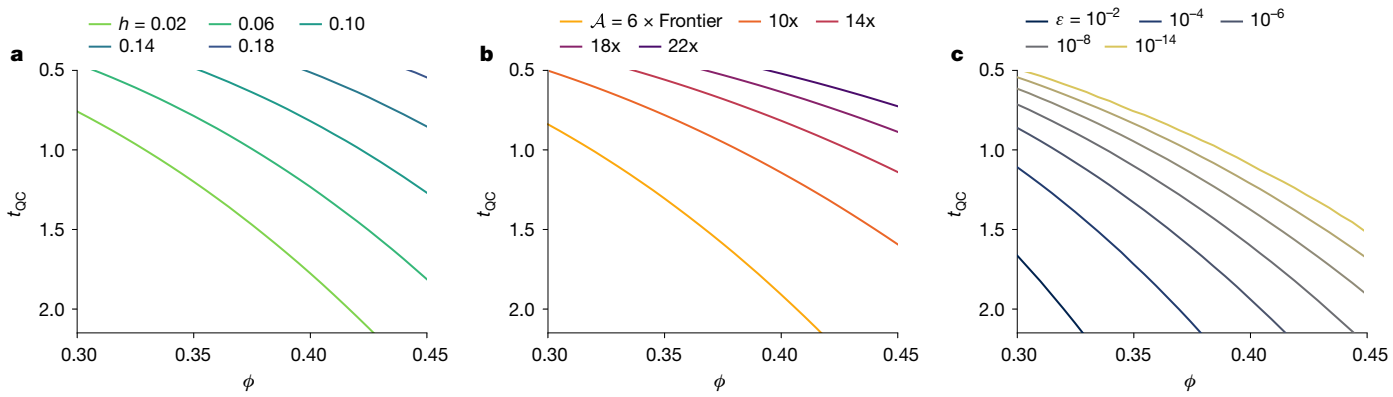

Fig. 3 | Future improvements. Improvement in metrics as fidelity ϕ and time per sample t_{QC} improve. All panels assume the same verification budget as this experiment, classical simulation numerical efficiency of 50% for both verification and spoofing, and target failure probability $P_{fail} = 10^{-4}$. **a**, Smooth min-entropy rate, $h = H_{\min}^{\varepsilon_s}/(M \cdot n)$, against an adversary four times as powerful as Frontier with $\varepsilon_{sou} = 10^{-6}$ and $\varepsilon_s = \varepsilon_{sou}/4$. **b**, Adversarial power that still allows $h = 0.01$ to be guaranteed with $\varepsilon_{sou} = 10^{-6}$. **c**, Soundness parameter ε_{sou} that still allows $h = 0.01$ to be guaranteed with an adversary that is four times as powerful as Frontier.

The submitted manuscript includes contributions from UChicago Argonne, Operator of Argonne National Laboratory ('Argonne'). Argonne, a US Department of Energy Office of Science laboratory, is operated under contract no. DE-AC02-06CH11357. The US government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide licence in said Article to reproduce, prepare derivative works, distribute copies to the public and perform publicly and display publicly, by or on behalf of the government. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan http://energy.gov/downloads/doe-public-access-plan.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-025-08737-1.

