# Verifying BQP computations on noisy devices with minimal overhead

Editors' Suggestion

# Verifying BQP Computations on Noisy Devices with Minimal Overhead

Dominik Leichtle,[1] Luka Music,[1] Elham Kashefi,[2,1] and Harold Ollivier[3,1,*]

[1] *Laboratoire d'Informatique de Paris 6, CNRS, Sorbonne Université, 4 Place Jussieu, Paris 75005, France*
[2] *School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, United Kingdom*
[3] *INRIA, 2 rue Simone Iff, Paris 75012, France*

(Received 14 January 2021; revised 31 May 2021; accepted 8 July 2021; published 4 October 2021)

With the development of delegated quantum computation, clients will want to ensure confidentiality of their data and algorithms and the integrity of their computations. While protocols for blind and verifiable quantum computation exist, they suffer from high overheads and from oversensitivity: when running on noisy devices, imperfections trigger the same detection mechanisms as malicious attacks, resulting in perpetually aborted computations. We introduce the first blind and verifiable protocol for delegating bounded-error quantum polynomial (BQP) computations to a powerful server, with repetition as the only overhead. It is composable and statistically secure with exponentially low bounds and can tolerate a constant amount of global noise.

DOI: 10.1103/PRXQuantum.2.040302

## I. INTRODUCTION

Remotely accessible quantum computing platforms free clients from the burden of maintaining complex physical devices in house. Yet, when delegating computations, they want their data and algorithms to remain private and these computations to be executed as specified. Several methods have been devised to achieve this (e.g., Refs. [1,2]; for a review, see Ref. [3]). Nonetheless, a practical solution remains to be found, as all known protocols are too sensitive to noise. Indeed, they have been designed for perfect devices, thus aborting as soon as the smallest deviation is detected. Unfortunately, the replacement of such machines by even slightly noisy ones would make the verification procedure abort constantly, mistaking plain imperfections for the signature of malicious behavior.

To deal with this oversensitivity, previous research has: given up on blindness [4]; imposed restrictions on the noise model [5]; switched to a setting with two noncommunicating servers and classical clients [6]; or introduced computational assumptions [7]. Yet, these protocols either only achieve inverse-polynomial security or obtain exponential security by requiring an additional fault-tolerant encoding of the computation on top of the one used to suppress device noise.

We tackle this problem for bounded-error quantum polynomial (BQP) computations—i.e., the class of decision problems that quantum computers can solve efficiently—by introducing a protocol that provides noise robustness, verification, blindness, and delegation. The protocol repeats the client's computation framed in the measurement-based quantum computation (MBQC) model—a natural choice for delegating computations—several times in a blind fashion while interleaving these executions with test rounds that aim at detecting dishonest behavior of the server. A final majority vote over the computation rounds mitigates possible errors, thus providing the desired robustness.

Combined with blindness, this forces the server to attack at least a constant fraction of the rounds to corrupt the computation, hence increasing its chances of getting caught by the tests. Information-theoretic security is proven in the composable framework of abstract cryptography (AC) [8], ensuring that security is not jeopardized by sequential or simultaneous instantiations with other protocols.

Crucially, our protocol has *no space overhead* for each round when compared to the insecure computation in the MBQC model: the only price to pay for exponential security and correctness is a *polynomial number of repetitions* of computations similar to the unprotected one. This lets the client use the full extent of the available hardware for its computational tasks and any increase in the capabilities of the quantum devices can be used entirely to scale up these computations. These properties make it, to our knowledge, the first experimentally realizable solution for verification of BQP computations, thus going beyond experimental feasibility demonstrations of verifiable

*harold.ollivier@inria.fr

building blocks [9–12] and potentially serving as a blueprint for the development of future quantum network applications.

## II. PRELIMINARIES

### A. BQP computations

The complexity class BQP contains the decisions problems that can be solved with bounded error probability using a polynomial-size quantum circuit. More formally, a language $L$ is in BQP if there is a family of polynomial-size quantum circuits that decides the language with an error probability of at most $p$. The chosen value for $p$ is arbitrary as long as it is fixed and is usually taken to be $1/3$. Hence, a BQP computation for $L$ will have output $F(x) = 1$ for $x \in L$ with probability at least $1 - p$, while it will have output $F(x) = 0$ for $x \notin L$ with probability at least $1 - p$. In the following, for a given BQP computation, $p$ will be referred to as the *inherent error probability* to distinguish it from errors due to external causes such as the use of noisy devices.

### B. Measurement-based quantum computation

An MBQC algorithm (also called a *measurement pattern*) consists of a graph $G = (V, E)$, two vertex sets $I$ and $O$ defining input and output vertices, a list of angles $\{\phi_v\}_{v\in V}$ with $\phi_v \in \Theta := \{k\pi/4\}_{0\leq k\leq 7}$, and a flow. To run it, the client instructs the server to prepare the graph state $|G\rangle$: for each vertex in $V$, the server creates a qubit in the state $|+\rangle$ and performs a control-$Z$ (CZ) gate for each pair of qubits in $E$. The client then asks the server to measure each qubit of $V$ along the basis $\{|+_{\phi'_v}\rangle\langle+_{\phi'_v}|, |-_{\phi'_v}\rangle\langle-_{\phi'_v}|\}$ in the order defined by the flow of the computation, with $|+_\alpha\rangle = (|0\rangle + e^{i\alpha}|1\rangle)/\sqrt{2}$. The corrected angle $\phi'_v$ is given by $\phi'_v = (-1)^{s_v^X}\phi_v + s_v^Z\pi$ for binary values of $s_v^X$ and $s_v^Z$ that depend only on the outcomes of previously measured qubits and the flow. More details about the flow and the update rules for the measurement angles can be found in Refs. [13,14].

As shown in Ref. [15], the MBQC model is equivalent to the circuit model, so that any BQP algorithm in the circuit model can be translated into the MBQC model with at most polynomial overhead.

### C. Hiding the computation

A computation can easily be hidden if, instead of the server preparing each qubit, the client: (i) for all $v \in V$ sends $|+_{\theta_v}\rangle$ with $\theta_v$ chosen uniformly at random in $\Theta$; (ii) asks the server to measure the qubits in the basis defined by the angle $\delta_v = \phi'_v + \theta_v + r_v\pi$ for $r_v$ a random bit, while keeping $\theta_v$ and $r_v$ hidden from the server; and (iii) uses $s_v = b_v \oplus r_v$, where $b_v$ is the measurement outcome to compute $s_v^X$ and $s_v^Z$ defined above. Here, the angle $\theta_v$ acts as a one-time pad for $\phi'_v$, while $r_v$ does the same for the measurement outcomes. This idea was first formalized in the *universal blind quantum computation* (UBQC) Protocol in Ref. [1].

### D. Verifiability through trap insertion

Verifiable protocols allow the client to check that its computation has been done correctly. To do this, the client enlarges the graph used for the computation to insert traps. These traps are made from qubits randomly prepared in $|+_\theta\rangle$ states and disconnected from the subgraph used for performing the desired computation with the help of *dummy qubits*—i.e., randomly initialized qubits sent by the client in states $\{|0\rangle, |1\rangle\}$. The first verification protocol via trapification has been introduced in Ref. [2]. It has been further optimized into the *verifiable blind quantum computation* (VBQC) protocol of Refs. [16,17], achieving a linear overhead.

## III. NOISE-ROBUST VERIFIABLE PROTOCOL

Our noise-robust VBQC protocol is formally defined in Protocol 1, where test rounds are used in conjunction with computation rounds to provide verifiability. We introduce it more intuitively in the following paragraphs and discuss the features that make it suitable for practical purposes.

### A. Trap insertion for BQP computations

Because BQP computations have classical inputs and classical outputs, there exists a more economical trap insertion than is available for quantum input and quantum output computations. More concretely, it does not require any enlargement of the graph to insert traps alongside the computation. Rather, the idea is to interleave pure *computation rounds* (i.e., without inserted traps) and *pure test rounds* (i.e., only made up of traps).

Given a UBQC computation defined by a graph $G$, we construct test rounds based on a $k$ coloring $\{V_i\}_{i\in[k]}$ of $G$. A partition of a graph in $k$ sets—called colors—is a valid $k$ coloring if all adjacent vertices in the graph have different colors. Therefore, by definition, a $k$ coloring satisfies $\bigcup_{i=1}^k V_i = V$, and $\forall i \in [k]$, $\forall v \in V_i : N_G(v) \cap V_i = \emptyset$, where $N_G(v)$ are the neighbors of $v$ in $G$. Hence, for each color $i$, the client can decide to insert traps for all vertices of $V_i$ and dummies in all other positions. This defines the test round associated with color $i$. These tests require the same sequence of operations for the server as regular UBQC computations, making them undetectable.

### B. Informal presentation of the protocol

Suppose that the client wishes to delegate a BQP computation corresponding to a measurement pattern on a graph $G$ to the server. The client chooses a coloring $\{V_i\}_{i\in[k]}$ of $G$ and two integers $d$ and $t$. All these parameters are fixed

**Protocol 1** Noise-Robust VBDQC for BQP Computations

**Client's Inputs:** Angles $\{\phi_v\}_{v\in V}$ and flow $f$ on graph $G$, classical input to the computation $x \in \{0,1\}^{\#I}$ (where $\#X$ is the size of $X$).
**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 sub-protocol (the Client may send message $\mathsf{Redo}_j$ 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 colour $\mathsf{V}_j \in_R \{V_k\}_{k\in[K]}$ (this is the set of traps for this test round).
   (b) The Client sends $\#V$ qubits to the Server. If $j \in T$ and the destination qubit $v \notin \mathsf{V}_j$ is a non-trap qubit (therefore a dummy), then the Client chooses uniformly at random $d_v \in_R \{0,1\}$ and sends the state $|d_v\rangle$. Otherwise, the Client chooses at random $\theta_v \in_R \Theta$ and sends the state $|+_{\theta_v}\rangle$.
   (c) The Server performs a $\mathsf{CZ}$ gate between all its qubits corresponding to an edge in the set $E$.
   (d) For $v \in V$, the Client sends a measurement angle $\delta_v$, the Server measures the appropriate corresponding qubit in the $\delta_v$-basis, returning outcome $b_v$ to the Client. The angle $\delta_v$ is defined as follows:
   - 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 $\delta_v$.
   - If $j \in T$ (test): if $v \notin \mathsf{V}_j$ (dummy qubit), the Client chooses it uniformly at random from $\Theta$; if $v \in \mathsf{V}_j$ (trap qubit), it chooses uniformly at random $r_v \in_R \{0,1\}$ and sets $\delta_v = \theta_v + r_v\pi$.
3. For all $j \in T$ (test round) and $v \in \mathsf{V}_j$ (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 neighbouring 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 $c_{fail} \geq w$ then the Client aborts by sending message $\mathsf{Abort}$ 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 $\#\{y_j \,|\, j \in C,\, y_j = y\} > \frac{d}{2}$. If such a value $y$ exists (this is then the majority output), it sets it as its output and sends message $\mathsf{Ok}$ to the Server. Otherwise it sends message $\mathsf{Abort}$ to the Server.

for a given instantiation of the protocol and are publicly available to both parties.