1. Alexeev, Y. et al. Quantum computer systems for scientific discovery. *PRX Quantum* **2**, 017001 (2021).
2. Herman, D. et al. Quantum computing for finance. *Nat. Rev. Phys.* **5**, 450–465 (2023).
3. Aaronson, S. & Hung, S.-H. Certified randomness from quantum supremacy. In *Proc. 55th Annual ACM Symposium on Theory of Computing* 933–944 (ACM, 2023).
4. DeCross, M. et al. The computational power of random quantum circuits in arbitrary geometries. Preprint at arxiv.org/abs/2406.02501 (2024).
5. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).
6. Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. 35th Annual Symposium on Foundations of Computer Science* 124–134 (IEEE, 1994).
7. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
8. Shaydulin, R. et al. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *Sci. Adv.* **10**, eadm6761 (2024).
9. Liu, Y., Arunachalam, S. & Temme, K. A rigorous and robust quantum speed-up in supervised machine learning. *Nat. Phys.* **17**, 1–5 (2021).
10. Berry, D. W., Ahokas, G., Cleve, R. & Sanders, B. C. Efficient quantum algorithms for simulating sparse Hamiltonians. *Commun. Math. Phys.* **270**, 359–371 (2007).
11. Hoefler, T., Häner, T. & Troyer, M. Disentangling hype from practicality: On realistically achieving quantum advantage. *Commun. ACM* **66**, 82–87 (2023).
12. Wu, Y. et al. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* **127**, 180501 (2021).
13. Zhu, Q. et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Sci. Bull.* **67**, 240–245 (2022).
14. Morvan, A. et al. Phase transitions in random circuit sampling. *Nature* **634**, 328–333 (2024).
15. Acín, A. & Masanes, L. Certified randomness in quantum physics. *Nature* **540**, 213–219 (2016).
16. Herrero-Collantes, M. & Garcia-Escartin, J. C. Quantum random number generators. *Rev. Mod. Phys.* **89**, 015004 (2017).
17. Mannalatha, V., Mishra, S. & Pathak, A. A comprehensive review of quantum random number generators: concepts, classification and the origin of randomness. *Quantum Inf. Process.* **22**, 439 (2023).
18. Brakerski, Z., Christiano, P., Mahadev, U., Vazirani, U. & Vidick, T. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *Proc. 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)* 320–331 (IEEE, 2018).
19. Liu, J., Liu, Q. & Qian, L. Beating classical impossibility of position verification. In *Proc. 13th Innovations in Theoretical Computer Science Conference (ITCS 2022)* (ed. Braverman, M.) 100:1–100:11 (Dagstuhl Publishing, 2022).
20. Amer, O. et al. Certified randomness implies secure classical position-verification. Preprint at arxiv.org/abs/2410.03982 (2024).
21. Pironio, S. et al. Random numbers certified by Bell's theorem. *Nature* **464**, 1021–1024 (2010).
22. Liu, Y. et al. Device-independent quantum random-number generation. *Nature* **562**, 548–551 (2018).
23. Foreman, C., Wright, S., Edgington, A., Berta, M. & Curchod, F. J. Practical randomness amplification and privatisation with implementations on quantum computers. *Quantum* **7**, 969 (2023).
24. Bierhorst, P. et al. Experimentally generated randomness certified by the impossibility of superluminal signals. *Nature* **556**, 223–226 (2018).
25. Bassirian, R., Bouland, A., Fefferman, B., Gunn, S. & Tal, A. On certified randomness from quantum advantage experiments. Preprint at arxiv.org/abs/2111.14846 (2021).
26. Boixo, S. et al. Characterizing quantum supremacy in near-term devices. *Nat. Phys.* **14**, 595–600 (2018).
27. Aaronson, S. & Chen, L. Complexity-theoretic foundations of quantum supremacy experiments. In *Proc. 32nd Computational Complexity Conference* 1–67 (ACM, 2017).
28. Aaronson, S. & Gunn, S. On the classical hardness of spoofing linear cross-entropy benchmarking. *Theory Comput.* **16**, 1–8 (2020).
29. Liu, Y. et al. Verifying quantum advantage experiments with multiple amplitude tensor network contraction. *Phys. Rev. Lett.* **132**, 030601 (2024).
30. Markov, I. L., Fatima, A., Isakov, S. V. & Boixo, S. Quantum supremacy is both closer and farther than it appears. Preprint at arxiv.org/abs/1807.10749 (2018).
31. Frontier user guide. *OLCF* https://docs.olcf.ornl.gov/systems/frontier_user_guide.html (2025).
32. Summit user guide. *OLCF* https://docs.olcf.ornl.gov/systems/summit_user_guide.html (2025).
33. Perlmutter architecture. *NERSC Documentation* https://docs.nersc.gov/systems/perlmutter/architecture/ (2025).
34. Polaris machine overview. *ALCF* https://www.alcf.anl.gov/support-center/training/polaris-overview-0 (2025).
35. Foreman, C., Yeung, R., Edgington, A. & Curchod, F. J. Cryptomite: a versatile and user-friendly library of randomness extractors. *Quantum* **9**, 1584 (2025).
36. Kelsey, J., Brandão, L. T. A. N., Peralta, R. & Booth, H. A reference for randomness beacons: format and protocol version 2. *NIST Technical Report* Report No. NISTIR 8213 (NIST, 2019).