The client runs the UBQC protocol $n := t + d$ times successively. For $d$ of the rounds chosen at random (computation rounds), the client updates the measurement angles according to the measurement pattern of its desired computation. The remaining $t$ rounds are test rounds. For each such test round, the client secretly chooses a color at random and sends traps for vertices of that color and dummies everywhere else. The client instructs the server to measure all qubits as in computation rounds but with the measurement angle of trap qubits corresponding to the basis in which they are prepared and a random measurement basis for the dummies. Because the trap qubits are isolated from each other, they should remain in their initial state. A test round is said to have *passed* if all the traps yield the expected measurement results and *failed* otherwise. Figure 1 depicts such a possible succession of rounds.

At the end of the protocol, the client counts the number of failed test rounds. If this number is higher than a given threshold $w$, it aborts the protocol by sending the message $\mathsf{Abort}$ to the server [18]. Otherwise, it sets the majority outcome of the computation rounds as its output and sends message $\mathsf{Ok}$ to the server.

In this construction, all rounds share the same underlying graph $G$ and the same order for the measurements of qubits, and all angles are chosen from the same uniform distribution. We prove formally later that this implies blindness—i.e., the server cannot distinguish computation and test rounds, nor tell which qubits are traps—which in turn makes this trap-insertion strategy efficient to obtain verifiability. The range and influence of the parameters on verifiability and noise-robustness bounds are detailed in the next section.

### C. Redo feature

Because the client or the server may experience unintentional device failures, they might wish to discard and redo a round $j \in [n]$. In this case, our protocol allows each party to send a $\mathsf{Redo}_j$ request to the other, in which case both parties simply repeat the exact same round, albeit with fresh randomness. $\mathsf{Redo}_j$ requests are allowed only so long as the party asking for them is still supposed to be manipulating the qubits of round $j$. We show that this does not impact the blindness or the verifiability of the scheme. This means that a dishonest server cannot use $\mathsf{Redo}$ requests to trick the client into accepting an incorrect result. Such a capability of our protocol is crucial in practice: without it, detected honest failures of devices happening during a test round would be counted as a failed test round, thus drastically decreasing the likelihood

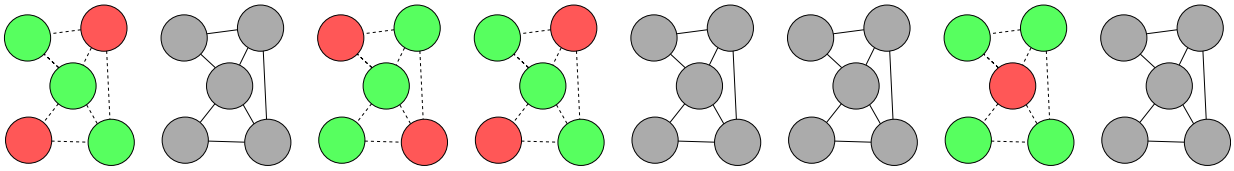

FIG. 1. An example of rounds of the proposed protocol. Graphs in gray denote computation rounds, while graphs containing red nodes (traps) and green nodes (dummies) are test rounds. Each qubit is always included in one type of test round. The server remains completely oblivious to the differences between the rounds, which are solely known to the client.

of successfully completing the protocol. Since the rounds concerned can be safely repeated, the only consequence of experimental failures caught during an execution is an increase in the expected number of rounds.

### D. Exponential security amplification

The above approach to trap insertion is efficient as the only overhead is the repetition of the same subprotocol. Yet, the use of a single computation round and $n-1$ test rounds would leave at least a $1/n$ chance for the server to corrupt the computation. The only previously known method to obtain an exponentially low cheating probability has been to insert traps into a single computation round at the expense of drastically increasing the complexity of the graph and then using fault-tolerant encoding on top to amplify the security. By restricting the computation to BQP computations, we prove that a classical repetition error-correcting code is sufficient to achieve an exponentially low cheating probability. This amplification technique is common in purely classical scenarios, where attacks can be classically correlated across various rounds. Although this claim has also been made in the quantum case in previous works [2,5,16], up to now it has remained unproven. The difficulty, which we address below, is that quantum attacks entangled across rounds are much more powerful than classical correlations allow.

## IV. SECURITY RESULTS AND NOISE ROBUSTNESS

This section presents the security properties of the protocol in the AC framework of Ref. [8] and its noise robustness on honest devices. See the Appendix for formal definitions and proofs of Theorems 1 and 2.

### A. Security analysis

In AC, security is defined as indistinguishability between an ideal resource, which is secure by definition, and its real-world implementation, i.e., the protocol. This framework ensures a higher standard of security than in other approaches (see, e.g., Ref. [19] and Sec. 5.1 of Ref. [20]) and is inherently composable, meaning that security holds when the protocol is repeated sequentially or in parallel with others. This property is crucial, as delegated protocols are important stepping stones toward more complex functionalities (e.g., a subroutine for building multiparty-quantum-computation protocols [21]).

Our security proof uses the results of Ref. [22] that reduce the composable security of a verifiable delegated quantum computation protocol to four *stand-alone criteria*:

(a) $\epsilon_{\text{cor}}$-local-correctness: the protocol with honest players produces the expected output.
(b) $\epsilon_{\text{bl}}$-local-blindness: the server's state at the end of the protocol is indistinguishable from the one that it could have generated on its own.
(c) $\epsilon_{\text{ver}}$-local-verifiability: the client either accepts a correct computation or aborts the protocol.
(d) $\epsilon_{\text{ind}}$-independent-verification: the server can determine on its own, using the transcript of the protocol and its internal registers, whether or not the client will decide to abort.

Then, the local-reduction theorem (Corollary 6.9 from Ref. [22]) states that if a protocol implements a unitary transformation on classical inputs and is $\epsilon_{\text{cor}}$-locally-correct, $\epsilon_{\text{bl}}$-locally-blind and $\epsilon_{\text{ver}}$-locally-verifiable with $\epsilon_{\text{ind}}$-independent-verification, then it is $\epsilon$-composably-secure with:

$$\epsilon = \max\{\epsilon_{\text{sec}}, \epsilon_{\text{cor}}\} \text{ and } \epsilon_{\text{sec}} := 4\sqrt{2\epsilon_{\text{ver}}} + 2\epsilon_{\text{bl}} + 2\epsilon_{\text{ind}}. \tag{1}$$

With this at hand, we can state our main result:

**Theorem 1** (Security of Protocol 1)**.** *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 $[0,(1/k)(2p-1)/(2p-2)]$, where $p$ is the inherent error probability of the BQP computation, Protocol 1 with $d$ computation rounds, $t$ test rounds, and a maximum number of tolerated failed test rounds of $w$ is $\epsilon$-composably-secure, with $\epsilon$ exponentially small in $n$.*

### B. Simple upper bound on the probability of failure

The $\epsilon_{\text{ver}}$-local-verifiability amounts to upper bounding the probability that an erroneous result is accepted by $\epsilon_{\text{ver}}$.

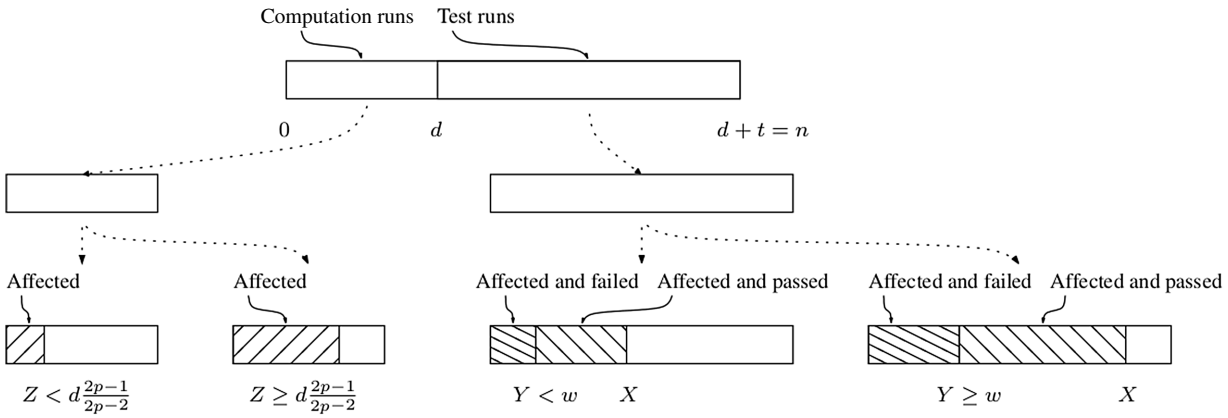


FIG. 2. The four cases needed to determine a closed-form upper bound for the probability of failure. First, we determine the probability for the number of affected computation rounds. If it is low enough $[Z < d(2p-1)/(2p-2)]$, there is no need to abort. If it is high $[Z \geq d(2p-1)/(2p-2)]$, we find a bound on the probability that the number of failed test rounds $Y$ is below or above $w$.

Figure 2 presents the four possible configurations for corrupted computations and traps. Given a BQP computation that decides whether or not $x$ belongs to the language $L$, our protocol would yield the correct result after the majority vote whenever less than $d/2$ computation rounds yield $F(x) \oplus 1$. These erroneous results can be due to malicious behavior of the server, to its use of noisy devices, or to inherent errors of the BQP algorithm. It is expected that, in $pd$ computation rounds, the BQP computation will give an inherently erroneous result and that this will happen for a fraction greater than $p$ only with negligible probability. Therefore, the result obtained by running our protocol will be correct whenever it is possible to guarantee that there is a negligible probability that the server corrupts more than $[(1/2) - p - \varphi]\,d$ computation runs for some $\varphi > 0$. To this end, we use the trapification paradigm. First, this ensures that each nontrivial deviation to the computation will be detected by at least one of the $k$ possible types of test rounds. Second, because the deviations are distributed equally among test and computation runs, we can conclude that if fewer than $[(1/2) - p - \varphi - \varepsilon_1]t$ test runs are corrupted for some $\varepsilon_1 > 0$, then fewer than $[(1/2) - p - \varphi]d$ computations are corrupted with overwhelming probability. This implies that setting $w = [(1/k) - \varepsilon_2][(1/2) - p - \varphi - \varepsilon_1]t$ for $\varepsilon_2 > 0$ yields an exponentially low probability of failure. Since $\varphi, \varepsilon_1, \varepsilon_2$ can be chosen arbitrarily small, we conclude that $\epsilon_{\text{ver}}$ can be made negligible for $0 < w/t < (1/k)[(1/2) - p]$.

### C. Improved upper bound on the probability of failure

The former bound can be improved by realizing that some situations leading to incorrect results are double counted. Indeed, we need to consider inherent errors from the BQP computation solely for the computation rounds that are unaffected by the server's malicious behavior. This is due to the blindness of the scheme ensuring that the server's deviation will be distributed equally among computation rounds with or without inherent errors. Denoting by $m$ the total number of rounds affected by the server's deviation, we expect $[md + (n-m)pd]/n$ computation rounds to be erroneous. The first term comes from deviations of the server, while the second comes from inherent errors in the BQP computation when the server has not deviated on these rounds. The requirement that this quantity is below $d/2$ amounts to guaranteeing that $m < n(2p-1)/(2p-2)$, which can be obtained following the line of argument given in the previous paragraph whenever $w$ satisfies $0 < w/t < (1/k)(2p-1)/(2p-2)$.