Methods

The goal of the certified randomness protocol is to achieve two properties:

1. Randomness certification: outputs generated by the protocol should be close to unpredictable and uniformly distributed, uncorrelated with any side information the client, server, and the environment might possess.
2. Randomness expansion: the entropy the client certifies in the protocol should be larger than the entropy it consumes in generating the circuits and selecting the set for validation.

The M bitstrings received from the server, which we denote as X^M, do not directly satisfy the randomness certification requirement as they are not uniformly distributed. They are passed to a randomness extractor Ext along with an extractor seed K_{ext}, which is private to the client, to obtain the final output bits K that are uniformly distributed along with some side information. The possible side information we consider is any classical information possessed by the client, the server and the environment before the start of the protocol, and we denote this 'snapshot' of initial classical information as I_{sn}. This snapshot includes any initial randomness possessed by the client or the server.

An ideal randomness certification protocol outputs a string of bits (in the register K) that is uniformly random and independent of I_{sn}. That is to say, the ideal output of a successful randomness certification protocol is precisely $\tau_K \otimes \rho_{I_{\text{sn}}}$, where τ_K is a maximally mixed state and $\rho_{I_{\text{sn}}}$ is the quantum state representing any side information. If the protocol aborts, the output is expected to be abort state. We quantify the security or soundness of our protocol by the closeness (as given by a trace distance) between the ideal output and the actual output produced by the protocol. As a lower bound to the smooth min-entropy of the M raw samples returned by the server suffices to guarantee soundness by the use of randomness extractors, we present our main result in terms of bounds on the smooth min-entropy of the returned samples.

Protocol details

Our primary objective in the protocol design is to minimize the time between the client submitting a quantum circuit and receiving the corresponding bitstring. As a result, our protocol is designed to mitigate the following experimental considerations:

1. There is a marked latency due to network communication and the time to load a circuit into the quantum device controls. Furthermore, there is also overhead associated with executing a circuit. To ameliorate this, instead of submitting circuits one at a time, we group the circuits into batches of 15 or 20 jobs, with each job consisting of two circuits joined by a layer of mid-circuit measurements and reset. Each batch of size b, therefore, consists of $2b$ circuits.
2. There is downtime associated with the device, such as during periodic calibrations. Before submitting a batch, a client probes the machine for readiness using a predetermined precheck circuit C_{precheck}. This circuit announces the intent of the client to submit a batch of circuits and triggers any server-side maintenance if necessary.
3. To ensure that the device does not stall and to keep the average time per sample low, we demand that the entire batch be returned within a cutoff time of $2.5 \times 2b$ s. If the entire batch is not received within this cutoff time, we cancel all outstanding jobs in the batch, and we discard all bitstrings received from this batch.

To formally describe our experimental protocol with all details accurately represented (including details on challenge circuits generation and randomness extraction), we present the following protocol.

Protocol arguments.

$n \in \mathbb{N}$: Number of qubits
$d \in \mathbb{N}$: Circuit depth
$M \in \mathbb{N}$: Total number of samples
$b \in \mathbb{N}$: Batch size
$m \in \mathbb{N}$: Test set size
$K_{\text{seed}} \in \{0,1\}^r$: Random bitstring that is private to client
$T_{b,\text{cutoff}}$: Round-trip communication time threshold between the client and the server for a batch
$t_{\text{threshold}}$: Threshold on the overall average time-per-sample
χ : Threshold for the XEB test
Ext: : $(\kappa, \varepsilon_{\text{ext}})$-Quantum-proof strong extractor (see definition 4 in Supplementary Information)
$K_{\text{ext}} \in \{0,1\}^s$: A random seed for the extractor
C_{precheck} : A predetermined 'precheck' instruction used to announce the client's readiness to submit a batch

Protocol steps.

1. Set the samples collected $\mathcal{M}_{\text{keep}} = \emptyset$.
2. Set $i = 0$, $T_{\text{tot}} = 0$.
3. Initialize a pseudorandom generator with an r-bit seed K_{seed}.
4. While $|\mathcal{M}_{\text{keep}}| < M$, run the following steps:
   a. Challenge circuit generation subroutine: the client generates each of the circuits $\{C_{i\cdot 2b+k}\}_{k=1}^{2b}$ as follows.
      i. Initialize an empty circuit on n qubits.
      ii. For $j = 1, \ldots, d$, run the following steps:
         A. Sample n SU(2) gates using the seeded pseudorandom generator and apply them to all n qubits.
         B. Apply the two-qubit gates corresponding to layer T_j of the chosen edge-coloured circuit topology.
      iii. Sample n SU(2) gates using the seeded pseudorandom generator and apply them to all n qubits.
   b. Precheck: the client submits the precheck circuit C_{precheck} and waits for a response.
   c. Client–server interaction subroutine:
      i. Start a timer.
      ii. The client submits the batch of circuits $\{C_{i\cdot 2b+k}\}_{k=1}^{2b}$ to the server.
      iii. The server responds with a batch of $2b$ bitstrings $\{x_{i\cdot 2b+k}\}_{k=1}^{2b}$.
      iv. Stop the timer. Record interaction time T_b.
      v. Time out scenario: If $T_{\text{b}} > T_{b,\text{cutoff}}$, then discard the batch.
      vi. If the batch is not discarded, then client computes $\mathcal{M}_{\text{keep}} = \mathcal{M}_{\text{keep}} \bigcup_{k=1}^{2b} \{x_{i\cdot 2b+k}\}$ and accumulates the time $T_{\text{tot}} = T_{\text{tot}} + T_{\text{b}}$.
      vii. Client increments the counter, $i = i + 1$.
5. Abort condition 1: if $T_{\text{tot}}/|\mathcal{M}_{\text{keep}}| > t_{\text{threshold}}$, then abort the protocol.
6. XEB score verification subroutine:
   a. Test set construction: the client samples a subset $\mathcal{V}$ of size m randomly from $\mathcal{M}_{\text{keep}}$ using the seeded pseudorandom generator.
   b. Compute the score $\text{XEB}_{\text{test}} = \left((2^n/m) \cdot \sum_{j\in\mathcal{V}} |\langle x_j|C_j|0\rangle|^2\right) - 1$.
   c. Abort condition 2: if $\text{XEB}_{\text{test}} < \chi$ then abort the protocol.
7. If not-abort, the client feeds the M samples $x_1, \ldots, x_M$ together with the random seed K_{ext} to the extractor Ext.

Output: Conditioned on the protocol not aborting, the protocol returns $\text{Ext}(K_{\text{ext}}, (x_1, \ldots, x_M))$ as the final bitstring.

Protocol security

Our primary theoretical contribution is the security of the implemented protocol against a restricted adversary. Our adversarial model considers realistic and near-term adversaries using best-known strategies (see Supplementary Information section IIIC for details). In brief, our adversary has a bounded classical computer and a quantum computer and uses both to generate the samples. Specifically, we make the following key assumptions about the adversary (further elaborated in Supplementary Information section IIIC):

1. The server does not perform any postselection attacks; that is, the M detected rounds in the protocol are a fair representation of the adversary behaviour.
2. Of the M valid samples, the server a priori selects Q rounds for which it honestly returns samples by executing the challenge circuit on the quantum computer. For the remaining $M - Q$ samples, it returns deterministic samples obtained by simulating the circuits on a powerful classical computer (of power $\mathcal{A}$, measured in terms of number of floating point operations per second).
3. For each of the Q quantum rounds, it interacts only with the quantum computer once (it does not attempt to oversample a circuit).

In practice, these assumptions are probably stronger than necessary; we leave adaptation of the formal cryptographic protocol for a relaxed set of assumptions for future work.

To prove the security of the protocol, we prove a lower bound to the smooth min-entropy $H_{\min}^{\varepsilon_s}(X^M|\tilde{I}_{sn})$ of the bits before the extractor given this adversary, where $\tilde{I}_{sn}$ is the initial snapshot of side information minus the randomness extractor seed. To do so, we first provide a bound on the probability that the server executing a fixed number Q of quantum rounds passes the XEB test with threshold χ (see Supplementary Information section IIID). We denote the event in which the protocol does not abort as Ω, the probability of not aborting as $\Pr[\Omega]$ and the upper bound on the probability as $\varepsilon_{adv}(Q,\chi)$.

Now, given a target not-abort probability $\varepsilon_{accept} = 4\varepsilon_s$ (for an ε_{sou}-sound protocol, $\varepsilon_{accept} = 4\varepsilon_s = \varepsilon_{sou}$), the upper bound to $\Pr[\Omega]$ allows us to compute $Q_{min} = \min\{Q: \varepsilon_{adv}(Q,\chi) \geq 4\varepsilon_s\}$, which represents the minimum number of quantum rounds that the server needs to perform for the protocol to not abort with probability $4\varepsilon_s$. Given Q_{min}, we bound the smooth min-entropy of the samples X^M given classical side information $\tilde{I}_{sn}$ using the following theorem.