### D. Local correctness on honest-but-noisy devices

None of the stand-alone criteria introduced above consider device imperfections. In fact, the analysis of correctness, blindness, and verification makes no distinction between device imperfections and potentially malicious behavior. Although satisfactory—these properties make our protocol a concrete implementation of the ideal resource for verifiable delegated quantum computation—it could still fall short of expectations in terms of usability because nonmalicious device imperfections could cause unintentional aborts. Fortunately, for a class of realistic imperfections, our protocol is capable of correcting their impact and accepts with high probability. In such case, the final outcome is the same as that obtained on noiseless devices with honest participants.

This additional *noise-robustness* property, the main innovation of this paper, means that Protocol 1 also satisfies the local-correctness property with negligible $\epsilon_{\text{cor}}$ for a noisy but honest client and/or server. This property holds under the following restrictions:

(a) The noise can be modeled by round-dependent Markovian processes—i.e., a possibly different

arbitrary completely positive trace-preserving (CPTP) map acting on each round.

(b) The probability that at least one of the trap measurements fails in any single test round is upper bounded by some constant $p_{\max} < (1/k)(2p-1)/(2p-2)$ and lower bounded by $p_{\min} \leq p_{\max}$.

Theorem 2 states that, in order for the protocol to terminate correctly with overwhelming probability on these noisy devices, $w$ should be chosen such that $w/t > p_{\max}$. Conversely, for any choice of $w/t < p_{\min}$, we show that the protocol aborts with overwhelming probability.

**Theorem 2** (Local Correctness of VDQC Protocol on Noisy Devices, Informal)**.** *As before, p denotes the inherent error probability for the BQP computation. Assume a Markovian round-dependent model for the noise on the client and server devices and let* $p_{\min} \leq p_{\max} < (1/k)(2p-1)/(2p-2)$ *be, respectively, a lower and an upper bound on the probability that at least one of the trap-measurement outcomes in a single test round is incorrect. If* $w/t > p_{\max}$*, Protocol 1 is* $\epsilon_{\text{cor}}$*-locally-correct with exponentially low* $\epsilon_{\text{cor}}$*. On the other hand, if* $w/t < p_{\min}$*, then the probability that Protocol 1 terminates without aborting is exponentially low.*

Using the local-reduction theorem from Ref. [22] again, this new bound concerning local correctness on noisy devices can be combined with noise-independent blindness, input-independent verification, and verifiability to yield a composably secure protocol for $\epsilon = \max\{\epsilon_{\text{sec}}, \epsilon_{\text{cor}}\}$. Here, $\epsilon$ might depend on the noise level of the devices through $\epsilon_{\text{cor}}$.

## V. DISCUSSION

### A. Role of noise assumptions in correctness analysis

Our security proof does not rely on any assumption regarding the form or amplitude of the noise: it considers any deviation as potentially malicious and shows that the protocol provides information-theoretic verification and blindness. The assumptions on the noise—limited strength and Markovianity—are used only to show that correctness holds not only in the honest and noiseless case but also when the imperfections of the devices are mild. In such cases, their impact on the computation can be mitigated and the protocol will accept with high probability.

### B. Fine-tuning the number of repetitions

For specific computations with fixed security and correctness targets as well as noise levels, several parameters can be tuned to optimize the total run time of our protocol. First, distributing rounds across different machines is an effective way to reduce the overall execution time, while composability ensures that security is preserved. Second, for a fixed graph, a smaller value of $k$ allows a larger value of $p_{\max}$, since exponential verification and correctness require $p_{\max} < w/t < (1/k)(2p-1)/(2p-2)$: finding a small $k$ coloring of the graph used for the computation widens the gap between the chosen threshold ratio $w/t$ and $(1/k)(2p-1)/(2p-2)$, thereby reducing the number of rounds required to obtain the desired security and correctness levels [23]. Third, the ratio $d/t$ also influences the number of repetitions. Given fixed values for $p$, $k$, $w/t$, and the security and correctness levels, the optimal ratio can be determined numerically using Eqs. (E5) and (F1), which explicitly relate the failure and success probabilities to these parameters.

### C. Decoupling verifiability and fault tolerance

Because a single trap has bounded sensitivity—the probability $\alpha$ of not detecting an attack at a given vertex is bounded away from 0—it must be boosted to obtain exponential security. Previous work has resorted to fault-tolerant encoding of the computation path to ensure that $r$ errors can be corrected (see Ref. [2,16]). This forces attackers to corrupt at least $r$ locations to affect the computation, which decreases the probability of not detecting such attacks to $\alpha^r$. Increasing the security of these protocols simultaneously increases the minimum distance of the fault-tolerant amplification scheme, thereby reducing the number of available qubits to perform the computation.

The repetition of test rounds and the majority vote in our protocol serve the same purpose but with a much lighter impact. Because our detection-probability amplification relies on a classical procedure, all qubits can be devoted to useful computations irrespective of the desired security level.

Additionally, our protocol does not abort at the first failed trap, while previous approaches do. This means that, in the presence of noise, other protocols always require an exponentially low global residual-error level to accept with overwhelming probability. On the contrary, our protocol only needs the average ratio of failed test rounds to be upper bounded away from $(1/k)(2p-1)/(2p-2)$, which requires us to bring the global residual error level to a constant only. This promises to drastically ease the experimental feasibility of verified quantum computations.

## ACKNOWLEDGMENTS

We thank Theodoros Kapourniotis and Atul Mantri for fruitful discussions. We acknowledge support from the European Union (EU) H2020 Program under Grant Agreement No. 820445 (Quantum Internet Alliance). D.L. acknowledges support from the EU H2020 Program under Grant Agreement No. ERC-669891 (Almacrypt), and by the French ANR Projects ANR-18-CE39-0015 (CryptiQ) and ANR-18-CE47-0010 (QUDATA).

## APPENDIX A: USEFUL INEQUALITIES FROM PROBABILITY THEORY

The following definitions and lemmata are useful tools for our proof: for more in-depth definitions, see Ref. [24].

**Definition 1** (Hypergeometric distribution). *Let $N, K, n \in \mathbb{N}$, with $0 \leq n, K \leq N$. A random variable $X$ is said to follow the hypergeometric distribution, denoted as $X \sim \text{Hypergeometric}(N, K, n)$, if its probability mass function is described by*

$$\Pr[X = k] = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}.$$

*As one possible interpretation, $X$ describes the number of drawn marked items when drawing $n$ items from a set of size $N$ containing $K$ marked items, without replacement.*

**Lemma 1** (Tail bound for the hypergeometric distribution). *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $0 < t < K/N$. It then holds that*

$$\Pr\left[X \leq \left(\frac{K}{N} - t\right) n\right] \leq \exp\left(-2t^2 n\right).$$

**Corollary 1.** *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $0 < \lambda < (nK/N)$. It then holds that*

$$\Pr[X \leq \lambda] \leq \exp\left[-2n\left(\frac{K}{N} - \frac{\lambda}{n}\right)^2\right].$$

**Lemma 2** (Serfling's bound for the hypergeometric distribution [25,26]). *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $\lambda > 0$. It then holds that*

$$\Pr\left[\sqrt{n}\left(\frac{X}{n} - \frac{N}{K}\right) \geq \lambda\right] \leq \exp\left(-\frac{2\lambda^2}{1 - \frac{n-1}{N}}\right).$$

**Corollary 2.** *Let $X \sim \text{Hypergeometric}(N, K, n)$ be a random variable and $\lambda > (nK/N)$. It then holds that*

$$\Pr[X \geq \lambda] \leq \exp\left[-2n\left(\frac{\lambda}{n} - \frac{K}{N}\right)^2\right].$$

Note the symmetry of Corollaries 1 and 2.

**Lemma 3** (Hoeffding's inequality for the binomial distribution). *Let $X \sim \text{Binomial}(n, p)$ be a random variable. For any $k \leq np$, it then holds that*

$$\Pr[X \leq k] \leq \exp\left(-2\frac{(np-k)^2}{n}\right).$$

*Similarly, for any $k \geq np$, it holds that*

$$\Pr[X \geq k] \leq \exp\left(-2\frac{(np-k)^2}{n}\right).$$

## APPENDIX B: FORMAL SECURITY DEFINITIONS

We model $N$-round two-party protocols between players $A$ (the honest client) and $B$ (the potentially dishonest server) as a succession of $2N$-CPTP maps $\{\mathcal{E}_i\}_{i\in[1,N]}$ and $\{\mathcal{F}_j\}_{j\in[1,N]}$. The maps $\{\mathcal{E}_i\}_i$ act on $\mathcal{A}$, the register of $A$, and $\mathcal{C}$, a shared communication register between $A$ and $B$. Similarly, the maps $\{F_j\}_j$ act on $\mathcal{B}$ and $\mathcal{C}$. Note that $\mathcal{B}$ and the maps $\{\mathcal{F}_j\}_j$ can be chosen arbitrarily by $B$ and, thus, unless $B$ is specified to be behaving honestly, there is no guarantee that they are those implied by our protocol. Since we are only interested in protocols where $A$ is providing a classical input $x$, we will equivalently write the input as the corresponding computational-basis state $|x\rangle$ used to initialize $\mathcal{A}$, whereas $\mathcal{B}$ and $\mathcal{C}$ are initialized in a fixed state $|0\rangle$.

Below, we denote by $\Delta(\rho, \sigma) = (1/2)\|\rho - \sigma\|$, the distance on the set of density matrices induced by the trace norm $\|\rho\| = \text{Tr}\sqrt{\rho^\dagger \rho}$. We first define $\mathcal{S}$ the ideal resource for verifiable delegated quantum computation and then the local properties from Ref. [22].

### 1. Ideal resource for verifiable delegated quantum computation

The ideal resource $\mathcal{S}$ has interfaces for two parties, interface of $A$ and interface of $B$. The interface of $A$ takes two inputs: a classical input string $x$ and the description of $\mathcal{U}$, the computation to perform. The interface of $B$ is filtered by a bit $b$. When $b = 0$, there is no further legitimate input from $B$, while for $b = 1$, it is allowed to send a bit $c$ that determines the output of the computation available at the interface of $A$. When $b = 0$ or $c = 0$, the output at the interface of $A$ is equal to $\mathcal{M}_{\text{Comp}} \circ \mathcal{U}(|x\rangle)$, where $\mathcal{M}_{\text{Comp}}$ is the computational-basis measurement. This corresponds to a "no cheating" behavior. When $c = 1$, $B$ decides to cheat and $A$ receives the Abort message, which can be given as a quantum state of $\mathcal{A}$ that is taken orthogonal to any other possible output state. At the interface of $B$, $\mathcal{S}$ outputs nothing for $b = 0$, while for $b = 1$, $B$ receives $l(\mathcal{U}, x)$, the permitted leakage. For generic MBQC computations, the permitted leakage is set to $G$, the graph used in the computation. When $G$ is a universal graph for MBQC computation, the permitted leakage reduces to an upper bound on the size of the computation No. $\mathcal{U}$.

For this ideal resource, the blindness is an immediate consequence of the server receiving at most the permitted

leak, while verifiability is a consequence of the computation being correct when the server is not cheating, while being aborted otherwise.

#### 2. $\epsilon_{\text{cor}}$-local-correctness

Let $\mathcal{P}_{AB}$ be a two-party protocol as defined above with the honest CPTP maps for players $A$ and $B$. We say that such a protocol implementing $\mathcal{U}$ is $\epsilon_{cor}$-*locally-correct* if for all possible inputs $x$ for $A$, we have

$$\Delta\left[\mathrm{Tr}_{\mathrm{B}} \circ \mathcal{P}_{\mathrm{AB}}(|\mathrm{x}\rangle), \mathcal{U}(|\mathrm{x}\rangle)\right] \leq \epsilon_{\mathrm{cor}}. \tag{B1}$$

#### 3. $\epsilon_{\text{bl}}$-local-blindness

Let $\mathcal{P}_{AB}$ be a two-party protocol as defined above and where the maps $\{\mathcal{E}_i\}_i$ are the honest maps. We say that such a protocol is $\epsilon_{\text{bl}}$-*locally-blind* if, for each choice of $\{\mathcal{F}_i\}_i$, there exists a CPTP map $\mathcal{F}' : L(\mathcal{B}) \to L(\mathcal{B})$ such that, for all inputs $x$ for $A$, we have

$$\Delta\left[\mathrm{Tr}_{\mathrm{A}} \circ \mathcal{P}_{\mathrm{AB}}(\rho), \mathcal{F}' \circ \mathrm{Tr}_{\mathrm{A}}(|\mathrm{x}\rangle)\right] \leq \epsilon_{\mathrm{bl}}. \tag{B2}$$

#### 4. $\epsilon_{\text{ind}}$-independent verification

Let $\mathcal{P}_{AB}$ be a verifiable two-party protocol as defined above, where the maps $\{\mathcal{E}_i\}_i$ are the honest maps. Let $\bar{B}$ be a qubit extending the register of $B$ and initialized in $|0\rangle$. Let $\mathcal{Q}_{A\bar{B}} : L(\mathcal{A} \otimes \bar{\mathcal{B}}) \to L(\mathcal{A} \otimes \bar{\mathcal{B}})$ be a CPTP map which, conditioned on $\mathcal{A}$ containing the state $|\mathsf{Abort}\rangle$, switches the state in $\bar{\mathcal{B}}$ from $|0\rangle$ to $|1\rangle$ and does nothing in the other cases.

We say that the verification procedure for such a protocol is $\epsilon_{\text{ind}}$-*independent* from the input of player $A$ if there exist CPTP maps $\mathcal{F}'_i : L(\mathcal{C} \otimes \mathcal{B} \otimes \bar{\mathcal{B}}) \to L(\mathcal{C} \otimes \mathcal{B} \otimes \bar{\mathcal{B}})$ such that

$$\Delta\left[\mathrm{Tr}_A \circ \mathcal{Q}_{A\bar{B}} \circ \mathcal{P}_{AB}(\rho), \mathrm{Tr}_A \circ \mathcal{P}'_{AB\bar{B}}(\rho)\right] \leq \epsilon_{\mathrm{ind}}, \tag{B3}$$

where

$$\mathcal{P}'_{AB\bar{B}} := \mathcal{E}_1 \circ \mathcal{F}'_1 \circ \ldots \circ \mathcal{E}_n \circ \mathcal{F}'_n.$$

#### 5. $\epsilon_{\text{ver}}$-local-verifiability

Let $\mathcal{P}_{AB}$ be two-party protocols as defined above, where the maps for $A$ are the honest maps, while the maps $\{\mathcal{F}_j\}_j$ for $B$ are not necessarily corresponding to the ideal (honest) ones. Let $x$ be the input given by $A$ in the form of a computational state $|x\rangle$ and let $\mathcal{U}$ be the computation it wants to perform. The protocols $\mathcal{P}_{AB}$ are $\epsilon_{\text{ver}}$-*locally-verifiable* for $A$ if, for each choice of CPTP maps $\{\mathcal{F}_j\}_j$, there exists $p \in [0, 1]$ such that we have

$$\Delta[\mathrm{tr}_{\mathrm{B}}\mathcal{P}_{\mathrm{AB}}(|\mathrm{x}\rangle), \mathrm{p}\mathcal{U}(|\mathrm{x}\rangle) + (1-\mathrm{p})|\mathsf{Abort}\rangle\langle\mathsf{Abort}|] \leq \epsilon_{\mathrm{ver}}.$$

## APPENDIX C: COMPOSABLE SECURITY

In the paragraphs below, we show that our protocol satisfies each of the stand-alone criteria before combining them to obtain composable security.

#### 1. Perfect local correctness

On perfect (non-noisy) devices, local correctness is implied by the correctness of the underlying UBQC protocol. This is because all the completed computation rounds correspond to the same deterministic UBQC computation and that on such devices, general UBQC protocols have been proven to be perfectly correct [1,22]. Thus $\epsilon_{\text{cor}} = 0$.

#### 2. Perfect local blindness

In the event that the computation is accepted, each round looks exactly like a UBQC computation to the server. Therefore, the blindness comes directly from the composability of the various UBQC rounds that make up our protocol [22]. In the event that the computation is aborted, we need to take into account the fact that a possibly malicious server could deduce the position of a trap qubit. That could be the case if it attacked a single position in the test rounds and got caught. Yet, as the position of the traps is not correlated to the input or to the computation itself, knowing it does not grant additional attack capabilities to the server and blindness is recovered again as a consequence of the blindness of UBQC. More detailed statements can be found in Appendix D, where it is also shown that Redo requests have no effect on the local blindness of the scheme.

#### 3. Perfect local independent verification

Because in our protocol, the client shares with the server whether the computation is a success or an abort, this is trivially verified.

#### 4. Exponential local verifiability

Local verifiability is satisfied if any deviation by the possibly malicious server yields a state that is $\epsilon_{\text{ver}}$ close to a mixture of the correct output and the Abort message. Equivalently, the probability that the server makes the client accept an incorrect outcome is bounded by $\epsilon_{\text{ver}}$. Let $d/n$, $t/n$, and $w/t$ be the ratios of test, computation, and tolerated failed test rounds. The local verifiability of our protocol is given by Theorem 3 and proven subsequently.

#### 5. Proof of exponential composable security

Our protocol has perfect correctness (for noiseless devices), blindness, and input-independent verification. In addition, it is $\epsilon_{\text{ver}}$-locally-verifiable, with $\epsilon_{\text{ver}}$ exponentially small in $n$. Therefore, by the local-reduction theorem, it is $\epsilon$-composably-secure, with $\epsilon = \epsilon_{\text{sec}} = 4\sqrt{2\epsilon_{\text{ver}}}$ and $\epsilon$

exponentially small in $n$. Note that because we use the local-reduction theorem to obtain fully composable security, we incur an additional square root on our verifiability bound given by Eq. (1) and need to satisfy the additional independence property. This is, of course, not required if the protocol is only used sequentially with other schemes, which will probably be the case in early quantum computations, since the machines will not be able to handle multiple protocols at the same time. In this case, the standalone model would be sufficient, since it provides sequential composition, but would fail if parallel composition is needed.

## APPENDIX D: PROOF OF PERFECT LOCAL BLINDNESS

*Proof.* To prove that Eq. (B2) holds for $\epsilon_{\mathrm{bl}} = 0$, first note that at the end of our protocol, the client $A$ reveals to the server $B$ whether the computation is accepted or aborted. Hence, each case can be analyzed separately. Second, we show that the interrupted rounds that have triggered a Redo can be safely ignored. Indeed, each one of them is the beginning of an interrupted UBQC computation and, because UBQC is composable and perfectly blind [22], no information can leak to the server through the transmitted qubits. In addition, our protocol restricts the honest party $A$ in its ability to emit Redo requests, so that no correlations are created between the index of the interrupted rounds and $\mathcal{U}$ or the secret random parameters used in the rounds (angle and measurement padding, and trap preparations). As a consequence, from the point of view of $B$, the state of the interrupted rounds is completely independent of the state of the noninterrupted ones and does not contain information regarding the input, the computation, or secret parameters. That is, its partial trace over $A$ can be generated by $B$ alone.

For the noninterrupted rounds, we can invoke the same kind of independence argument between the computation rounds and the test rounds. As a result, blindness of our protocol stems from the blindness of the underlying computation rounds. In the event that the full protocol is a success, we can rely on the composability of the perfect blindness of each UBQC computation round to have perfect local blindness. For an abort, we can consider a situation that is more advantageous for $B$ by supposing that alongside the Abort message sent by $A$, it also gives away the location of the trap qubits. In this modified situation, the knowledge of the computation being aborted does not convey additional information to $B$, as it only reveals that one of the attacked positions is a trap qubit, which $B$ now already knows. Using our independence argument between the trap location on the one hand and the inputs, the computation, and other secret parameters, we conclude that revealing the location of the trap qubits does not affect the blindness of the computation rounds. Hence, using composability again and combining the abort and accept cases, we arrive at Eq. (B2), with $\epsilon_{\mathrm{bl}} = 0$. ■

## APPENDIX E: PROOF OF VERIFIABILITY

**Theorem 3** (Local Verifiability of Protocol 1)**.** *Let* $0 < w/t < (1/k)(2p-1)/(2p-2)$ *and* $0 < d/n < 1$ *be fixed ratios, for* $k$ *different test rounds and where* $p$ *is the inherent error probability of the BQP computation. Then, Protocol 1 is* $\epsilon_{\mathrm{ver}}$*-locally-verifiable for exponentially low* $\epsilon_{\mathrm{ver}}$.

*Proof.* The proof of the verifiability of a computation amounts to upper bounding the probability of yielding a wrong output while not aborting. This could be the result of the inherent randomness of the BQP computation that gives the wrong outcome with probability $p$ or of the server deviating from the instructed computation. In the following, although rounds are expected to be run sequentially, the proof will examine the state of the *combined computation*. This state corresponds to the server having simultaneous unrestricted access to all quantum systems sent by the client and possibly operating on them as a whole irrespective of the underlying rounds to which they belong. In particular, the server could decide to perform some action on a qubit given measurements in one or several of the underlying runs or to entangle the various underlying runs together.