Theorem 1. Let Ω denote the event in which the randomness certification protocol in Supplementary Information section IA does not abort and let σ be the state over registers X^M and $\tilde{I}_{sn}$. Given $\varepsilon_s \in (0, 1/4)$, the protocol either aborts with a probability greater than $1 - 4\varepsilon_s$ or

$$H_{\min}^{\varepsilon_s}(X^M|\tilde{I}_{sn}) \geq Q_{min}(n-1) + \log\varepsilon_s, \tag{4}$$

where $Q_{min} = \arg\min_Q\{\varepsilon_{adv}(Q,\chi) \geq 4\varepsilon_s\}$ and $\varepsilon_{adv}(Q,\chi)$ is the upper bound to $\Pr(\Omega)$.

Data availability

The full data presented in this work are available at Zenodo (https://doi.org/10.5281/zenodo.12952178).

Code availability

The code required to verify and reproduce the results presented in this work is available at Zenodo (https://doi.org/10.5281/zenodo.12952178).

Acknowledgements We thank J. Dimon, D. Pinto and L. Beer for their executive support of the Global Technology Applied Research Center of JPMorganChase and our work in Quantum Computing. We thank the technical staff at the Global Technology Applied Research Center of JPMorganChase for their invaluable contributions to this work. We are thankful to J. Gray for helpful discussions on tensor network contraction path optimization using CoTenGra. We acknowledge the entire Quantinuum team for their many contributions toward the successful operation of the H2 quantum computer with 56 qubits, and we acknowledge Honeywell for fabricating the trap used in this experiment. J.L., M.L., Y.A. and D.L. acknowledge support from the US Department of Energy, Office of Science, under contract DE-AC02-06CH11357 at Argonne National Laboratory and the US Department of Energy, Office of Science, National Quantum Information Science Research Centers. S.A. and S.-H.H. acknowledge the support from the US Department of Energy, Office of Science, National Quantum Information Science Research Centers and Quantum Systems Accelerator. T.S.H. was supported by the US Department of Energy, Office of Science, Advanced Scientific Computing Research program office under the quantum computing user program. This research used supporting resources at the Argonne and the Oak Ridge Leadership Computing Facilities. The Argonne Leadership Computing Facility at Argonne National Laboratory is supported by the Office of Science of the US DOE under contract no. DE-AC02-06CH11357. The Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory is supported by the Office of Science of the US DOE under contract no. DE-AC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC award DDR-ERCAP0030284.

Author contributions M.P., C.L. and R.S. devised the project. M.L., R.S., P.N., M.D. and M.F.-F. designed the protocol implementation. M.L., R.S. and P.N. implemented the code for circuit generation and client–server interaction. P.N. executed the experiments on the quantum computer and collected the data. M.L., R.S., P.N., A.A., J.L. and D.L. implemented and benchmarked the tensor-network-based verification code. M.L. executed the verification on supercomputers and collected the data. Y.A. and T.S.H. provided support for supercomputer runs. M.L. and P.N. analysed the data. M.L., R.S., P.N., S.C., S.-H.H. and S.A. developed the complexity-theoretic analysis. M.L., R.S., P.N., W.Y.K., E.C.-M., K.C., O.A. and C.L. developed the main security analysis. C.L., M.P., R.S., N.K., S.E. and F.J.C. improved the adversarial model and enhanced its connection to applications. K.J.B., J.M.D., N.E., C.F., D.H., M.M., S.A.M., J.W., B.N. and P.S. maintained, optimized and operated the trapped-ion hardware and software stack. M.P. led the overall project as the lead principal investigator. All authors contributed to technical discussions and the writing and editing of the paper.

Competing interests M.P., M.L., P.N. and R.S. are co-inventors on a patent application related to this work (no. 18/625,605, filed on 3 April 2024 by JPMorgan Chase). The authors declare no other competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-025-08737-1.
Correspondence and requests for materials should be addressed to Ruslan Shaydulin, Charles Lim or Marco Pistoia.
Peer review information *Nature* thanks Arthur Mehta and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.