Note that because the parties can only ask for redoing a run independently of the input, of the computation, of the used randomness, and of the output of the computation itself (comprising the result of trap measurements), interrupted runs can be safely ignored in the verification analysis, as the state corresponding to these runs is uncorrelated to that of the completed runs. ■

### 1. Output of the combined computation

First, consider the output density operator $B(\{\mathcal{F}_j\}_j, \nu)$ representing all the classical messages that the client $A$ receives during its interaction with the server $B$, comprising the final message containing the encrypted measurement outcomes. Below, the CPTP maps $\{\mathcal{F}_j\}_j$ represent the chosen deviation of $B$ on the combined computation. By encoding the classical messages as quantum states in the computational basis, the output density operator satisfies

$$B(\{\mathcal{F}_j\}_j, \nu) = \mathrm{Tr}_{\mathrm{B}}\Big\{ \sum_{\mathrm{b}} |\mathrm{b}+\mathrm{c_r}\rangle\langle \mathrm{b}| \mathcal{F}\mathcal{P} \times \left(|0\rangle\langle 0|_B \otimes |\Psi^{\nu,b}\rangle\langle\Psi^{\nu,b}|\right) \times \mathcal{P}^\dagger\mathcal{F}^\dagger, |b\rangle\langle b+c_r| \Big\} \tag{E1}$$

where $b$ is the list of measurement outcomes defining the computation branch; $\nu$ is a composite index relative to the secret parameters chosen by $A$, i.e., the type of each underlying run, the padding of the measurement angles and measurement outcomes, and the trap setup; $|b+c_r\rangle\langle b|$ ensures that only the part corresponding to the current computation branch is taken into account and removes the one-time-pad encryption on nonoutput and nontrap qubits while leaving output and trap qubits unaffected, i.e., encrypted; $|0\rangle\langle 0|_B$ is some internal register for $B$ in a fixed initial state; and $\left|\Psi^{\nu,b}\right\rangle$ is the state of the qubits sent by $A$ to $B$ at the beginning of the protocol tensored with quantum states representing the measurement angles of the computation branch $b$.

To obtain this result, the line of proof of Ref. [2] can be applied to the combined computation. This works by noting that for a given computation branch $b$ and given random parameters $\nu$, all the measurement angles are fully determined. Therefore, provided that the computation branch is $b$, the measurement angles can be included in the initial state. This defines $\left|\Psi^{\nu,b}\right\rangle$. Then, each $\mathcal{F}_j$ is decomposed into an honest part and a pure deviation. All the deviations are commuted and collected into $\mathcal{F}$, applied after $\mathcal{P}$, the unitary part of the honest protocol, is applied. The projections onto $|b\rangle$ then ensures that, after the deviation induced by $B$, the perceived computation branch is $b$. This, together with the decrypting of nonoutput nontrap qubits, gives Eq. (E1).

### 2. Probability of failure

Recall that a failure for the combined computation on input $x$ occurs when the result after decrypting the outputs and performing the majority vote differs from $F(x)$ while the computation is accepted.

For the combined computation to be accepted, no more than $w$ test runs should have a trap-qubit measurement outcome opposite to what is expected. Let $\mathtt{T}$ denote the set of trap qubits, which is determined by $T$, the set of test runs, and the type of each test run. In the absence of any deviation on the combined computation, their expected value is $|r_{\mathtt{T}}\rangle = \bigotimes_{\mathtt{t}\in\mathtt{T}} |r_{\mathtt{t}}\rangle$, where $r_{\mathtt{T}} = (r_{\mathtt{t}})_{\mathtt{t}\in\mathtt{T}}$ denotes the measurement-outcome padding values restricted to trap qubits. Therefore, the projector onto the states of the trap qubits yielding to an accepted combined computation can be written as $Q_\perp = \sum_{\mathtt{w}\in\mathtt{W}} X_{\mathtt{T}}^{\mathtt{w}} |r_{\mathtt{T}}\rangle\langle r_{\mathtt{T}}| X_{\mathtt{T}}^{\mathtt{w}}$, with $X_{\mathtt{T}}^{\mathtt{w}} = \bigotimes_{\mathtt{t}\in\mathtt{T}} X_{\mathtt{t}}^{\mathtt{w}\mathtt{t}}$ and where $\mathtt{W}$ is the set of length $|\mathtt{T}|$ binary vectors $\mathtt{w}$ that have at least a one in no more than $w$ underlying (test) runs.

Similarly, define the set of output qubits by $\mathtt{O}$. The correct value for these output qubits is $|F(x)_{\mathtt{O}} + r_{\mathtt{O}}\rangle$. Then, for $\mathtt{V}$ the set of length $\#\mathtt{O}$ binary vectors $\mathtt{v}$ that have at least $d/2$ ones in the underlying (computation) runs, the operator $P_\perp = \sum_{\mathtt{v}\in\mathtt{V}} X_{\mathtt{O}}^{\mathtt{v}} |F(x)+r_{\mathtt{O}}\rangle\langle F(x)+r_{\mathtt{O}}| X_{\mathtt{O}}^{\mathtt{v}}$ with $X_{\mathtt{O}}^{\mathtt{v}} = \bigotimes_{\mathtt{o}\in\mathtt{O}} X_{\mathtt{o}}^{\mathtt{v}\mathtt{o}}$ is the projector onto the subspace of states that yield an incorrect result for the whole computation. This is because when each output has been decrypted by the client—the one-time padding $r_{\mathtt{O}}$ is removed—the majority vote will output $F(x)+1$ because more than half of the outputs are equal to $F(x)+1$.

Combining these two projectors allows to write the probability of failure:

$$\Pr[\text{fail}] = \sum_{\nu} \sum_{b,k,\sigma,\sigma'} \Pr[\nu] \mathrm{Tr}\Big\{ (\mathrm{P}_\perp \otimes \mathrm{Q}_\perp) \times \left(\alpha_{k\sigma}\alpha^*_{k\sigma'} |b+c_r\rangle\langle b|\sigma\mathcal{P}|\Psi^{\nu,b}\rangle\langle\Psi^{\nu,b}|\mathcal{P}^\dagger\sigma'|b\rangle\langle b+c_r|\right)\Big\},$$

where $\mathcal{F}$ is decomposed into Kraus operators indexed by $k$, that are in turn decomposed onto the Pauli basis through the coefficients $\alpha_{k\sigma}$ and $\alpha_{k\sigma'}$. Consequently, $\sigma$ and $\sigma'$ are Pauli matrices.

Using the explicit expressions for $P_\perp$ and $Q_\perp$, the above formula can be simplified:

$$\begin{aligned}\Pr[\text{fail}] = \sum_{\nu} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \sum_{b',k,\sigma,\sigma'} \Pr[\nu] \Big\{ & \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| \otimes \left\langle b'\right| (X_{\mathtt{O}}^{\mathtt{v}} \otimes X_{\mathtt{T}}^{\mathtt{w}}) \\ & \times \left(\alpha_{k\sigma}\alpha^*_{k\sigma'} \mathcal{P}|\Psi^{\nu,b}\rangle\langle\Psi^{\nu,b}|\mathcal{P}^\dagger\sigma'\right) \\ & (X_{\mathtt{O}}^{\mathtt{v}} \otimes X_{\mathtt{T}}^{\mathtt{w}}) |F(x)_{\mathtt{O}} + r_{\mathtt{O}}\rangle \otimes |r_{\mathtt{T}}\rangle \otimes \left|b'\right\rangle \Big\},\end{aligned}$$

where $b'$ is the binary vector obtained from $b$ by restricting it to nonoutput and nontrap qubits. This is obtained using the circularity of the trace and the fact that $\sum_b \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| (X_{\mathtt{O}}^{\mathtt{v}} \otimes X_{\mathtt{T}}^{\mathtt{w}}) |b+c_r\rangle\langle b| = \sum_{b'} \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| \otimes |b'+c_r\rangle\langle b'| (X_{\mathtt{O}}^{\mathtt{v}} \otimes X_{\mathtt{T}}^{\mathtt{w}})$, since there is no decoding for output and trap qubits—i.e., $c_r$ is 0.

### 3. Using blindness of the scheme

At this point, standard proofs of verifiability sum over the secret parameters defining the encryption to twirl the deviation of the server and trace out nontrap qubits. Here, because it is necessary to assess the probability of having more than half of the output qubits yielding the wrong measurement output $F(x)+1$, the trace is taken on nontrap and nonoutput qubits only.

The design of the protocol yielding the combined computation ensures blindness. This implies that the resulting state of any set of qubits after applying $\mathcal{P}$ and taking the average over their possible random preparation parameters is a completely mixed state. This can be applied in the above equation for the set of nonoutput and nontrap qubits. For output and trap qubits, the inner products must be computed before taking the sum over their random preparation parameters $\nu_{\mathtt{O}}$ and $\nu_{\mathtt{T}}$, respectively.

This gives

$$\Pr[\text{fail}] = \sum_{\nu_{\mathtt{O}},\nu_{\mathtt{T}},\mathtt{u}} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \sum_{b',k,\sigma,\sigma'} \Pr[\nu_{\mathtt{O}},\nu_{\mathtt{T}}]\alpha_{k\sigma}\alpha^*_{k\sigma'} \times \Big\{ \langle F(x)_{\mathtt{O}} + r_{\mathtt{O}}| \otimes \langle r_{\mathtt{T}}| \otimes \langle b'| \left(X_{\mathtt{O}}^{\mathtt{v}} \otimes X_{\mathtt{T}}^{\mathtt{w}}\right) \times \sigma' \left( |s_{\mathtt{O}} + r_{\mathtt{O}}\rangle\langle s_{\mathtt{O}} + r_{\mathtt{O}}| \otimes |r_{\mathtt{T}}\rangle\langle r_{\mathtt{T}}| \otimes \frac{\mathbb{I}}{\mathrm{Tr}\mathbb{I}} \right) \sigma' \times \left(X_{\mathtt{O}}^{\mathtt{v}} \otimes X_{\mathtt{T}}^{\mathtt{w}}\right) |F(x)_{\mathtt{O}} + r_{\mathtt{O}}\rangle \otimes |r_{\mathtt{T}}\rangle \otimes |b'\rangle \Big\},$$

where $|s_{\mathtt{o}}\rangle$ is the state of the output qubit $\mathtt{o} \in \mathtt{O}$ when no deviation is applied by the server.

In the above equation, the contribution of each qubit factorizes. For $l \notin \mathtt{O} \cup \mathtt{T}$, because the Pauli matrices are traceless save for the identity, the only nonvanishing terms are obtained for $\sigma_l = \sigma'_l$, where subscript $l$ is used to select the action of $\sigma$ and $\sigma'$ on qubit $l$. In such a case, the corresponding multiplicative factor equals 1. A direct calculation shows that, for an output qubit $\mathtt{o} \in \mathtt{O}$,

$$\sum_{r_{\mathtt{o}}} \langle F(x)_{\mathtt{o}} + r_{\mathtt{o}}| X_{\mathtt{o}}^{\mathtt{v}_{\mathtt{o}}} \sigma_{\mathtt{o}} |s_{\mathtt{o}} + r_{\mathtt{o}}\rangle \langle s_{\mathtt{o}} + r_{\mathtt{o}}| \sigma'_{\mathtt{o}} X_{\mathtt{o}}^{\mathtt{v}_{\mathtt{o}}} |F(x)_{\mathtt{o}} + r_{\mathtt{o}}\rangle = 0$$

for $\sigma_{\mathtt{o}} \neq \sigma'_{\mathtt{o}}$. Similarly, for a trap qubit $\mathtt{t} \in \mathtt{T}$, $\sum_{r_{\mathtt{t}}} \langle r_{\mathtt{t}}| X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} \sigma_{\mathtt{t}} |r_{\mathtt{t}}\rangle \langle r_{\mathtt{t}}| \sigma'_{\mathtt{t}} X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} |r_{\mathtt{t}}\rangle$ vanishes for $\sigma_{\mathtt{t}} \neq \sigma'_{\mathtt{t}}$. Combining these yields

$$\begin{aligned}\Pr[\text{fail}] &= \sum_{\nu_{\mathtt{O}},\nu_{\mathtt{T}}} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} \sum_{k,\sigma} \Pr[\nu_{\mathtt{O}},\nu_{\mathtt{T}}] |\alpha_{k\sigma}|^2 \\ &\quad \times \prod_{\mathtt{o}\in\mathtt{O}} |\langle F(x)_{\mathtt{o}} + r_{\mathtt{o}}| X_{\mathtt{o}}^{\mathtt{v}_{\mathtt{o}}} \sigma_{\mathtt{o}} |s_{\mathtt{o}} + r_{\mathtt{o}}\rangle|^2 \\ &\quad \times \prod_{\mathtt{t}\in\mathtt{T}} |\langle r_{\mathtt{t}}| X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} \sigma_{\mathtt{t}} |r_{\mathtt{t}}\rangle|^2 \\ &= \sum_k \sum_\sigma |\alpha_{k\sigma}|^2 f(\sigma),\end{aligned}$$

with

$$\begin{aligned}f(\sigma) = \sum_{\nu_{\mathtt{O}},\nu_{\mathtt{T}}} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} &\Pr[\nu_{\mathtt{O}},\nu_{\mathtt{T}}] \\ &\times \prod_{\mathtt{o}\in\mathtt{O}} |\langle F(x)_{\mathtt{o}} + r_{\mathtt{o}}| X_{\mathtt{o}}^{\nu_{\mathtt{o}}} \sigma_{\mathtt{o}} |s_{\mathtt{o}} + r_{\mathtt{o}}\rangle|^2 \\ &\times \prod_{\mathtt{t}\in\mathtt{T}} |\langle r_{\mathtt{t}}| X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} \sigma_{\mathtt{t}} |r_{\mathtt{t}}\rangle|^2. \qquad \text{(E2)}\end{aligned}$$

In short, this proves that the overall deviation $\mathcal{F}$ has the same effect as a convex combination of Pauli deviations $\sigma$, each occurring with probability $\sum_k |\alpha_{k,\sigma}|^2$.

### 4. Implicit upper bound

Because $\sum_{k,\sigma} |\alpha_{k\sigma}|^2 = 1$, the worst-case scenario for the bound in Eq. (E2) is when $\alpha_{k\sigma} = 1$ for $\sigma$ such that $f(\sigma)$ is maximum. Hence, the probability of failure is upper bounded as follows:

$$\Pr[\text{fail}] \leq \max_\sigma f(\sigma).$$

Protocol 1 defines the trap and output qubit configuration $\nu_{\mathtt{O}}, \nu_{\mathtt{T}}$ by: (i) the set $\mathtt{T}$ of trap qubits, itself entirely determined by the positions and kinds of test runs within the sequence of runs; and (ii) the preparation parameters $\theta_l$ and $r_l$ of each trap and output qubit. Each parameter of (i) and (ii) being chosen independently, the probability of a given configuration $\nu_{\mathtt{O}}, \nu_{\mathtt{T}}$ can be decomposed into the probability $\Pr[\mathtt{T}]$ for a given configuration of trap locations multiplied by the probability of a given configuration for the prepared state of the trap and output qubits, $\prod_{l\in\mathtt{O}\cup\mathtt{T}} \sum_{\theta_l,r_l} \Pr[\theta_l, r_l]$. Using this, one can rewrite $f(\sigma)$:

$$\begin{aligned}f(\sigma) = \sum_{\mathtt{T}} \sum_{\mathtt{v}\in\mathtt{V},\mathtt{w}\in\mathtt{W}} &\Pr[\mathtt{T}] \\ &\times \prod_{\mathtt{o}\in\mathtt{O}} \sum_{\theta_{\mathtt{o}},r_{\mathtt{o}}} \Pr[\theta_{\mathtt{o}},r_{\mathtt{o}}] |\langle F(x)_{\mathtt{o}} + r_{\mathtt{o}}| X_{\mathtt{o}}^{\mathtt{v}_{\mathtt{o}}} \sigma_{\mathtt{o}} |s_{\mathtt{o}} + r_{\mathtt{o}}\rangle|^2 \\ &\times \prod_{\mathtt{t}\in\mathtt{T}} \sum_{\theta_{\mathtt{t}},r_{\mathtt{t}}} \Pr[\theta_{\mathtt{t}},r_{\mathtt{t}}] |\langle r_{\mathtt{t}}| X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}}.\sigma_{\mathtt{t}} |r_{\mathtt{t}}\rangle|^2. \qquad \text{(E3)}\end{aligned}$$

For $\sigma$ a Pauli deviation, denote by $\sigma_{|X}$ the binary vector indexed by qubit positions of the combined computation, where ones mark qubit positions for which $\sigma$ acts as $X$ or $Y$. Abusing notation, in the following, $\mathtt{O}$ denotes the binary vector over qubit positions $i$ of the combined computation where ones are positioned for qubits in $O$—that is, the vector $(\mathbb{1}_{i\in\mathtt{O}})_i$ for $i$ a qubit location. Similarly, $\mathtt{T}$ will also denote $(\mathbb{1}_{i\in\mathtt{T}})_i$.

Using the fact that $|\langle r_{\mathtt{t}}| X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} \sigma_{\mathtt{t}} |r_{\mathtt{t}}\rangle|^2$ is 1 for $X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} \sigma_{\mathtt{t}} \in \{I, Z\}$ and 0 otherwise, the product over the trap qubits can be written as

$$\prod_{\mathtt{t}\in\mathtt{T}} \sum_{\theta_{\mathtt{t}},r_{\mathtt{t}}} \Pr[\theta_{\mathtt{t}},r_{\mathtt{t}}] |\langle r_{\mathtt{t}}| X_{\mathtt{t}}^{\mathtt{w}_{\mathtt{t}}} \sigma_{\mathtt{t}} |r_{\mathtt{t}}\rangle|^2 = \begin{cases} 1, & \text{for } \mathtt{T}.\sigma_{|X} = \mathtt{w}, \\ 0, & \text{otherwise,} \end{cases}$$

where, for $a$ and $b$ binary vectors, $a.b$ is the bit-wise binary product vector.

For output qubits, before attempting the same computation, it is important to point out an important dependency of the deviation for the output qubits. Failing to take it into account would yield an overly optimistic bound. This dependency is due to the fact that, contrarily to trap qubits, where the perfect protocol performs the identity, the output qubits are the result of more complex computation.

More precisely, the guarantee given by the protocol at this stage is only blindness. Following the definition of the blind-computing ideal resource given in Appendix B—Eq. (B2)—the server is able to choose a deviation $\mathcal{E}$ and have it applied to the unprotected input of the protocol $x$, while itself not getting either $x$ or $\mathcal{E}(x)$. While $\mathcal{E}$ has been reduced here to a convex sum of Pauli deviations applied after the perfect protocol, nothing prevents these Pauli deviations from incorporating a dependency on the input $x$ or on the unencrypted output of the perfect protocol. In short, this means that the server can craft a deviation in such a way that only outputs equal to $F(x)$ are flipped, leaving those yielding $F(x)+1$ unaffected.

Going forward with the computation of factors for output qubits in Eq. (E3), it is thus necessary to distinguish output qubits that belong to computation rounds where no nontrivial deviation takes place and those that do not. Define $\mathtt{u}$ to be the random binary vector of length $\#\mathtt{O}$ such that $s_\circ = F(x)+\mathtt{u}_\circ$. For an output qubit that is part of a computation round without a nontrivial deviation,

$$\begin{aligned}&\sum_{\theta_\circ,r_\circ}\Pr[\theta_\circ,r_\circ]\,|\,\langle F(x)_\circ+r_\circ|\,X_\circ^{\mathtt{v}_\circ}\sigma_\circ\,|s_\circ+r_\circ\rangle\,|^2\\&\quad=\sum_{\theta_\circ,r_\circ}\Pr[\theta_\circ,r_\circ,\mathtt{u}_\circ]\\&\quad\times\,|\,\langle F(x)_\circ+r_\circ|\,X_\circ^{\mathtt{v}_\circ}\sigma_\circ X_\circ^{\mathtt{u}_\circ}\,|F(x)+r_\circ\rangle\,|^2\\&\quad=\begin{cases}\Pr[\mathtt{u}_\circ], & \text{for } \sigma_{|X,\circ}+\mathtt{u}_\circ=\mathtt{v}_\circ\\ 0, & \text{otherwise.}\end{cases}\end{aligned}$$

When the output qubit is part of a computation round with a nontrivial deviation, the dependency argument given above yields

$$\begin{aligned}&\sum_{\theta_\circ,r_\circ}\Pr[\theta_\circ,r_\circ]\\&\quad\times\,|\,\langle F(x)_\circ+r_\circ|\,X_\circ^{\mathtt{v}_\circ}\sigma_\circ X_\circ^{\mathtt{u}_\circ}\,|F(x)+r_\circ\rangle\,|^2\le\Pr[\mathtt{u}_\circ].\end{aligned}$$

Hence, for a fixed $\sigma$, a necessary condition on $\mathtt{u}$ and $\mathtt{T}$ for having a nonzero contribution to $f(\sigma)$ is thus

$$wt(\mathtt{T}.\sigma_{|X})\le w\quad\text{and}\quad wt(\mathtt{u}.\neg\mathtt{S})\ge d/2-\#\mathtt{S},$$

where $wt(.)$ is the Hamming weight of a binary vector, $\mathtt{S}$ is a length-$\#\mathtt{O}$ binary vector where the ones are located on output qubits where at least one nontrivial deviation is performed in the corresponding computation round, and $\neg\mathtt{S}$ is the bitwise negation of $\mathtt{S}$.

Combining the corresponding bounds and summarizing the necessary condition with $(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma$, we obtain

$$f(\sigma)\le\sum_{(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma}\Pr[\mathtt{T},\mathtt{u}].$$

In other words, to record a failure of the protocol, the number of incorrect trap rounds needs to be below the threshold $w$, while the number of nontrivially attacked computation rounds needs to be greater than $d/2$ reduced by the amount of incorrect outcomes on nonattacked rounds due to the inherent randomness of the algorithm.

### 5. Explicit upper bound

Now, assume that the maximum of the bound above is attained for some $\sigma$ that happens to nontrivially affect one of the rounds, say $k$, on more than one qubit. Consider $\sigma'$ with the sole difference compared to $\sigma$ that $\sigma'$ restricted to one of these two qubits is equal to the identity. Then, $\sigma'$ still affects the round $k$ nontrivially, which implies that all configurations $(\mathtt{T},\mathtt{u})$ in $\Upsilon_\sigma$ are also in $\Upsilon_{\sigma'}$. Therefore,

$$\Pr[\text{fail}]\le\max_m\max_{\sigma\in E_m}\sum_{(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma}\Pr[\mathtt{T},\mathtt{u}],$$

where $E_m$ denotes the set of Pauli operators with $m$ single-qubit nontrivial deviations all in distinct rounds.

Because the bound above depends on $\mathtt{u}$ only through $wt(\mathtt{u}.\neg\mathtt{S})$ and because for any such subset the random variable $wt(\mathtt{u}.\neg\mathtt{S})$ is less than $B[wt(\neg\mathtt{S}),p]$ in the usual stochastic order, we obtain

$$\Pr[\text{fail}]\le\max_m\max_{\sigma\in E_m}\sum_{(\mathtt{T},\mathtt{u})\in\Upsilon_\sigma}\Pr[\mathtt{T}]\times\Pr[\tilde{\mathtt{u}}=\mathtt{u}],$$

in which $\tilde{\mathtt{u}}$ is a random binary vector where each coordinate follows a Bernouilli law with probability $p$ and where $B(n,p)$ is the binomial distribution for $n$ draws and probability $p$. Using the fact that the random choice of test runs is completely uniform, the right-hand side is invariant under permutations of the test and computation runs. It is thus possible to restrict the range of the maximum to the specific Pauli operators $\sigma_m$ with a deviation on a single qubit in each of the first $m$ runs:

$$\Pr[\text{fail}]\le\max_m\sum_{T\in\Upsilon_{\sigma m}}\Pr[\mathtt{T}].\tag{E4}$$

### 6. A closed from for the upper bound

To find a closed-form upper bound for the soundness error, we now distinguish between two regimes for $m$, controlled by the parameter $\varphi>0$:

1. For $m\le[(1/k)(2p-1)/(2p-2)-\varphi]\,n$, we find a small upper bound on the probability that the client obtains a wrong result.
2. On the other hand, for $m\ge[(1/k)(2p-1)/(2p-2)-\varphi]n$, we find a small upper bound on the probability that the client accepts the outcome of the protocol, i.e., that the verification passes.

In the following, we define the constant ratios of test, computation, and tolerated failed test runs as $\delta := d/n$, $\tau := t/n$ and $\omega := w/t$. Let $Z$ be a random variable counting the number of affected computation runs (by the server's deviation or by inherent failure of the algorithm) and let $Y$ be a random variable counting the number of failed test runs, i.e., the number of affected test runs where the deviation hits a trap. We have that

$$\begin{aligned}\Pr[\text{fail}] &\leq \max_m \sum_{T\in\Upsilon_{\sigma_m}} \Pr[T] = \max_m \Pr\left[Z \geq \frac{d}{2} \ \wedge \ Y \leq w\right] \\ &\leq \max\left\{\max_{m\leq\left(\frac{2p-1}{2p-2}-\varphi\right)n} \Pr\left[Z \geq \frac{d}{2}\right], \max_{m\geq\left(\frac{2p-1}{2p-2}-\varphi\right)n} \Pr[Y \leq w]\right\}.\end{aligned}$$

Since $\Pr[Z \geq d/2]$ and $\Pr[Y \leq w]$ are, respectively, increasing and decreasing with the number of attacked runs, both inner maximums are attained for $m = [(1/k)(2p-1)/(2p-2) - \varphi]\, n$ and we therefore focus on this case.

Analogously to the verification proof of the original protocol, the second term can be bounded from above by first determining the minimum number of affected test runs before calculating the probability that the server's attack triggers a sufficient number of traps.

Hence, with $X$ denoting the number of test runs affected by the server's deviation, tail bounds for the hypergeometric distribution imply, for all $\varepsilon_1 > 0$, that

$$\Pr\left[X \leq \left(\frac{m}{n} - \varepsilon_1\right) t\right] \leq \exp\left(-\frac{2\tau^2\epsilon_1^2}{\frac{2p-1}{2p-2} - \varphi} n\right).$$

Further, it follows by Hoeffding's bound for the binomial distribution that

$$\begin{aligned}&\Pr\left[Y \leq \left(\frac{1}{k} - \varepsilon_2\right)\left(\frac{m}{n} - \epsilon_1\right) t \;\middle|\; X = \left(\frac{m}{n} - \varepsilon_1\right) t\right] \\ &\quad\leq \exp\left[-2\left(\frac{2p-1}{2p-2} - \varphi - \varepsilon_1\right)\tau\varepsilon_2^2 n\right].\end{aligned}$$

All in all, we therefore obtain

$$\begin{aligned}\Pr[Y \leq w] \leq &\exp\left(-\frac{2\tau^2\epsilon_1^2}{\frac{2p-1}{2p-2} - \varphi} n\right) \\ &+ \exp\left[-2\left(\frac{2p-1}{2p-2} - \varphi - \varepsilon_1\right)\tau\varepsilon_2^2 n\right],\end{aligned}$$

where the threshold of tolerated failed test runs is set to $w = (1/k - \epsilon_2)\,[(1/k)(2p-1)/(2p-2) - \varphi - \varepsilon_1]\, t$.

Let us now focus on the first term and introduce the hypergeometrically distributed random variable $\bar{Z}$ counting the number of computation runs that are affected by the server's deviation. Then, for $\varepsilon_3 > 0$, tail bounds on the hypergeometric distribution imply

$$\Pr\left[\bar{Z} \geq \left(\frac{m}{n} + \varepsilon_3\right) d\right] \leq \exp\left(-\frac{2\delta^2\varepsilon_3^2}{\frac{2p-1}{2p-2} - \varphi} n\right).$$

Next, let $Z'$ be the random variable counting the number of computation runs that have not been affected by the server's deviation but that give a result distinct from $\bar{x}$ because of inherent failures of the algorithm. Note that $Z'$ conditioned on $\bar{Z}$ fixed to a specific value is binomially distributed. It hence follows that

$$\begin{aligned}&\Pr\left[Z' \geq (p + \varepsilon_4)\left(1 - \frac{m}{n} - \varepsilon_3\right) d \;\middle|\; \bar{Z} = \left(\frac{m}{n} + \varepsilon_3\right) d\right] \\ &\quad\leq \exp\left[-2\left(1 - \frac{2p-1}{2p-2} + \varphi - \varepsilon_3\right)\delta\varepsilon_4^2 n\right].\end{aligned}$$

Note that it holds that $Z = \bar{Z} + Z'$. Therefore, it follows that

$$\begin{aligned}\Pr\left[Z \geq \frac{d}{2}\right] &\leq \Pr\left[Z \geq \frac{d}{2} \;\middle|\; \bar{Z} \leq \left(\frac{m}{n} + \varepsilon_3\right) d\right] \\ &\quad + \Pr\left[\bar{Z} \geq \left(\frac{m}{n} + \varepsilon_3\right) d\right] \\ &\leq \Pr\left[Z' \geq \frac{d}{2} - \left(\frac{m}{n} + \varepsilon_3\right) d \;\middle|\; \bar{Z} \right. \\ &\quad \left. = \left(\frac{m}{n} + \varepsilon_3\right) d\right] \\ &\quad + \Pr\left[\bar{Z} \geq \left(\frac{m}{n} + \varepsilon_3\right) d\right].\end{aligned}$$

Using the inequalities from above, we arrive at

$$\Pr\left[Z \geq \frac{d}{2}\right] \leq \exp\left[-2\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)\delta\varepsilon_4^2 n\right] + \exp\left(-\frac{2\delta^2\varepsilon_3^2}{\frac{2p-1}{2p-2}-\varphi}n\right),$$

where we set

$$\frac{d}{2}-\left(\frac{m}{n}+\varepsilon_3\right)d = (p+\varepsilon_4)\left(1-\frac{m}{n}-\varepsilon_3\right)d.$$

This condition can be rewritten as

$$\frac{1}{2}-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3 = (p+\varepsilon_4) \times\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)$$

or, equivalently,

$$\varepsilon_4 = \left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)^{-1} \times\left(\frac{1}{2}-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)-p.$$

It can readily be seen that this equation has solutions $\varepsilon_3, \varepsilon_4 > 0$ when $\varphi$ is fixed.

We finally conclude that

$$\begin{aligned}\Pr[\text{fail}] \leq \max\Bigg\{&\exp\left[-2\left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)\delta\varepsilon_4^2 n\right] \\ &+\exp\left(-\frac{2\delta^2\varepsilon_3^2}{\frac{2p-1}{2p-2}-\varphi}n\right), \\ &\exp\left(-\frac{2\tau^2\epsilon_1^2}{\frac{2p-1}{2p-2}-\varphi}n\right) \\ &+\exp\left[-2\left(\frac{2p-1}{2p-2}-\varphi-\varepsilon_1\right)\tau\varepsilon_2^2 n\right]\Bigg\} \quad \text{(E5)}\end{aligned}$$

for

$$\begin{aligned} w &= (1/k-\epsilon_2)\left(\frac{2p-1}{2p-2}-\varphi-\varepsilon_1\right)t, \\ 0 &< \varphi < \frac{2p-1}{2p-2}, \\ 0 &< \varepsilon_1 < \frac{1}{2}-\varphi, \\ 0 &< \varepsilon_2 < \frac{1}{k}, \\ 0 &< \varepsilon_3 < \varphi, \\ \varepsilon_4 &= \left(1-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)^{-1} \\ &\quad\times\left(\frac{1}{2}-\frac{2p-1}{2p-2}+\varphi-\varepsilon_3\right)-p. \end{aligned}$$

To obtain an optimal bound, this expression must be minimized over $\varepsilon_1$, $\varepsilon_2$, $\varepsilon_3$, and $\varphi$.

Irrespective of the exact form of the optimal bound, choosing $\varphi$, $\varepsilon_1$, $\varepsilon_2$, and $\varepsilon_3$ sufficiently small implies the existence of protocols with verification exponential in $n$, for any fixed $0 < w/t < (1/k)(2p-1)/(2p-2)$ and fixed $(d/n), (t/n) \in (0,1)$.

## 7. Optimality of the bound

To obtain the improved bound above, $Z_2$ is introduced as the count of nonaffected computation runs yielding the correct result—i.e., accept on yes instances and reject on no instances. Making sure that $Z_2$ would be greater than $d/2$ ensures that no matter what happens on computation runs that would yield an incorrect result, there is no possibility of being mistaken and rejecting in place of accepting and vice versa. Yet, one might wonder if the situation is not more favorable: if the deviation by the server induces a flip of the accept or reject, then could it be possible that some of the runs yielding an incorrect result would be corrected by the deviation? At first sight, this could be motivated by the fact that the computation being blind, the server could not possibly craft an attack that would selectively affect the runs yielding the correct results. Unfortunately, this intuition is wrong: blindness does not rule out attacks that have different effects depending on the result of the computation itself.

To see this, consider the following situation. Consider an algorithm solving a decision problem deterministically, so that in case of a yes instance, the algorithm outputs $|+\rangle$ and in the case of a no instance, the output is $|-\rangle$. This deterministic algorithm yields a trivial randomized algorithm where a second qubit is generated in state $\alpha|0\rangle+\beta|1\rangle$, with $|\alpha|^2 > 2/3$. The new algorithm would take the output of the first one and apply a CZ gate between both qubits so that when the second qubit is traced out, the first one yields the correct answer with probability $|\alpha|^2$. Yet, nothing could rule out an alternate implementation where after the control-$Z$ gate, the state of the first qubit undergoes two $H$ gates controlled by the second qubit being $|0\rangle$. Clearly, this operation applies the identity to the first qubit as $H^2 = I$. However, if the server applies a $X$ gate on the first qubit between these two control-$H$ gates, it will amount to a deviation consisting of a $Z$ gate applied only when the second qubit is $|0\rangle$. As a result, its

attack only affects runs with the correct result. Note that the attack affects correct outcomes only, because in between the two control-$H$ gates, the computational branch for correct outcomes yields a state in the computational basis, while for incorrect ones it is the $|\pm\rangle$ basis. This property is true independent of the quantum one-time-pad encryption of the states and can hence be applied on an encrypted computation.

This example might seem excessively artificial, but such situations cannot be ruled out *a priori*, i.e., without an extensive understanding of the algorithm being implemented and of the proposed implementation. In fact, a similar situation [21] has already been encountered in the context of multiparty quantum computation, where attacks could be crafted to evade detection when using less obvious inappropriate implementations.

## APPENDIX F: PROOF OF NOISE ROBUSTNESS

Recall that the constant ratios of test, computation, and tolerated failed test rounds are given by $\delta = d/n$, $\tau = t/n$ and $\omega = w/t$. We define the acceptance of the protocol to be the probability that the client does not abort at the end of an execution. We then bound this probability in two regimes: (i) if the maximal noise $p_{\max}$ is smaller than the (ratio) threshold of failed test runs, the protocol accepts with high probability; (ii) if the noise of the device is too large, i.e., $p_{\min}$ is already too large compared to the threshold, the protocol will most certainly abort.

**Lemma 4** (Acceptance on Noisy Devices)**.** *Assume a Markovian round-dependent model for the noise on the client and server devices and let $p_{\min} \leq p_{\max} < 1/2$ be, respectively, a lower and an upper bound on the probability that at least one of the trap-measurement outcomes in a single test round is incorrect.*

*If $\omega > p_{\max}$, then the probability that the client does not accept at the end of Protocol 1 is bounded by exponentially small $\epsilon_{\text{rej}}$ where*

$$\epsilon_{\text{rej}} = \exp\left[-2(\omega - p_{\max})^2 \tau n\right]. \tag{F1}$$

*On the other hand, if $\omega < p_{\min}$, then the client's acceptance in Protocol 1 is exponentially small and bounded by* $\exp\left[-2(p_{\min} - \omega)^2 \tau n\right]$.

*Proof.* We define the random variable $Y$ that corresponds to the number of failed test rounds during one execution of the protocol. We call Ok the event that the client accepts at the end of the protocol—if not too many test rounds fail, meaning that $Y < w$.

1. **For $\omega > p_{\max}$**
   Equivalently, we have that $w > tp_{\max}$. We are looking to lower bound the probability that an honest round does not abort:

   $$\Pr[\mathsf{Ok}] = \Pr[Y < w].$$

   Note that $Y$ describes exactly the number of test rounds in which at least one trap-measurement outcome is incorrect (by definition of a failed test round). The probability that a given test round fails is therefore upper bounded by $p_{\max}$. Let $\hat{Y}_1$ be a random variable following a $(t, p_{\max})$-binomial distribution. Since we suppose that the noise is not correlated across rounds, $Y$ is upper bounded by $\hat{Y}_1$ in the usual stochastic order:

   $$\Pr[Y < w] \geq \Pr\left[\hat{Y}_1 < w\right] = 1 - \Pr\left[\hat{Y}_1 \geq w\right]$$

   Further, since $\mathbb{E}\left[\hat{Y}_1\right] = tp_{\max} < w$, application of Lemma 3 yields

   $$\begin{aligned}\Pr\left[\hat{Y}_1 \geq w\right] &\leq \exp\left(-2\frac{(tp_{\max} - w)^2}{t}\right) \\ &= \exp\left[-2(\omega - p_{\max})^2 \tau n\right] = \epsilon_{\text{rej}}.\end{aligned}$$

2. **For $\omega < p_{\min}$**
   In that case, we have that $w < tp_{\min}$. We show that the probability of accepting is upper bounded by a negligible function. Let $\hat{Y}_2$ be a random variable following a $(t, p_{\min})$-binomial distribution; $Y$ then is lower bounded by $\hat{Y}_2$ in the usual stochastic order:

   $$\Pr[Y < w] \leq \Pr\left[\hat{Y}_2 < w\right].$$

   Since $w < tp_{\min}$, using Lemma 3 directly and with the same simplifications as above, we obtain

   $$\Pr\left[\hat{Y}_2 < w\right] \leq \exp\left[-2(p_{\min} - \omega)^2 \tau n\right],$$

   concluding the proof. ■

**Theorem 4** (Local Correctness of VDQC Protocol on Noisy Devices)**.** *Assume a Markovian round-dependent model for the noise on the client and server devices and let $p_{\max}$ be an upper bound on the probability that at least one of the trap-measurement outcomes in a single test round is incorrect.*

*If $p_{\max} < \omega < (1/k)(2p - 1)/(2p - 2)$, then the protocol is $\epsilon_{\text{cor}}$ locally correct with exponentially small $\epsilon_{\text{cor}} = \epsilon_{\text{rej}} + \epsilon_{\text{ver}}$, with $\epsilon_{\text{rej}}$ from Lemma 4 and $\epsilon_{\text{ver}}$ from Theorem 3.*

*Proof.* We call Ok the event that the client accepts at the end of the protocol—if not too many test rounds fail—and Correct the event corresponding to a correct output—if only a few of the computation rounds have their output bits flipped.

We are looking to lower bound the probability of an honest round producing the correct outcome and not aborting:

$$\Pr[\mathsf{Correct} \wedge \mathsf{Ok}] = \Pr[\mathsf{Ok}] - \Pr[\neg\mathsf{Correct} \wedge \mathsf{Ok}].$$

As $p_{\max} < (1/k)(2p-1)/(2p-2) < 1/2$, from Lemma 4 we have

$$\Pr[\mathsf{Ok}] \geq 1 - \epsilon_{\mathrm{rej}}.$$

Since $\omega < (1/k)(2p-1)/(2p-2)$, the parameters of Protocol 1 comply with Theorem 3, from which we obtain that

$$\Pr[\neg\mathsf{Correct} \wedge \mathsf{Ok}] \leq \epsilon_{\mathrm{ver}}.$$

It follows that

$$\Pr[\mathsf{Correct} \wedge \mathsf{Ok}] \geq 1 - \epsilon_{\mathrm{rej}} - \epsilon_{\mathrm{ver}},$$

which concludes the proof. ■


[1] A. Broadbent, J. Fitzsimons, and E. Kashefi, *Measurement-Based and Universal Blind Quantum Computation* (Springer-Verlag, Berlin, 2010), p. 43, ISBN 978-3-642-13678-8, https://doi.org/10.1007/978-3-642-13678-8˙2.

[2] J. F. Fitzsimons and E. Kashefi, Unconditionally verifiable blind quantum computation, Phys. Rev. A **96**, 012303 (2017).

[3] A. Gheorghiu, T. Kapourniotis, and E. Kashefi, Verification of quantum computation: An overview of existing approaches, Theory Comput. Syst. **63**, 715 (2019), ISSN 1433-0490.

[4] A. Gheorghiu, M. J. Hoban, and E. Kashefi, A simple protocol for fault tolerant verification of quantum computation, Quantum Sci. Technol. **4**, 015009 (2018).

[5] T. Kapourniotis and A. Datta, Nonadaptive fault-tolerant verification of quantum supremacy with noise, Quantum **3**, 164 (2019), ISSN 2521-327X.

[6] T. Morimae and K. Fujii, Secure Entanglement Distillation for Double-Server Blind Quantum Computation, Phys. Rev. Lett. **111**, 020502 (2013).

[7] U. Mahadev, in *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7–9, 2018*, edited by M. Thorup (IEEE Computer Society, Paris, 2018), p. 259, https://doi.org/10.1109/FOCS.2018.00033.

[8] U. Maurer and R. Renner, in *Innovations in Computer Science* (Tsinghua University Press, Beijing, 2011), p. 1, ISBN 978-7-302-24517-9, https://conference.iiis.tsinghua.edu.cn/ICS2011/content/papers/14.html.

[9] S. Barz, E. Kashefi, A. Broadbent, J. F. Fitzsimons, A. Zeilinger, and P. Walther, Demonstration of blind quantum computing, Science **335**, 303 (2012), ISSN 0036-8075.

[10] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, Experimental verification of quantum computation, Nat. Phys. **9**, 727 (2013), ISSN 1745-2481.

[11] C. Greganti, M.-C. Roehsner, S. Barz, T. Morimae, and P. Walther, Demonstration of measurement-only blind quantum computing, New J. Phys. **18**, 250 (2016).

[12] W. McCutcheon, A. Pappa, B. A. Bell, A. McMillan, A. Chailloux, T. Lawson, M. Mafu, D. Markham, E. Diamanti, and I. Kerenidis *et al.*, Experimental verification of multipartite entanglement in quantum networks, Nat. Commun. **7**, 13251 (2016), ISSN 2041-1723.

[13] M. Hein, J. Eisert, and H. J. Briegel, Multiparty entanglement in graph states, Phys. Rev. A **69**, 062311 (2004).

[14] V. Danos and E. Kashefi, Determinism in the one-way model, Phys. Rev. A **74**, 052310 (2006).

[15] V. Danos, E. Kashefi, and P. Panangaden, The measurement calculus, J. ACM **54**, 8-es (2007), ISSN 0004-5411.

[16] E. Kashefi and P. Wallden, Optimised resource construction for verifiable quantum computation, Journal of Physics A: Mathematical and Theoretical, arXiv:1510.07408 (2017), http://iopscience.iop.org/10.1088/1751-8121/aa5dac.

[17] Q. Xu, X. Tan, and R. Huang, Improved resource state for verifiable blind quantum computation, Entropy **22**, 996 (2020), ISSN 1099-4300, https://www.mdpi.com/1099-4300/22/9/996.

[18] Variable $w$ would typically be set by the client given its *a priori* understanding of the quality of the server. As explained in Sec. V, this does not affect security: a higher value would induce more rounds than necessary to achieve a given confidence level, while a lower value would risk aborting with high probability.

[19] R. König, R. Renner, A. Bariska, and U. Maurer, Small Accessible Quantum Information Does Not Imply Security, Phys. Rev. Lett. **98**, 140502 (2007).

[20] C. Portmann and R. Renner, Cryptographic security of quantum key distribution, arXiv:1409.3525 (2014).

[21] T. Kapourniotis, E. Kashefi, L. Music, and H. Ollivier, *Delegating multi-party quantum computations vs. dishonest majority in two quantum rounds*, arxiv:2102.12949 (2021).

[22] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner, in *Advances in Cryptology—ASIACRYPT 2014*, edited by P. Sarkar and T. Iwata (Springer-Verlag, Berlin, 2014), p. 406, ISBN 978-3-662-45608-8.

[23] This can be done once by the server for its architecture and later shared with the client before starting the protocol as a service.

[24] W. Feller, *An Introduction to Probability Theory and Its Applications* (Wiley, 1991), ISBN 978-0-471-25709-7, https://www.wiley.com/en-us/An+Introduction+to+Probability+Theory+and+Its+Applications%2C+Volume+2%2C+2nd+Edition-p-9780471257097.

[25] E. Greene and J. A. Wellner, Exponential bounds for the hypergeometric distribution, Bernoulli **23**, 1911 (2017), ISSN 1350-7265.

[26] R. J. Serfling, Probability inequalities for the sum in sampling without replacement, Ann. Statist. **2**, 39 (1974).